การพัฒนาระบบการจัดกลุ่มข้อมูลโดยใช้ ROCK Algorithm

โดย

นาย วรพงษ์ ริ้วเลิศธรรม

รหัส 43067096

อาจารย์ที่ปรึกษา

ผศ.ดร. วรพจน์ กรีสุระเดช

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2544

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาระบบการจัดกลุ่มข้อมูลโดยใช้ ROCK Algorithm
นักศึกษา	นายวรพงษ์ ริ้วเลิศธรรม
อาจารย์ที่ปรึกษา	ผศ.ดร.วรพจน์ กรีสุระเดช
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

เนื่องจากการตัดสินใจทางธุรกิจจำเป็นต้องใช้สารสนเทศค่อนข้างมาก ซึ่งได้มาจากการประมวลผลฐานข้อมูลขนาดใหญ่ ทำให้มีการนำเทคนิคต่างๆ ของ Data Mining มาใช้มากขึ้นเพื่อค้นหาความรู้และความสัมพันธ์ที่ซ่อนอยู่ในฐานข้อมูลเหล่านั้น

การพัฒนาโครงการครั้งนี้มีเป้าหมายเพื่อนำเทคนิคของ Data Mining มาใช้ในการจัดกลุ่มลูกค้าหรือที่เรียกว่าการทำ Clustering โดยข้อมูลที่นำมาใช้เป็นข้อมูลการพิจารณาอนุมัติสินเชื่อของธนาคารแห่งหนึ่ง ซึ่งส่วนใหญ่เป็นข้อมูลประเภท Categorical โครงการนี้จึงมุ่งเน้นที่การพัฒนาและปรับปรุงเทคนิคของ Algorithm เพื่อให้สามารถจัดการกับ Categorical Data ในฐานข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ

แนวทางการพัฒนาโครงการนี้ จะใช้ระบบจัดการฐานข้อมูล SQLServer 7.0 และใช้ภาษา VBA ในการเขียนโปรแกรม โดยได้เลือก Algorithm ROCK ซึ่งเป็น Hierarchical Clustering Technique ประเภทหนึ่งที่สามารถจัดกลุ่มข้อมูลประเภท Categorical ได้ดี ซึ่งคาดหวังว่าสามารถปรับปรุงจนนำมาใช้กับฐานข้อมูลขนาดใหญ่ได้ในระดับที่น่าพอใจ

Title	Developing clustering tools to implement ROCK Algorithm
Student	Mr.Worraphong Ryulerttham
Advisor	Asst.Prof.Dr.Worapoj Kreesuradej
Level of study	Master of Science in Information Technology
Major	Information Technology Management /Information Science
Academic Year	2001

ABSTRACT

It is the fact that in business decision making, a lot of information is needed and most of them were retrieved from many very large data warehouses. A various technique in Data Mining was also applied to mine the knowledge and the relationship of these data hidden in the data warehouses.

The development of this project aimed to applied the Data Mining technique called 'clustering/segmentation' to grouping the customer who belong to the same profile or same behavior. This project focused on developing the tool to manage the categorical data of banking credit approval efficiently. It was also expected to be able to applied to other type of business data.

The tool was developed by using VBA, Transact SQL on SQL Server 7.0 to implement ROCK algorithm, one type of hierarchical agglomerative clustering technique.

สารบัญ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมา

ในธุรกิจที่มุ่งหวังผลกำไรนั้น การดำเนินงานจะมุ่งเน้นที่การเพิ่มยอดขาย การลดต้นทุน รวมถึงการจัดการกับความเสี่ยงที่อาจจะเกิดขึ้นในรูปแบบต่างๆ ยอดขายที่เพิ่มขึ้นมานั้น ส่วนหนึ่งจะเกิดจากกิจกรรมทางการตลาดในการจัดโปรแกรมส่งเสริมการขายรูปแบบต่างๆ เพื่อตอบสนองลูกค้าแต่ละกลุ่มอย่างเหมาะสม เพื่อให้ได้รับอัตราการตอบรับที่สูง และมีความเสี่ยงอยู่ในระดับที่ยอมรับได้ กิจกรรมเหล่านี้มักจะเริ่มต้นด้วยการกำหนดกลุ่มลูกค้าเป้าหมายแยกเป็นกลุ่มๆ อย่างชัดเจน

การจัดการกับข้อมูลทางการตลาดอย่างมีประสิทธิภาพ จะช่วยให้กิจกรรมข้างต้นเป็นไปอย่างถูกต้อง เหมาะสม และสร้างสรรค์ Data Mining จึงถูกนำมาใช้ในการค้นหาความรู้ที่ซ่อนอยู่ในฐานข้อมูลขนาดใหญ่ เพื่อจัดกลุ่มข้อมูลที่มีลักษณะเหมือนกันหรือคล้ายๆ กัน โดยเรียกกระบวนการนี้ว่าการทำ Clustering หรือ Segmentation

สำหรับธุรกิจการให้สินเชื่อของธนาคารที่มีลูกค้าจำนวนมากนั้นจำเป็นต้องมีการคัดเลือกเฉพาะลูกค้าที่ดี หรือมีแนวโน้มที่ดีมาเป็นลูกค้า เพื่อลดความเสี่ยงในการทำธุรกิจ โดยผ่านระบบการให้คะแนนตามข้อมูลที่กรอกมาในใบสมัคร กระบวนการนี้เรียกว่าการทำ Credit Scoring ส่วนเครื่องมือและเกณฑ์ต่างๆ ในการให้คะแนนจะเรียกรวมกันว่า Credit Scorecard

การทำงานของ Scoring นั้น จำเป็นต้องมีการกำหนดลักษณะและคุณสมบัติของลูกค้าว่าอะไรดี อะไรไม่ดี และมีความเสี่ยงมากน้อยเพียงใด โดยใช้ข้อมูลจากประวัติลูกค้าในอดีตในการคาดการณ์พฤติกรรมของลูกค้าในอนาคต สิ่งนี้ถือเป็นหัวใจสำคัญของ Scorecard ว่าจะสามารถคัดเลือกลูกค้าได้ตรงตามกลุ่มเป้าหมายที่วางแผนไว้หรือไม่ หากเกณฑ์ในการคัดเลือกลูกค้าของ Scorecard ไม่ดีหรือไม่เหมาะสม ก็จะทำให้ได้ลูกค้าที่มีความเสี่ยงสูงเข้ามา อันจะสร้างความเสียหายต่อธนาคารได้

ด้วยเหตุนี้ธนาคารจึงมีความพยายามจะนำ Data Mining มาใช้ในการสร้าง Customer Profile ของลูกค้ากลุ่มต่างๆ ตามระดับความเสี่ยง การจัดกลุ่มลูกค้าเพื่อให้สามารถออกผลิตภัณฑ์การกำหนดโปรแกรมส่งเสริมการขายได้เหมาะสม รวมถึงการบริหารความเสี่ยงจากกิจกรรมเหล่านี้ได้ดียิ่งขึ้น

1.2 วัตถุประสงค์

1. พัฒนาเครื่องมือสำหรับจัดกลุ่มข้อมูลประเภท Categorical โดยในโครงการนี้ใช้ข้อมูลลูกค้าสินเชื่อธนาคารเป็นข้อมูลทดสอบ
2. เพื่อเพิ่มความเข้าใจในการออกแบบและการประยุกต์ใช้เทคโนโลยีการไมนิ่งข้อมูลด้วยวิธี Clustering และวิธีการนำมาใช้กับข้อมูลประเภท Categorical
3. เพื่อเป็นแนวทางในการ ออกแบบและพัฒนาเครื่องมือสำหรับการทำไมนิ่งเพื่อจัดกลุ่มข้อมูลประเภท Categorical ในรูปแบบอื่นๆ ต่อไป
4. สามารถนำผลลัพธ์ไปในใช้ประกอบการตัดสินใจ การกำหนดกลยุทธ์ทางธุรกิจได้อย่างเหมาะสม

1.3 แนวทางการศึกษา

1. ศึกษาแนวทางการนำ Data Mining มาประยุกต์ใช้กับการจัดกลุ่มข้อมูลประเภท Categorical
2. กำหนดขอบเขตการทำงาน
3. ศึกษา Algorithm การจัดกลุ่มข้อมูลประเภท Categorical ข้อดีและข้อเสียของแต่ละวิธี
4. พัฒนาโปรแกรม โดยใช้ VBA ในการสร้าง User Interface เพื่อติดต่อกับผู้ใช้งาน และใช้ Transact SQL สำหรับ SQL Server 7.0 ในการสร้าง Stored Procedure ฝังไว้ที่ฝั่ง Database Server
5. ทดสอบและสรุปผลลัพธ์จากการทำงานของโปรแกรม

1.4 ขอบเขตโครงการ

โปรแกรมที่พัฒนาขึ้นตามโครงการนี้ มีเป้าหมายเพื่อจัดทำเครื่องมือสำหรับ Clustering ข้อมูลประเภท Categorical เพื่อให้ทราบถึง Customer profile ของลูกค้าแต่ละกลุ่ม รวมถึงระดับความเสี่ยงตามพฤติกรรมของลูกค้าแต่ละกลุ่มด้วย อันจะช่วยให้สามารถกำหนดกลุ่มเป้าหมายในการออกผลิตภัณฑ์และโปรแกรมต่างๆ ได้อย่างเหมาะสม และสร้างสรรค์

โปรแกรมที่พัฒนานี้สามารถประยุกต์ใช้กับข้อมูลธุรกิจประเภทอื่นๆ ได้ อย่างไรก็ตามในโครงการนี้ได้นำข้อมูลสินเชื่อธนาคารมาใช้ทดสอบ ส่วนการนำผลลัพธ์ไปประยุกต์ใช้ในการตัดสินใจทางธุรกิจนั้น ไม่ได้รวมอยู่ในขอบเขตของโครงการนี้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการทำงานของ Clustering Algorithm ที่เลือกใช้ในการไมนิ่งข้อมูล พร้อมทั้งสามารถพัฒนา และปรับปรุงการทำงานของ Algorithm นี้ให้ดียิ่งขึ้น
2. สามารถนำข้อมูลที่มีอยู่แล้วในองค์กรมาสร้างสารสนเทศที่มีประโยชน์เพิ่มขึ้นได้
3. สามารถนำผลลัพธ์ที่ได้ไปประยุกต์ใช้ในการตัดสินใจวางแผนนโยบายทางธุรกิจได้ดีและเหมาะสมขึ้น

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ความรู้เบื้องต้น

Clustering เป็นกระบวนการการจัดกลุ่มข้อมูลในฐานข้อมูลที่มีลักษณะเหมือนหรือใกล้เคียงกัน ข้อมูลแต่ละกลุ่มที่ได้นั้นจะเรียกว่า "Cluster" ซึ่งสามารถนำมาใช้ในการระบุกลุ่มลูกค้าประเภทต่างๆ ตามพฤติกรรมการบริโภค ก่อนจัดทำโปรแกรมส่งเสริมการขายให้เหมาะสมสำหรับลูกค้าแต่ละกลุ่ม เกิดอัตราการตอบสนองที่สูง (Response Rate) รวมทั้งช่วยในการคัดเลือกเฉพาะลูกค้าที่มีความเสี่ยงตามระดับที่ยอมรับได้

นอกจากนี้ Clustering ยังเป็นเทคนิคหนึ่งที่นิยมนำมาใช้ในกระบวนการทำ Data Preprocessing ของ Algorithm อื่นๆ เพื่อจัดการกับข้อมูลที่มีความผิดพลาด หรือขาดหายไป

Clustering เป็นการทำงานแบบ "Unsupervised learning" คือไม่ต้องมีการกำหนดรูปแบบและวิธีการทำงานไว้ล่วงหน้า โดยส่วนใหญ่การทำ Clustering เพื่อพิจารณาว่าข้อมูลแต่ละรายการมีความเหมือนกันมากน้อยเพียงใดนั้นจะใช้วิธีการหาค่าที่แสดงระยะห่างของข้อมูลแต่ละรายการกับค่ากลางค่าหนึ่งที่ถือเป็นตัวแทนของกลุ่มข้อมูลนั้น และเรียกวิธีการแบบนี้ว่า "Distance-based clustering analysis"

แนวโน้มการพัฒนาเทคนิคการทำ Clustering นั้นมักจะมุ่งเน้นที่วิธีการทำงานเพื่อให้สามารถวิเคราะห์ข้อมูลสำหรับฐานข้อมูลขนาดใหญ่ๆ ได้อย่างมีประสิทธิภาพ การวิจัยต่างๆ จึงเน้นไปที่การปรับปรุงการทำงานของ Algorithm ให้มีความเสถียร รองรับการวิเคราะห์และจัดกลุ่มข้อมูลได้ทั้งประเภท Categorical และ Numerical

2.2 ประเภทของ Clustering Method

ปัจจุบันมี Algorithm ในการทำ Clustering มากมาย การเลือกใช้ก็ต่างกันไปตามวัตถุประสงค์และประเภทข้อมูลที่ Algorithm นั้นรองรับ ซึ่งบาง Algorithm ต้องมีการนำหลายๆ Algorithm มาใช้ร่วมกัน

โดยทั่วไป สามารถแบ่งประเภทของ Clustering ได้เป็น 5 ประเภทหลักๆ ได้แก่

2.2.1 Partitioning methods

การทำงานจะแบ่งจำนวนข้อมูลในฐานข้อมูล (จำนวนเรคอร์ด) n ออกเป็น Cluster ต่างๆ จำนวน k Cluster โดย k <= n และ

- แต่ละกลุ่มต้องมีอย่างน้อย 1 เรคอร์ด
- แต่ละเรคอร์ด ต้องถูกจัดอยู่ในกลุ่มใดกลุ่มหนึ่งอย่างน้อย 1 กลุ่ม

การทำ Clustering วิธีนี้ ต้องกำหนดค่า k หรือจำนวน Cluster ว่าต้องการจัดกลุ่มเป็นกี่กลุ่ม จากนั้น algorithm จะทำการสุ่มสร้าง Cluster เริ่มต้นขึ้นมาแล้วใช้เทคนิคที่เรียกว่า "Iterative relocation" ในการวนลูปเพื่อย้ายเรคอร์ดที่มีคุณสมบัติใกล้เคียงกันไปยังกลุ่มต่างๆ ที่สร้างขึ้นมาจนครบทั้งฐานข้อมูล กระบวนการนี้จะถูกทำซ้ำจนกระทั่งค่าที่ใช้วัดความเหมาะสมในการทำ Clustering ที่เรียกว่าค่า square-error ที่เกิดขึ้นเป็นศูนย์หรือเบนเข้าหาค่าที่ตั้งไว้

Algorithm ที่ได้รับความนิยมสำหรับเทคนิคนี้ คือ "K-Means" ซึ่งใช้ mean เป็นค่าวัดความเหมือนหรือความต่างกันระหว่างข้อมูลแต่ละเรคอร์ดกับค่า Mean ของ Centroid ในแต่ละ Cluster

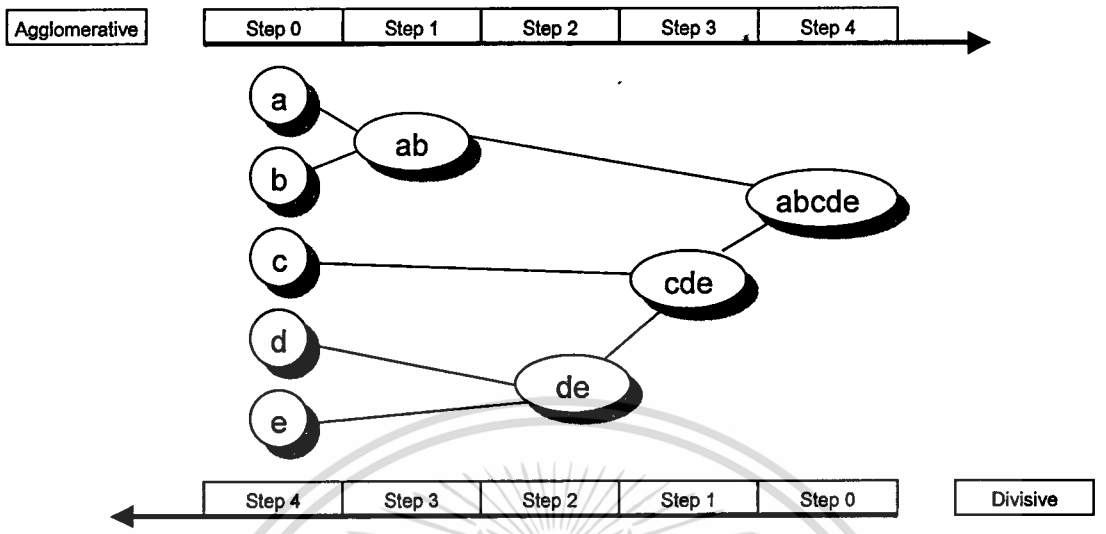
วิธีข้างต้นเหมาะสมกับข้อมูลประเภทตัวเลข และการจัดการฐานข้อมูลขนาดเล็กจนถึงขนาดกลาง แต่ไม่เหมาะสมในการใช้กับฐานข้อมูลขนาดใหญ่ อย่างไรก็ตาม Algorithm ประเภทนี้ก็ได้รับการพัฒนาและปรับปรุงให้สามารถนำไปใช้กับงานลักษณะดังกล่าวได้ ซึ่งจะเห็นได้จากมีเอกสารการวิจัยและ Extended procedure ออกมามากมายที่พัฒนาบน method ประเภทนี้

2.2.2 Hierarchical methods

เป็นการจัดกลุ่มข้อมูลโดยสร้างเป็นชั้นๆ แบบลำดับขั้น ซึ่งมี 2 ประเภทได้แก่

1) Agglomerative หรือแบบ Bottom-Up คือจะทำการรวมข้อมูลจากระดับล่างสุด (ทีละเรคอร์ด) ขึ้นมาเป็นระดับที่สูงขึ้น จนสุดท้ายได้เป็นกลุ่มข้อมูลกลุ่มใหญ่ (ดังรูป 1)

2) Divisiveหรือแบบ Top-Down ซึ่งจะมีการทำงานตรงกันข้ามกับ Agglomerative คือจากระดับบนสุดจะทำการแตกลงมาเป็นระดับล่างสุด



รูปที่ 2.1 แสดง Hierarchical Clustering แบบ Agglomerative และ Divisive

การวัดความเหมือนของแต่ละเรคอร์ด จะทำแบบทุกทิศทาง จากนั้นจึงทำการรวมกลุ่มเรคอร์ดที่มีค่าความเหมือนอยู่ในระดับเดียวกัน การหาความเหมือนกันของแต่ละเรคอร์ดนั้น จะเปรียบเทียบกันแบบทุกทิศทาง กล่าวคือถ้าฐานข้อมูลมี n เรคอร์ด แต่ละเรคอร์ดจะถูกเปรียบเทียบความเหมือนกัน (n-1) ครั้ง จากนั้นเรคอร์ดที่มีค่าความเหมือนอยู่ในระดับเดียวกันจะถูกรวมอยู่ในกลุ่มเดียวกัน

จากวิธีการข้างต้น พบว่าวิธีการนี้ค่อนข้างใช้เวลาในการประมวลนาน เพราะต้องมีการวัดค่าความเหมือนกันของทุกๆ เรคอร์ดในฐานข้อมูล นอกจากนี้หลังจากทำการ รวม หรือ แตก Cluster แล้วจะไม่สามารถกลับไปทำใหม่ได้ ทำให้ไม่สามารถแก้ไข error ที่เกิดขึ้นได้ อย่างไรก็ตามปัญหาดังกล่าวก็ได้รับการแก้ไขโดยใช้กระบวนการที่เรียกว่า CURE, Chameleon และ BIRCH (รายละเอียดของกระบวนการเหล่านี้ไม่ได้อยู่ในขอบเขตของรายงานแบบนี้)

2.2.3 Density-based method

Method นี้จะจัดกลุ่มข้อมูลโดยการพิจารณาที่ความหนาแน่นของข้อมูลแทนการใช้ระยะห่างของแต่ละเรคอร์ดกับค่าตัวแทนของ Cluster โดยจะทำการขยายขนาดของ Cluster แต่ละ Cluster ไปเรื่อยๆ ตราบใดที่ Cluster ใกล้ๆ กัน(neighborhood) มีค่าเกินค่าหนึ่งที่ตั้งไว้

ตัวอย่างของ Algorithm ประเภทนี้ได้แก่ DBSCAN และ OPTICS

2.2.4 Grid-based method

Method นี้จะทำการ Quantize ข้อมูลไปเรื่อยๆ จนได้ Cells จำนวนหนึ่งเพื่อใช้ในการสร้างโครงสร้าง Grid ข้อดีของ method นี้คือ ทำงานได้เร็ว ไม่ขึ้นกับจำนวนของเรคอร์ดของฐานข้อมูล

ตัวอย่างของ Algorithm ประเภทนี้ได้แก่ STING, CLIQUE และ Wave Cluster

2.2.5 Model-based method

Method นี้จะทำการจัดกลุ่มข้อมูลโดยการเทียบเคียงกับ Model ที่สร้างขึ้นมาซึ่งจะมี 2 Approach หลักๆ คือ Statistical approach และ Neural network approach

ตัวอย่างของ Algorithm ประเภทนี้ได้แก่ COBWEB, CLASSIT

2.3 Algorithm ที่เกี่ยวข้องในการทำ Clustering

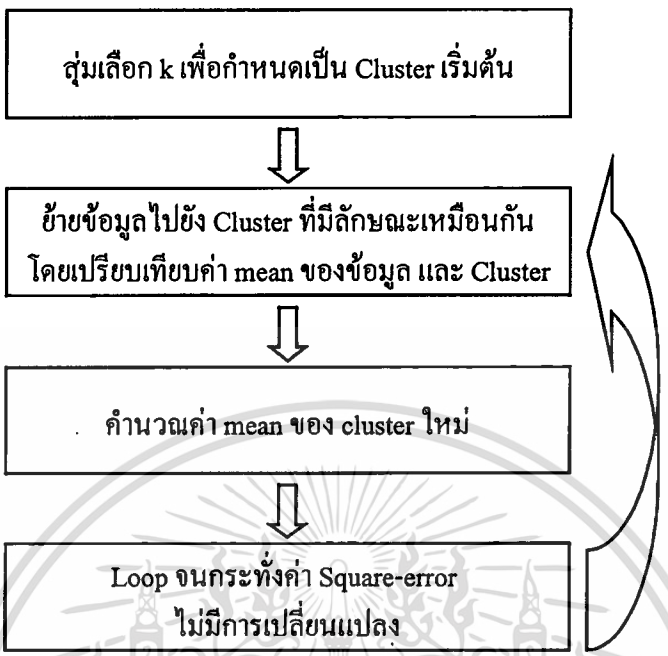
เนื้อหาในส่วนนี้เป็นการยกตัวอย่างการทำงานของ Algorithm K-Means ซึ่งเป็น Method ที่นิยมนำมาใช้งานในการทำ Clustering เนื่องจากไม่ซับซ้อน และสามารถทำงานได้ดีกับฐานข้อมูลที่มีขนาดไม่ใหญ่มากนัก ทั้งนี้เพื่อให้เห็นภาพรวมในการไมนิ่งข้อมูลเพื่อ Clustering ข้อมูลตลอดจนสามารถเปรียบเทียบและเลือกใช้ Algorithm ที่เหมาะสมกับข้อมูลแต่ละประเภทต่อไป

2.3.1 การทำงานของ K-Means

K-Means เป็น Algorithm ที่ได้รับความนิยมในการทำ Clustering เนื่องจากทำงานง่าย ไม่มีความซับซ้อนในการประมวลผล และจัดการกับฐานข้อมูลขนาดกลางได้ดี โดยซอฟต์แวร์ทางการค้าแทบทุกตัวจะสามารถทำ K-Means ได้ เช่น S-Plus, SAS, SPSS, Mineset

Algorithm นี้จะเริ่มต้นการทำงานด้วยการกำหนดค่าจำนวน Cluster หรือจำนวน partition ที่ต้องการ (k) ให้ เพื่อทำการจัดกลุ่มข้อมูลจำนวน n เรคอร์ด การหาค่าความเหมือนระหว่างเรคอร์ดต่างๆ กับ Cluster จะใช้การวัดระยะห่างตามการคำนวณแบบ Euclidean distance method โดยหากเรคอร์ดใดที่มีระยะห่างไม่เกินค่าที่ตั้งเอาไว้ ถือว่าเป็น Cluster เดียวกัน ก็จะทำการจัดกลุ่มเข้าไว้ด้วยกัน หากมีค่าระยะห่างมากกว่า ก็จะจัดไปอยู่กับ Cluster กลุ่มอื่น

การหาระยะห่างนี้จะทำแบบวนรอบไปเรื่อยๆ (Iterative) จนกระทั่งค่าเบี่ยงเบนความห่างนี้ (Square-Error) มีค่าไม่เปลี่ยนแปลงหรือเบนเข้าหาค่าค่าหนึ่งที่ตั้งเอาไว้



รูปที่ 2.2 แสดง Algorithm K-Means

ค่า Squared-error ซึ่งใช้เป็นตัวกำหนดความเหมาะสมในการจัดกลุ่มข้อมูลนี้ ใช้การคำนวณแบบ Euclidean distance method ซึ่งมีสูตรการคำนวณดังรูปข้างล่างนี้

กำหนด

เรคอร์ดแรกมีค่าของจุดต่างๆ เป็น $(x_1,x_2,x_3,\ldots x_d)$

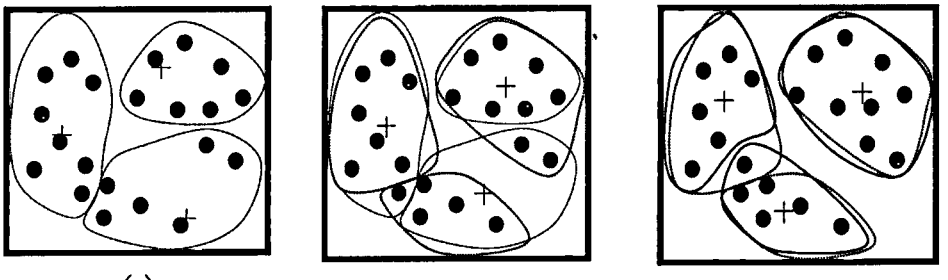
เรคอร์ดที่สองมีค่าของจุดต่างๆ เป็น $(y_1,y_2,y_3,\ldots y_d)$

ค่าที่แสดงระยะห่างระหว่าง 2 เรคอร์ดนี้คือ

$$\sum_{i=1}^{d} ((x_i - y_i)^2)^{1/2}$$

รูปที่ 2.3 แสดงการคำนวณหาค่าระยะห่างระหว่าง 2 เรคอร์ดด้วยวิธี Euclidean distance

จะเห็นได้ว่าการทำงานของ K-Means จำเป็นต้องมีการหาค่า mean ของ Cluster ทำให้ไม่เหมาะสมต่อการจัดกลุ่มข้อมูลประเภท Categorical เนื่องจากค่า mean ที่คำนวณอาจคลาดเคลื่อนได้ง่ายและไม่สมเหตุสมผลหากมีข้อมูลที่มี Noise หรือ Outlier อยู่ แม้จะมีจำนวนน้อยก็ตาม



รูปที่ 2.4 แสดงการทำงานของ K-Means ในการหาค่า mean ของ Cluster (แสดงด้วยเครื่องหมาย +) และการจัดกลุ่มที่เป็นการทำงานแบบ Iteration (Loop 3 ครั้ง ได้แก่ a-b-c)

จากข้อด้อยต่างๆ ของ Algorithm นี้ จึงได้มีการพัฒนาและปรับปรุงการทำงานในส่วนต่างๆ ของ Algorithm เช่น

1. กระบวนการในการสุ่มเลือกเรคอร์ดที่นำมาใช้เป็นค่า Cluster เริ่มต้น (initial k means) วิธีการคำนวณหาค่าความเหมือน/ความต่างเทียบกับค่ากลางของ Cluster (Dissimilarity) และวิธีการคำนวณหาค่า mean ของ Cluster โดยนำ Algorithm ของ Hierarchical agglomeration มาใช้ในการกำหนดจำนวน Cluster และการหาค่า Cluster เริ่มต้นก่อนทำ iterative relocation

2. การใช้ k-modes method ในการทำ Clustering ข้อมูลประเภท categorical โดยใช้ค่า Modes แทนการใช้ค่า Means

3. การใช้ EM (Expectation Maximization) ซึ่งจะพิจารณาค่าถ่วงน้ำหนักของความน่าจะเป็นของแต่ละเรคอร์ดในการจัดกลุ่มด้วย กล่าวคือจะไม่มีการกำหนดกรอบที่ตายตัวของ Cluster แต่ละ Cluster แต่จะขึ้นอยู่กับเกณฑ์ในการวัดหาค่าถ่วงน้ำหนักแทน

2.3.2 ข้อเสียของ K-Means ต่อข้อมูลประเภท Categorical

เนื่องจากข้อมูลสินเชื่อธนาคารส่วนใหญ่จะเป็นประเภท Categorical เช่น เพศ (ชาย, หญิง) อาชีพ (เกษตรกร, ครู, นักการเมือง, เจ้าหน้าที่ราชการ, วิศวกร, เลขาฯ ฯลฯ) จังหวัด-ภูมิลำเนา (กรุงเทพฯ นนทบุรี ชลบุรี ฯลฯ) เป็นต้น ซึ่งข้อมูลเหล่านี้หากนำ K-Means มาใช้อาจให้ผลลัพธ์ที่คลาดเคลื่อนได้ เนื่องจาก Algorithm นี้จะทำการแทนค่าข้อมูลประเภท Categorical เป็นข้อมูลประเภท Boolean ที่มีค่าเป็น True หรือ False เท่านั้น จากนั้นจึงหาระยะห่างของแต่ละเรคอร์ดกับค่า Mean ของ Centroid พิจารณาได้จากตัวอย่างข้างล่างนี้

ตัวอย่างการทำงานของ K-Means

กำหนดให้ข้อมูล 4 รายการมีข้อมูลดังนี้ {a,b,c,e} {b,c,d,e} {a,d} และ {f} ตามลำดับ

ในการจัดกลุ่มข้อมูล 4 รายการนี้ เซ็ตข้อมูลที่พิจารณาจะประกอบด้วย {a,b,c,d,e,f} ข้อมูลแต่ละรายการจะถูกแปลงเป็นค่า boolean ดังนี้ {1,1,1,0,1,0} {0,1,1,1,1,0} {1,0,0,1,0,0} และ {0,0,0,0,0,1} จากนั้นทำการหาค่าระยะห่างโดยใช้วิธีแบบ Euclidean distance ทำให้ค่าห่างของ 2 รายการแรกเป็น $2^{1/2}$ แล้วทำการ Merge 2 รายการนี้ทำให้ได้ค่าของ Centroid ใหม่เป็น {0.5,1,1,0.5,1,0} จากนั้นรายการที่ 3 และ 4 จะถูก Merge เข้าหากันเป็นอีก Cluster หนึ่ง เนื่องจากมีค่าระยะห่างเพียง $3^{1/2}$ ซึ่งน้อยกว่าค่าระยะห่างของ Centroid เดิมกับแต่ละรายการซึ่งมีค่าเป็น $3.5^{1/2}$ และ $4.5^{1/2}$ ตามลำดับ

จากตัวอย่างข้างต้นพบว่ารายการ {a,d} และ {f} จะถูกจัดอยู่ใน Cluster เดียวกัน ทั้งๆ ที่ไม่มีข้อมูลที่เกี่ยวข้องกันเลย ดังนั้นการจัดกลุ่มข้อมูลประเภท Categorical ด้วยวิธีการทางคณิตศาสตร์แบบ K-Means จึงไม่เหมาะสม

2.4 Algorithm ที่เลือก

จากตัวอย่างข้างต้นพบว่า ข้อมูลประเภท Categorical นั้นจะมีความสัมพันธ์ในลักษณะที่สามารถเชื่อมโยงข้อมูลถึงกันได้ (Link) ดังนั้นวิธีการที่ใช้ทำ Clustering นั้นควรพิจารณาถึงจำนวน Link ที่แต่ละรายการมีต่อกันมากกว่าการใช้ค่า Mean เพื่อหาระยะห่างระหว่างแต่ละรายการหรือระหว่าง Cluster

ผลลัพธ์ที่ได้จากการใช้ Link นั้นจะทำให้รายการประเภท {a,d} และ {f} ไม่มีทางที่จะจัดกลุ่มอยู่ใน Cluster เดียวกันได้เลย เนื่องจากทั้ง 2 รายการไม่มีข้อมูล (Attribute value) ที่สามารถเชื่อมโยงถึงกันได้

Clustering algorithm ที่นำเสนอนี้มีชื่อว่า **RO**bust **C**lustering using **LinK**s : ROCK ซึ่งเป็น Algorithm ประเภท Hierarchical ทำงานแบบ Agglomerative

2.4.1 หลักการของ ROCK

คำศัพท์ที่เกี่ยวข้อง

2.4.1.1 Neighbors

ข้อมูลแต่ละเรคอร์ดในฐานข้อมูลสามารถแทนค่าด้วยจุดจุดหนึ่ง เรียกว่า Point โดย Neighbors คือกลุ่มของ point ที่มีความเหมือนหรือใกล้เคียงกัน ซึ่งวัดจากค่า Similarity : $Sim(p_i, p_j)$

ค่า Similarity นี้จะใช้หาค่าความใกล้กันของ point คู่ต่างๆ คือ p_i และ p_j นั่นเอง โดยจะมีค่าอยู่ระหว่าง 0 ถึง 1 โดยยิ่งเป็นค่าที่เบนเข้าใกล้ 1 มากเท่าใด แสดงว่า point คู่นั้นมีความเหมือนกันมากเท่านั้น ส่วนค่า Sim ที่เป็น 0 แสดงว่า point คู่นั้นไม่มีความเหมือนหรือไม่มีความสัมพันธ์กันเลย

การพิจารณาว่าแต่ละ point มีความเหมือนกันหรือไม่นั้นจะพิจารณาจากค่า Sim ที่ต้องมีค่ามากกว่าค่าหนึ่งที่กำหนดขึ้นมา (Threshold) ดังนั้นนิยามของ Neighbor จึงมีค่าเท่ากับ

$SIM(p_i p_j) >= \theta$ (θ คือค่า Threshold ที่กำหนดโดย user)

รูปที่ 2.5 แสดงนิยามของ neighbor

Neighbors ใดๆ ที่มีค่า Similarity มากกว่า θ เรียกว่าเรคอร์ดที่พิจารณานั้นเป็น "Common neighbors" กัน

กำหนดให้ T1 และ T2 คือรายการทางธุรกิจที่อยู่ในฐานข้อมูล ฟังก์ชั่นที่ใช้หาค่า Similarity นั้นมีหลายฟังก์ชั่น แต่ในภาพรวมนั้นจะทำงานเหมือนกันคือ

$$SIM(T1, T2) = \frac{|T1 \cap T2|}{|T1 \cup T2|}$$

รูปที่ 2.6 แสดงฟังก์ชั่นที่ใช้หาค่า Similarity ระหว่าง Point

โดย T1 คือจำนวน item ในข้อมูลรายการ T1 ยิ่งจำนวน item ของรายการ T1 และ T2 สามารถเชื่อมโยงถึงกันได้มากเท่าไรหรือมีค่ามากเท่าไร ($T1 \cap T2$) แสดงว่ารายการทั้งสองมีความเหมือนกันมากขึ้นเท่านั้น และมีโอกาสอยู่ใน Cluster เดียวกันมากขึ้นด้วย

การที่ใช้จำนวนข้อมูลในเซ็ตของเรคอร์ดที่กำลังพิจารณา ($T1 \cup T2$) เป็นตัวหารจะทำให้ค่า sim มีค่าได้ระหว่าง 0 ถึง 1 เท่านั้น ซึ่งสามารถนำไปเปรียบเทียบกับค่าความเหมือนกับรายการอื่นๆ ได้

2.4.1.2 Links

ในการพิจารณาว่ารายการต่างๆ จัดอยู่ในกลุ่ม Cluster เดียวกันหรือไม่ จะพิจารณาจากจำนวน Common neighbors ว่ามีค่า Link มากน้อยเพียงใด โดยยิ่งจำนวน Link ของ Common neighbors มีมาก โอกาสที่ point คู่นั้น จะอยู่ใน Cluster เดียวกันก็จะมีมากขึ้นตามไปด้วย

พิจารณาจากนิยามข้างต้นแสดงว่าเราจะใช้จำนวน Item ที่เหมือนกันระหว่าง point ที่กำลังพิจารณา (p_i และ p_j) ในการกำหนดความเหมือนกันของ point เหล่านั้น และแทนด้วยสัญลักษณ์ Link (p_i, p_j) โดยยิ่งค่า Link(p_i, p_j) มีค่ามากขึ้นเท่าไร โอกาสที่ p_i และ p_j จะอยู่ใน Cluster เดียวกันก็จะมีมากขึ้นเท่านั้น

2.4.1.3 Missing value

ข้อมูลประเภท Categorical ส่วนใหญ่จะมีลักษณะเป็น Fixed dimension คือมีจำนวน Attribute ที่คงที่ เช่นข้อมูลของลูกค้าหนึ่งๆ จะประกอบด้วยข้อมูล เพศ ที่อยู่ อาชีพ การศึกษา สถานะการแต่งงาน เป็นต้น ในขณะที่ข้อมูลของธุรกิจการค้าอาจไม่เป็น fixed dimension เช่น กรณีลูกค้าคนหนึ่งสามารถซื้อสินค้าได้หลายอย่าง ทำให้ในแต่ละเรคอร์ดจะมีจำนวน Item ไม่เท่ากัน

ประโยชน์ของข้อมูลแบบ Fixed dimension คือ ข้อมูลที่ขาดหายไป จะถูกแยกกลุ่มออกมาระหว่างการทำ Clustering เนื่องจาก missing value เหล่านี้จะถูกจัดเป็นข้อมูลที่ไม่มีความเหมือนกัน ทำให้ไม่เกิดการจัดกลุ่มข้อมูลแบบผิดๆ ได้ อย่างไรก็ตาม ทั้งนี้ก็ขึ้นอยู่กับปริมาณของ Missing value ด้วยว่ามีมากน้อยเพียงใด เพราะหากมีมากไป กลุ่มข้อมูลที่ได้จากการ Clustering ก็อาจมีความสมเหตุสมผลน้อยลง หรือไม่ได้สารสนเทศที่ประโยชน์ในการนำไปใช้งาน

2.4.1.4 Criterion Function

กระบวนการทำ Clustering เป็นกระบวนการทำงานแบบ Iterative คือวนการทำงานไปเรื่อยๆ จนกระทั่งได้จำนวน Cluster ที่เหมาะสม

ค่าที่ใช้ในการวัดความเหมาะสมในการทำ Clustering นี้เรียกว่า "Criterion Function" ซึ่งจะแตกต่างกันไปตาม Algorithm แต่ละประเภท เช่น K-Means จะใช้ค่าระยะห่างคำนวณแบบ Euclidean distance ส่วนวิธีการแบบ ROCK จะพิจารณาที่ระดับความสัมพันธ์ Link(p_i, p_j) ของ point ต่างๆ ในฐานข้อมูลว่ามีความเหมือนกันมากน้อยเพียงใด โดยถือว่าแต่ละ point เป็นแต่ละ Cluster ที่แยกจากกัน ทำให้ได้สมการของ Criterion function ดังนี้

$$Et = \sum_{i=1}^{k} ni * \sum_{Pq,Pr \in Ci} \frac{link(Pq, Pr)}{ni^{1+2f(\theta)}}$$

โดย

C_i คือ Cluster ลำดับที่ i

P_i คือ Point ลำดับที่ i

n คือ จำนวนรายการภายใน Cluster หรือภายใน Point นั้นๆ

รูปที่ 2.7 แสดง Criterion Function ที่ใช้หาระดับความเหมือนกันของแต่ละ Point

เมื่อพิจารณาจากสมการข้างต้น พบว่าค่า $\sum_{i-1}\sum_{Pq,Pr \in Ci} link(Pq, Pr)$ แสดงถึงผลรวมของจำนวน link ระหว่าง point ที่กำลังพิจารณาเพื่อใช้หาค่าความเหมือนกันของ point ทั้งหมดใน Cluster i การคำนวณดังกล่าวยังไม่สามารถแยก point ที่มีจำนวน link น้อยๆ ออกจาก Cluster ทำให้อาจเกิดกรณีที่รายการต่างๆ รวมกันอยู่เป็น Cluster เดียวกันทั้งหมดได้ ดังนั้นแต่ละ Cluster จะถูก Normalize โดยการหารค่าข้างต้นด้วยจำนวน Cross-link ทั้งหมดที่คาดว่าจะเกิดขึ้นของ cluster C_i จากนั้นก็ถ่วงน้ำหนักด้วย จำนวน point (n_i) ใน Cluster C_i

จำนวน link ทั้งหมดที่คาดว่าจะเกิดขึ้น จะมีค่าเท่ากับ $n_i^{1+2f(\theta)}$ โดย $f(\theta)$ คือค่าเฉลี่ยความสมบูรณ์ของฐานข้อมูลที่แสดงถึงระดับความเหมือนกันของรายการทั้งหมดในฐานข้อมูล กล่าวคือถ้ามีความเหมือนกันมาก แสดงว่าเรคอร์ดต่างๆ สามารถเชื่อมโยงถึงกันได้ จำนวน Link ที่เป็นไปได้ก็จะลดลง ส่งผลให้ค่า Criterion สูงขึ้น

การคำนวณหาค่า $f(\theta)$ ที่เหมาะสมนั้นสามารถทำได้หลายวิธี แต่วิธีหนึ่งที่ง่ายและทำงานได้ดีคือการใช้ค่า $f(\theta)$ เท่ากับ $(1-\theta) / (1+\theta)$ กล่าวคือ กล่าวคือถ้ามีค่าเป็น 1 หมายถึงข้อมูลทั้งหมดในฐานข้อมูลมีความเหมือนกันทุกรายการ ทำให้ตัวหาร ($n_i^{f(\theta)}$) มีค่าเป็น n ค่า Cross link ที่เป็นไปได้ทั้งหมดก็จะเท่ากับจำนวน link ทั้งหมดนั่นเอง

หากค่าเป็น 0 หมายถึงข้อมูลในฐานข้อมูลส่วนใหญ่ไม่มีความเหมือนกันเลย ตัวหารจะมีค่าเป็น n^3 ค่า Criterion function ที่ได้จะมีค่าน้อยลง ทำให้ความเป็นไปได้ในการรวมรายการนี้เข้ามาใน Cluster นี้น้อยลงไป

ฟังก์ชั่นที่ใช้วัดค่าความเหมือนกันของแต่ละ Cluster นอกจากการใช้ค่า Criterion Function ดังที่ได้แสดงไว้ข้างต้นแล้วนั้น ยังมีอีกฟังก์ชั่นหนึ่งที่มีการทำงานในลักษณะเดียวกัน เรียกว่าค่า Goodness measure

ประเด็นหลักที่ถือเป็นข้อด้อยของ Algorithm ประเภท Hierarchical คือใช้เวลามากในการหา Link ระหว่าง point ทั้งหมดทั้งฐานข้อมูล โดยในกรณีที่แย่ที่สุดจะต้องทำการหาความสัมพันธ์กันถึง n^3 ครั้ง ดังนั้นในการคำนวณหาค่า Similarity จึงได้ปรับปรุงส่วนของตัวหาร โดยตัดคู่ความสัมพันธ์ที่ไม่จำเป็นทิ้งออกไป สุดท้ายจึงได้สูตรการหาค่า Similarity เป็นดังนี้

$$g(Ci, Cj) = \frac{link[Ci, Cj]}{(ni + nj)^{1+2f(\theta)} - ni^{1+2f(\theta)} - nj^{1+2f(\theta)}}$$

รูปที่ 2.8 แสดงสมการการคำนวณหาค่า Goodness measure เพื่อวัดระดับความเหมือนกันของแต่ละ Cluster

ค่า Goodness measure นี้พัฒนาขึ้นมาโดยการตัดคู่ความสัมพันธ์ที่ไม่จำเป็นทิ้งไป โดยจำนวน Cross link ทั้งหมดที่เกิดขึ้นเท่ากับ $(n_i + n_j)^{1+2f(\theta)}$ โดย n_i และ n_j คือจำนวนเรคอร์ดใน Cluster $_i$ และ Cluster j ตามลำดับ จากนั้นจึงทำการลบด้วยจำนวน Cross Link ที่ Link เข้าหาตัวเอง ซึ่งก็คือ $n_i^{1+2f(\theta)}$ และ $n_j^{1+2f(\theta)}$ นั่นเอง

ค่า Link (C_i, C_j) คือจำนวน Cross-link ระหว่าง Cluster C_i และ C_j ซึ่งหาได้จากผลรวมค่า Link ของทุกๆ point ใน Cluster i กับทุกๆ point ใน Cluster j สามารถแสดงด้วยสมการข้างล่างนี้

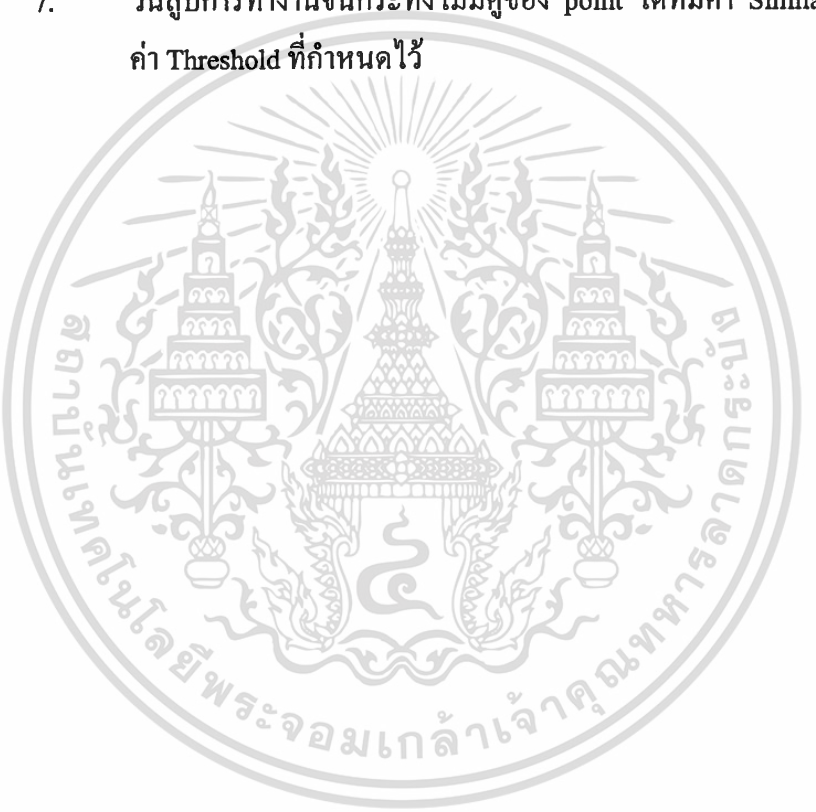
$$\sum_{Pq \in Ci, Pr \in Cj} link(Pq, Pr)$$

รูปที่ 2.9 แสดงการหาค่า Link ระหว่าง Cluster ใดๆ

2.4.1.5 การทำงานของ Algorithm ROCK

1. กำหนดค่า Threshold ของ Similarity (θ) เพื่อใช้วัดระดับความเหมือนของข้อมูลในแต่ละ Cluster
2. คำนวณหาค่า Link และค่า Similarity ของทุกๆ point ในฐานข้อมูล โดยใช้ Criterion Function ที่เลือก (ในที่นี้เลือกใช้ค่า Goodness measure)

3. เก็บค่า Similarity ที่มากที่สุดของแต่ละ point ไว้ในตารางแยกต่างหาก ชื่อ Global_Heap
4. พิจารณานำ point คู่ที่มีค่า Similarity มากที่สุดมาทำการ Merge เพื่อสร้างเป็น Cluster ใหม่
5. ปรับปรุงค่า Similarity ของ Cluster ใหม่นี้กับทุกๆ point ในฐานข้อมูล
6. ลบรายการของ point ที่ถูก merge แล้วออกจากฐานข้อมูล
7. วนลูปการทำงานจนกระทั่งไม่มีคู่ของ point ใดที่มีค่า Similarity สูงกว่าค่า Threshold ที่กำหนดไว้



บทที่ 3

การออกแบบและพัฒนาระบบงาน

ระบบงานที่พัฒนานี้มีการทำงานแบบ Client-Server โดย Client ทำหน้าที่ส่งค่าพารามิเตอร์ในการทำงานไปให้ Server ประมวลผลซึ่งได้เขียน Stored Procedure ฝังเอาไว้ ในที่นี้ Server ทำหน้าที่เป็น Database Server ซึ่งสามารถเป็นได้ทั้ง Local Server คือติดตั้งไว้ที่เครื่อง Client หรือเป็นเครื่อง Server โดยเฉพาะที่มีสมรรถนะสูง แล้วเรียกใช้งานผ่านเครือข่ายก็ได้

องค์ประกอบหลักของระบบงานจะประกอบด้วย 3 ส่วนได้แก่

1. Database

ทำหน้าที่เก็บข้อมูลซึ่งมี 2 ประเภท คือ ข้อมูลนำเข้า (ในกรณีนี้คือข้อมูลสินเชื่อ) และข้อมูลที่เกิดระหว่างการประมวลผลตาม Algorithm รวมถึงผลลัพธ์ที่ได้จากการประมวลผลด้วย

ข้อมูลเหล่านี้จะเก็บไว้ที่ Database Server ซึ่งใช้ SQL Server เป็น DBMS ทำหน้าที่จัดการ และประมวลผลคำสั่ง SQL ที่ส่งมาจาก Client

2. Programming module

ทำหน้าที่นำเข้าข้อมูล การทำ Transformation การทำ Data Cleaning และการClustering ตาม ROCK Algorithm โดย Client จะทำหน้าที่สร้างคำสั่งเพื่อกำหนดค่าพารามิเตอร์ต่างๆ ด้วยภาษา VBA และ Transact SQL ของ SQL Server 7.0 จากนั้นจึงส่งคำสั่งนี้ไปประมวลผลที่ Database server

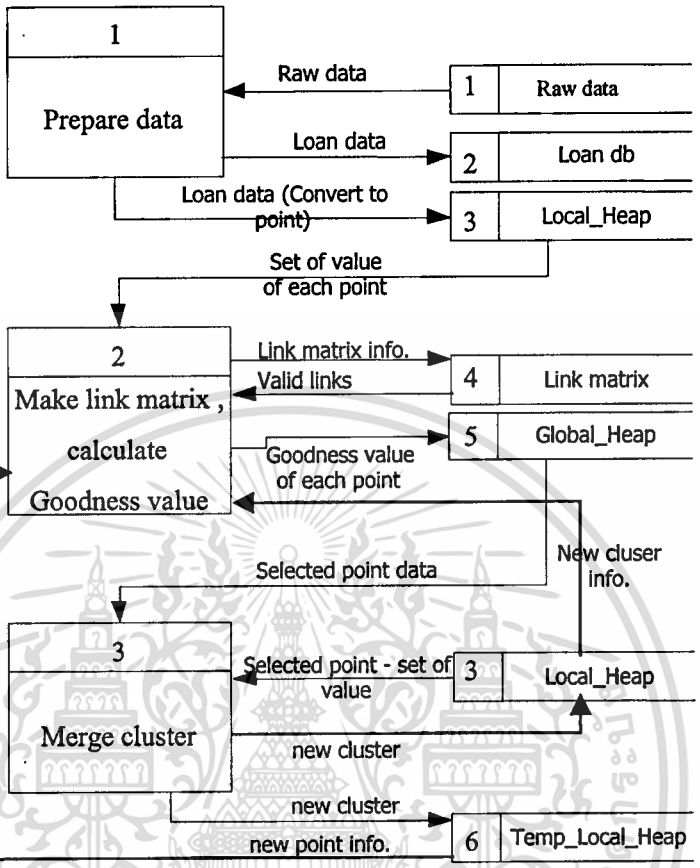
3. Reporting module

ทำหน้าที่สรุปผลลัพธ์ และจัดทำรายงานจากการ Clustering

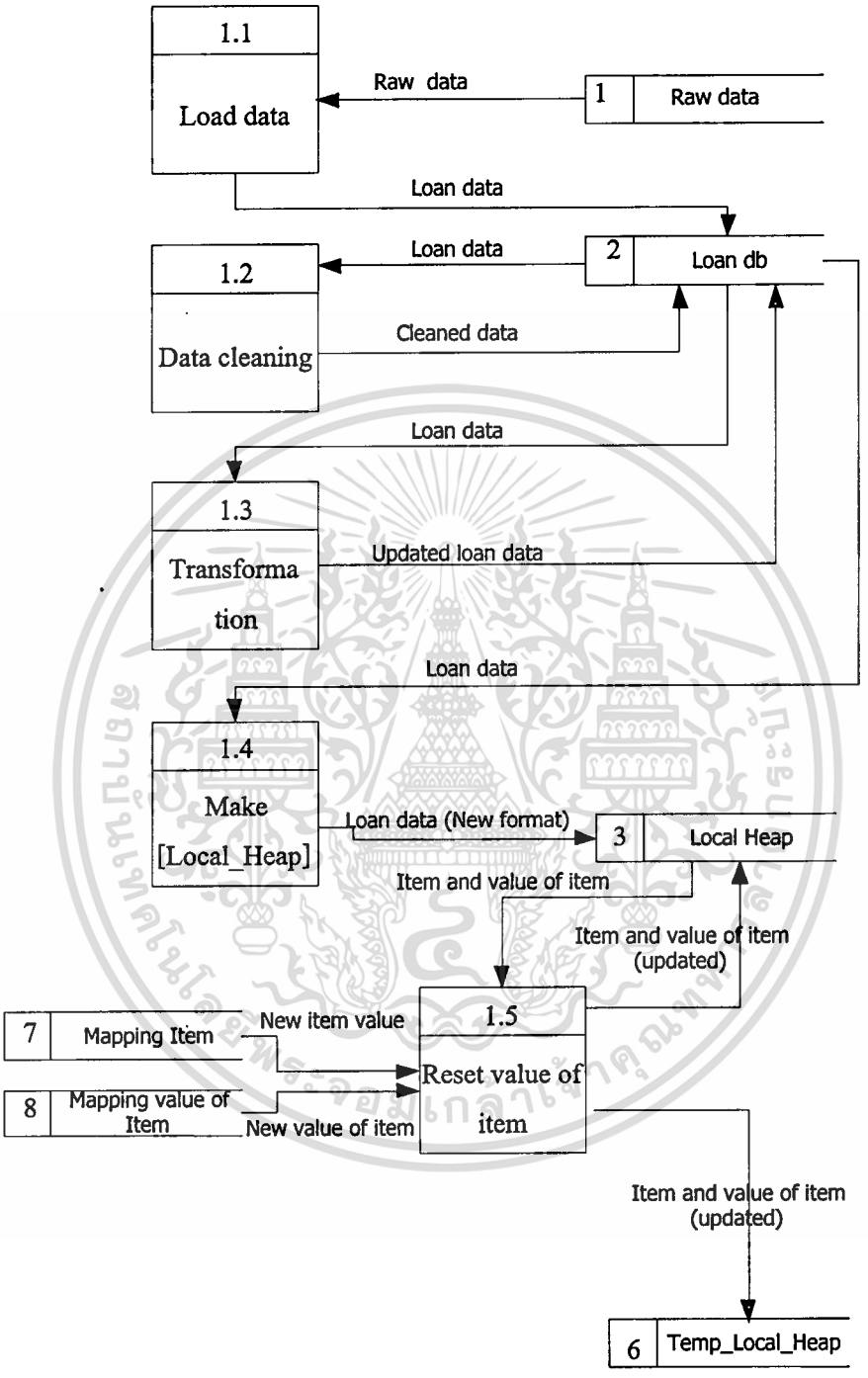
3.1 การออกแบบระบบงาน

ระบบงานนี้ได้ออกแบบให้มีลักษณะการทำงานแบ่งออกเป็น 2 ส่วนหลักๆ คือ ส่วนของการจัดเตรียมข้อมูล และส่วนของการประมวลผลตาม ROCK Algorithm เพื่อทำ Clustering ข้อมูลลูกค้าสินเชื่อ

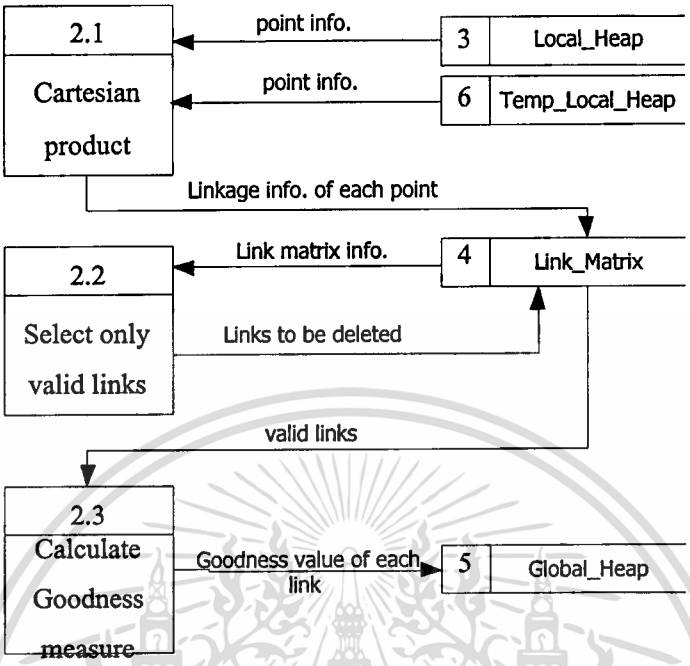
สามารถเขียนเป็น Context diagram และ Data flow diagram ได้ดังนี้



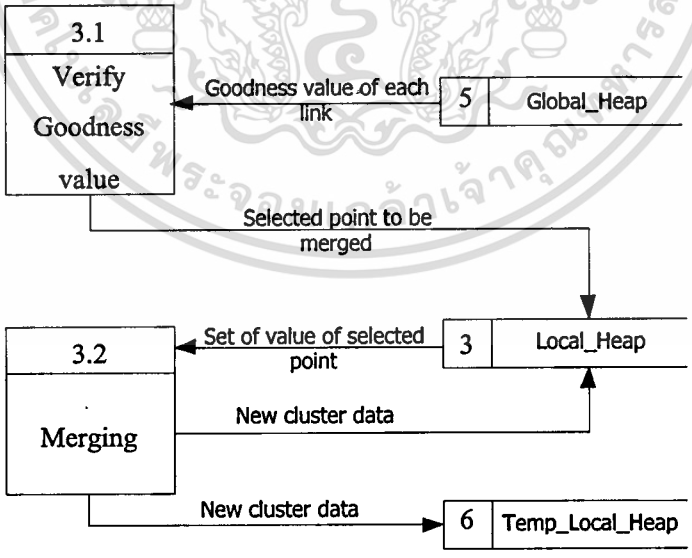
รูปที่ 3.1 แสดงขั้นตอนการทำงานทั้งหมดของระบบงาน



รูปที่ 3.2 แสดงรายละเอียดขั้นตอนการเตรียมข้อมูล



รูปที่ 3.3 แสดงรายละเอียดการคำนวณหาค่า Goodness



รูปที่ 3.4 แสดงรายละเอียดการ Merge cluster

3.2 การออกแบบฐานข้อมูล

ฐานข้อมูลของระบบงานนี้แบ่งเป็น 2 ประเภทตามหน้าที่งาน ได้แก่

3.2.1 ข้อมูลนำเข้า

ข้อมูลที่จะนำเข้าคือข้อมูลต้นทางจากระบบสินเชื่อของธนาคาร เป็นข้อมูลแสดงรายละเอียดต่างๆ ของลูกค้า ซึ่งจะถูกจัดกลุ่มโดย ROCK Algorithm แบ่งออกเป็น

3.2.1.1 ข้อมูลใบสมัคร แสดงรายละเอียดลูกค้าที่ระบุในใบสมัคร

3.2.1.2 ข้อมูลพฤติกรรมของลูกค้า แสดงจำนวนเงินและจำนวนวันที่ค้างชำระของลูกค้า

3.2.1.3 ข้อมูลลูกค้าสินเชื่อ คือตารางที่เกิดจากความสัมพันธ์ของข้อมูลใบสมัครและ ข้อมูลพฤติกรรมลูกค้า

อย่างไรก็ตามเพื่อให้ระบบงานนี้สามารถนำไปใช้กับข้อมูลประเภทอื่นได้ด้วย จึงไม่ได้ออกแบบตารางเพื่อเก็บข้อมูล 2 กลุ่มข้างต้น (ข้อมูลใบสมัครและข้อมูลพฤติกรรมของลูกค้า) ผู้ใช้งานต้องทำการจัดเตรียมข้อมูลให้เรียบร้อยเป็นตารางเดียว ซึ่งก็คือตารางข้อมูลลูกค้าสินเชื่อในข้อ 3.2.1.3 นั่นเอง โดยข้อมูลต้นทางจะอยู่ในรูปของ Text file เมื่อนำเข้าระบบแล้ว จึงสามารถใช้ระบบงานเพื่อทำการ Clean up ทำการ Transform ข้อมูล รวมถึงการสร้างฟิลด์และตารางใหม่ เพื่อเพิ่มประสิทธิภาพในการประมวลผล (รายละเอียดแสดงในส่วนของการทำงานของโปรแกรม)

3.2.2 ข้อมูลที่ใช้เก็บค่าการทำงานของ ROCK Algorithm

ข้อมูลส่วนใหญ่ที่สร้างขึ้นมาในกระบวนการทำงานนี้ มีวัตถุประสงค์หลักเพื่อแปลงข้อมูลต้นทางให้อยู่ในรูปของตัวเลขทางคณิตศาสตร์เพื่อความสะดวกในการประมวลผลด้วย ROCK Algorithm ดังนั้นรายละเอียดหน้าที่งานของตารางต่อไปนี้ ควรศึกษาควบคู่กับการทำงานของโปรแกรมจะทำให้เข้าใจหน้าที่งานของแต่ละตารางและการทำงานของโปรแกรมได้ดียิ่งขึ้น

การประมวลผลตามระบบงานนี้ จำเป็นต้องสร้างตารางเก็บข้อมูลเพิ่มขึ้นอีก 4 ประเภท ได้แก่

3.2.2.1 ตารางหลักสำหรับ Mapping ค่า Attribute ของแต่ละเรคอร์ด

ใช้ในการแปลงค่า Attribute ต่างๆ ให้เป็นตัวเลข เพื่อใช้หาค่า Link ของแต่ละ point ประกอบด้วย 2 ตารางสำหรับ Mapping Item (เช่น เพศ อาชีพ แปลงเป็น 1 และ 2 เป็นต้น) และ Value of Item (เช่น ชาย หญิง ครู วิศวกร แปลงเป็น 91, 92, 93, 94 เป็นต้น)

3.2.2.2 ตาราง Local Heap

ใช้เก็บค่า Attribute ของทุกๆ Point ในฐานข้อมูลหรือเก็บข้อมูลของ Cluster ใหม่ที่เกิดจากการ Clustering ข้อมูลในตารางนี้จะถูกนำไปหาค่า Link กับ point ต่างๆ แล้วนำไป Merge รวมกับ Cluster อื่นที่เหมาะสมต่อไป

3.2.2.3 ตาราง Temp Local Heap

มีรายละเอียดเช่นเดียวกับตาราง Local Heap ยกเว้นจะเก็บเฉพาะเรคอรด์ของ Cluster ใหม่ที่เกิดจากการ Merging เพื่อใช้ในการหาค่า Goodness measure ของทุกๆ point ที่มีต่อ Cluster ใหม่นี้ ส่วนผลลัพธ์จากการคำนวณค่า Goodness จะถูกจัดเก็บไว้ในตาราง Global Heap

3.2.2.4 ตาราง Global Heap

ใช้เก็บค่า Goodness ระหว่าง Point หรือระหว่าง Cluster ต่างๆ ในฐานข้อมูล

3.3 รายละเอียดของฐานข้อมูลแต่ละประเภท

เนื้อหาในส่วนนี้จะนำเสนอแนวทางและตัวอย่างข้อมูลที่นำมาใช้ในตารางแต่ละประเภท

ข้อมูลสินเชื่อลูกค้า (LoanDb)

ใช้สำหรับเก็บข้อมูลต้นทาง เป็นข้อมูลทางธุรกิจที่สนใจเพื่อใช้จัดกลุ่มลูกค้าแต่ละประเภท ชื่อฟิลด์และประเภทข้อมูลในตารางนี้สามารถเปลี่ยนแปลงได้ตามลักษณะของข้อมูลธุรกิจแต่ละประเภทที่นำเข้ามาทำ Clustering โดยความยาวและจำนวนฟิลด์นั้นจะขึ้นอยู่กับความสามารถของ SQL Server ว่ารองรับได้มากน้อยเพียงใด

ฟิลด์	ประเภทข้อมูล	ความหมาย
ID (ต้องเป็น Primary Key)	Numeric (Auto Number)	เลขที่ใบสมัคร
Gender	Text	เพศ
Age	Numeric (float)	อายุ
Occupation	Text	อาชีพ
Position	Text	ตำแหน่งงานปัจจุบัน
ResidentialProvince	Text	จังหวัดที่อยู่อาศัย
Income	Numeric (float)	รายได้ต่อเดือน
Limit	Numeric (float)	วงเงินที่ได้รับอนุมัติ
DaysPastDue	Numeric (float)	จำนวนวันค้างชำระ

3.3.2 ข้อมูลที่ใช้เก็บค่าการทำงานของ ROCK Algorithm

ตาราง Mapping Item

ฟิลด์	ประเภทข้อมูล	ความหมาย
Item_Original	Text	ชื่อ Item ทั้งหมดของฐานข้อมูล เช่น อายุ เพศ อาชีพ เป็นต้น
Item_New	Numeric (Integer)	ค่าใหม่ที่กำหนดให้เป็นตัวเลข (โปรแกรมจะสร้างให้โดยอัตโนมัติ)

ตาราง Mapping Value of Item

ฟิลด์	ประเภทข้อมูล	ความหมาย
Value_Original	Text	ค่าที่เป็นไปได้ทั้งหมดของแต่ละ Item เช่น ชาย หญิง เป็นต้น
Value_New	Numeric (Integer)	ค่าใหม่ที่กำหนดให้เป็นตัวเลข (โปรแกรมจะสร้างให้โดยอัตโนมัติ)

ตาราง Local Heap และตาราง Temp Local Heap

ฟิลด์	ประเภทข้อมูล	ความหมาย
ID	Numeric (Auto Number)	เลขที่ใบสมัคร (point)
Item	Numeric (Integer)	ชื่อ Item ต่างๆ ของลูกค้า เช่น เพศ อายุ อาชีพ เป็นต้น
Value	Numeric (Integer)	ค่าของแต่ละ Item เช่น ชาย หญิง เป็นต้น

ตาราง Link_Matrix

ตารางนี้เกิดจากการทำ Cartesian product ของตาราง Local Heap และตาราง Temp Local Heap เพื่อแสดงจำนวน Cross Link ระหว่าง Point ทั้งหมดในฐานข้อมูล กับ cluster ใหม่ที่เกิดขึ้นระหว่างการ Clustering

ฟิลด์	ประเภทข้อมูล	ความหมาย
ID_Linking	Numeric (Integer)	เลขที่ใบสมัครที่ Link กับ point อื่น
ID_Linked	Numeric (Integer)	เลขที่ใบสมัครที่ถูก Link จาก linking point อื่น
Item_Linking	Numeric (Integer)	เก็บค่า Item ของ point ที่ Link ไปยัง point อื่นๆ
Item_Linked	Numeric (Integer)	เก็บค่า Item ของ point ที่ถูก Link
Value_Linking	Numeric (Integer)	เก็บค่า Value of item ของ point ที่ Link ไปยัง point อื่นๆ
Value_Linked	Numeric (Integer)	เก็บค่า Value of item ของ point ที่ถูก Link
Diff_ID	Numeric (Integer)	ผลลัพธ์จากการนำ ID ของแต่ละ point ที่ link กันมาลบกัน
Diff_Item	Numeric (Integer)	ผลลัพธ์จากการนำ Item ของแต่ละ point ที่ link กันมาลบกัน
Diff_Value	Numeric (Integer)	ผลลัพธ์จากการนำ Value of Item ของแต่ละ point ที่ link กันมาลบกัน

ตาราง Global Heap

ฟิลด์	ประเภทข้อมูล	ความหมาย
ID_Linking	Numeric	เลขที่ใบสมัคร (Point) ที่ Link กับ point อื่น
ID_Linked	Numeric	เลขที่ใบสมัครที่ถูก Link จาก point อื่น
SimValue	Numeric	ค่า Goodness measure ที่ได้จากการคำนวณระหว่าง point ต่างๆ

3.4 รายละเอียดการทำงานของโปรแกรม

ระบบงานนี้จะแบ่งการทำงานออกเป็น 3 ส่วนหลักๆ คือส่วนของการจัดเตรียมข้อมูล ส่วนของการประมวลผลตาม ROCK Algorithm และส่วนของการจัดทำรายงาน โดยแต่ละส่วนมีรายละเอียดการทำงานดังนี้

3.4.1 การจัดเตรียมข้อมูล

3.4.1.1 การนำเข้าข้อมูล

3.4.1.2 การทำ Data cleaning

3.4.1.3 การทำ Data Transformation

3.4.1.4 การสร้างตารางเก็บค่าการทำงานของ ROCK Algorithm

3.4.2 การดำเนินการตาม ROCK Algorithm

3.4.2.1 การสร้างและ update Link ของแต่ละ point

3.4.2.2 การคำนวณหาค่า Goodness ระหว่าง point และ Cluster

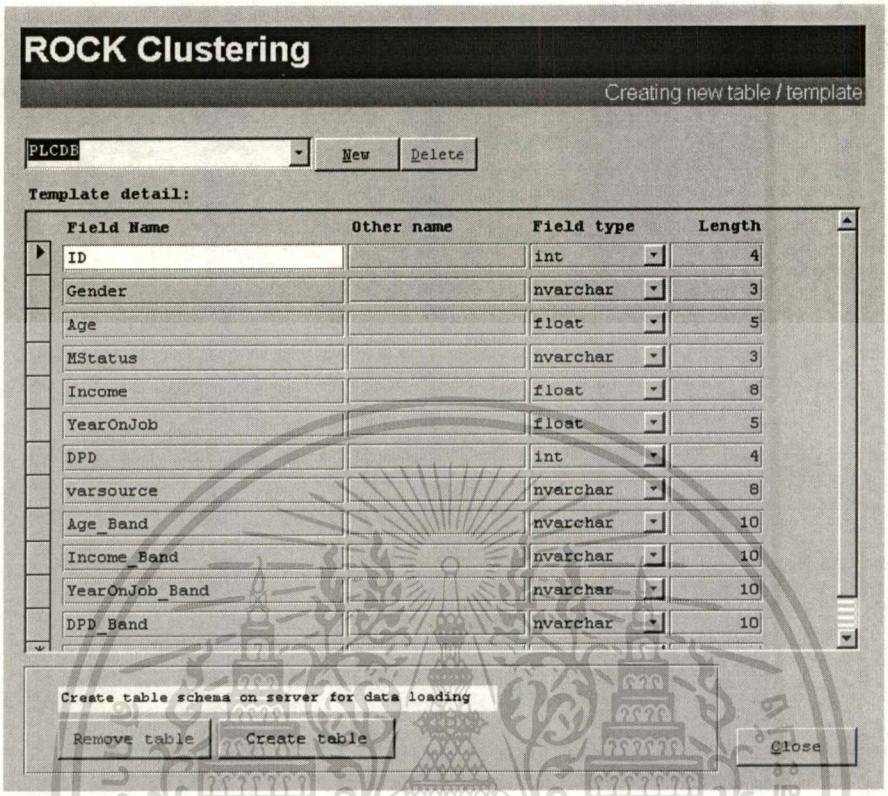
3.4.2.3 การรวม point เพื่อสร้าง Cluster ใหม่ (Merging)

3.4.3 การจัดทำรายงานและสรุปผล

3.4.1 การเตรียมข้อมูล

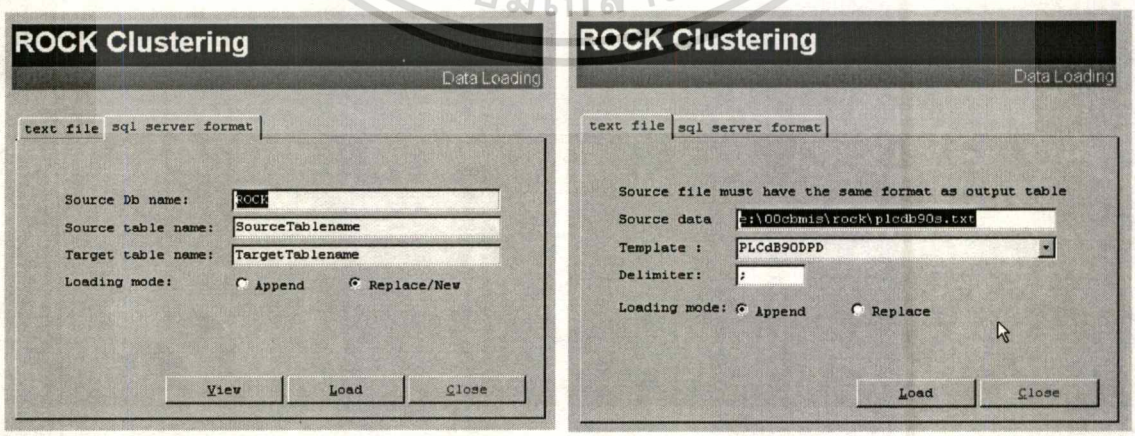
3.4.1.1 การนำเข้าข้อมูล

ข้อมูลนำเข้าจะอยู่ในรูปของ Text file เพียงแฟ้มข้อมูลเดียว (ผู้ใช้ต้องจัดเตรียมข้อมูลในรูปแบบนี้เอง) โดยแต่ละเรคอร์ด จะต้องมีฟิลด์ ID เพื่อใช้ในการอ้างอิงถึง point แต่ละ point ในตารางนี้ได้ ผู้ใช้สามารถกำหนดชื่อฟิลด์ ประเภทข้อมูลตามลักษณะข้อมูลที่นำมาใช้ได้เอง โดยระบบงานจะรองรับข้อมูลเพียง 2 ชนิดคือ ข้อมูลที่เป็นตัวเลขและข้อมูลตัวหนังสือ (Numeric และ Alphanumeric) ก่อนนำเข้าข้อมูลผู้ใช้งานต้องสร้าง Template ของตารางข้อมูลนำเข้าก่อน (ดังรูปที่ 3.5) ทั้งนี้จะทำให้ระบบงานสามารถนำไปใช้กับข้อมูลธุรกิจอื่นๆ ได้



รูปที่ 3.5 แสดงหน้าจอการสร้างตารางนำเข้าข้อมูล

สำหรับการนำเข้าข้อมูลจากฐานข้อมูล SQL Server7.0 โดยตรงนั้น ระบบสามารถทำการสร้างตารางนำเข้าข้อมูลข้างต้นให้โดยอัตโนมัติ ดังรูปที่ 3.6



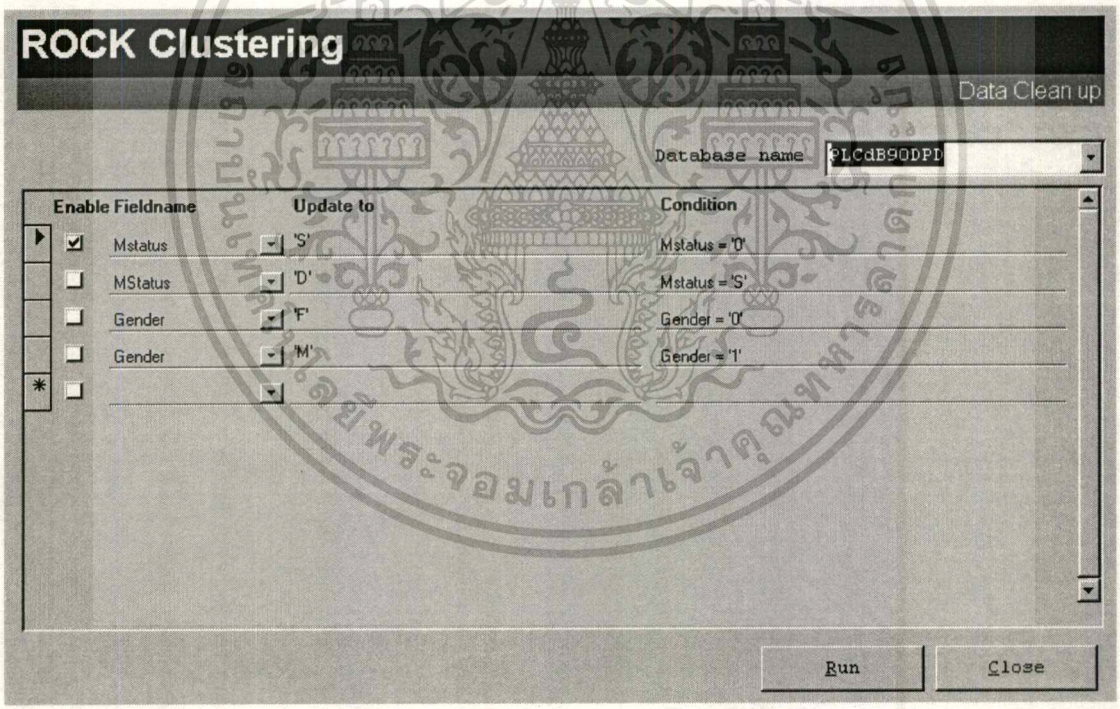
รูปที่ 3.6 แสดงหน้าจอการนำเข้าข้อมูลต้นทางที่เป็น Text file เข้ามายังตารางที่สร้างเตรียมไว้

3.4.1.2 การทำ Data Cleaning

การ Cleaning ข้อมูลมีวัตถุประสงค์หลักเพื่อกำจัด Outlier ออกจากฐานข้อมูล เพื่อให้ได้ข้อมูลที่มีค่าความถี่เพียงพอเทียบกับประชากรทั้งหมด ดังนั้น Outlier ในที่นี้จะหมายถึงข้อมูลที่บันทึกผิดพลาดและรวมถึงข้อมูลที่มีความถี่น้อยเกินไปต่อการสรุปผลด้วย

การ Cleaning จะมีลักษณะเป็นกึ่ง Manual คือการพิจารณาว่าข้อมูลใดที่มีความถี่น้อยและควรตัดออกจากการพิจารณา ขึ้นอยู่กับการตัดสินใจของผู้ใช้ โดยโปรแกรมจะทำการสรุปค่าความถี่ของแต่ละ Attribute และเปรียบเทียบเป็นเปอร์เซ็นต์กับประชากรรวม

การกำจัดข้อมูลที่เป็น Outlier จะใช้การกำหนดค่าใหม่ให้ เช่น กำหนดให้เป็น Other โดยผู้ใช้สามารถ Clean up ข้อมูลได้โดยอิสระ อย่างไรก็ตาม ผู้ใช้งานควรมีความรู้พื้นฐานเกี่ยวกับคำสั่ง SQL จึงจะสามารถใช้งานได้อย่างมีประสิทธิภาพ



รูปที่ 3.7 แสดงหน้าจอการแก้ไขค่าข้อมูลที่ผิดพลาด (Data cleaning)

3.4.1.3 การทำ Data Transformation

เมื่อกำจัดกลุ่มข้อมูล Outlier เรียบร้อยแล้ว ขั้นตอนต่อมาคือการแปลงข้อมูลตัวเลขที่มีค่าความต่อเนื่องให้เป็นข้อมูลแบบ Categorical และแบ่งเป็นช่วงๆ (เช่น อายุ รายได้ อายุการทำงาน วงเงินที่ได้รับ และยอดเงินค้างชำระ) เรียกว่าการทำ Discretization ขั้นตอนนี้เป็นสิ่งที่จำเป็น

เนื่องจาก ROCK Algorithm เหมาะสมสำหรับข้อมูลประเภท Categorical โดยพิจารณาค่าความเหมือนกันของแต่ละ point จากค่า Link ของแต่ละ point ดังนั้นข้อมูลประเภทมีค่าความต่อเนื่องมากจะมีโอกาสน้อยที่จะ Link กันได้

ตัวอย่างการทำ Data Discretization

อายุ

ข้อมูลเดิม	ข้อมูลใหม่
20	20-25
20.34	20-25
26.01	26-30
65	60 Up

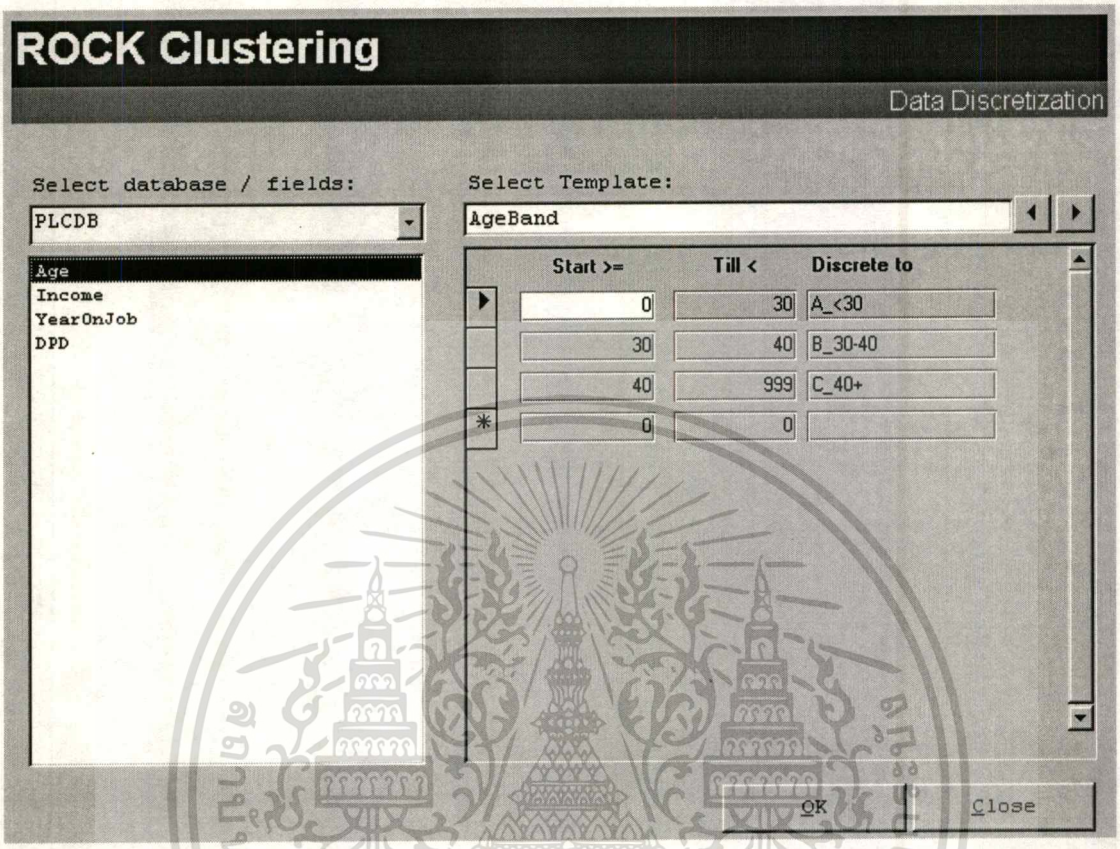
ตารางที่ 3.1 แสดงการทำ Discretization ข้อมูลอายุลูกค้า

รายได้ต่อเดือน

ข้อมูลเดิม	ข้อมูลใหม่
18,000	Less than 20,000
25,600	20001-30000
50,000	50001-60000
65,000	60001-65000
150,000	100000 Up

ตารางที่ 3.2 แสดงการทำ Discretization ข้อมูลรายได้ต่อเดือนของลูกค้า

การทำ Data Discretization ในระบบงานนี้ ผู้ใช้สามารถกำหนดช่วงของข้อมูลแต่ละรายการได้เอง หลังจากทำ Discretization แล้ว สามารถดูผลลัพธ์และกลับมาแก้ไขช่วงข้อมูลใหม่เพื่อให้ข้อมูลในแต่ละช่วงมีความถี่ในระดับที่ใกล้เคียงกันได้ ดังรูปข้างล่างนี้



รูปที่ 3.8 แสดงหน้าจอการทำ Data Transformation / Data Discretization ข้อมูลตัวเลขต่อเนื่อง

3.4.1.4 การสร้างตารางสำหรับเก็บค่าการทำงานของ ROCK Algorithm

ขั้นตอนสุดท้ายในส่วนของการจัดเตรียมข้อมูล คือการสร้างตาราง Local Heap และ Temp Local Heap ซึ่งเป็นตารางหลักที่ใช้ในการทำงานของ Algorithm โดยขั้นตอนนี้เป็นการทำงานภายในโปรแกรม และไม่ต้องติดต่อกับผู้ใช้

1) การสร้างตาราง Local Heap

ตารางนี้เกิดจากการหมุนตารางข้อมูลสินเชื่อลูกค้า (LoanDb) โดยจะแปลง Item ต่างๆ ซึ่งเป็นชื่อฟิลด์ของตารางข้อมูลหลัก ให้กลายเป็น Value ของ Item ใหม่ที่มีชื่อว่า “Item” ส่วนค่าของแต่ละฟิลด์เดิมนั้น ก็จะถูกหมุนให้กลายเป็น Value ของฟิลด์ใหม่ที่มีชื่อว่า “Value” ดังรูปข้างล่าง

Loan DB

ID	Sex	Occupation	Position	Province
1	Male	Engineer	Manager	BKK
2	Male	Engineer	Technician	BKK
3	Male	Accountant	Manager	BKK
4	Female	Dentist	Manager	BKK

Local Heap

ID	Item	Value
1	Sex	Male
1	Occupation	Engineer
1	Position	Manager
1	Province	BKK
2	Sex	Male
2	Occupation	Engineer
2	Position	Technician
2	Province	BKK
3	Sex	Male
3	Occupation	Accountant
3	Position	Manager
3	Province	BKK
4	Sex	Female
4	Occupation	Dentist
4	Position	Manager
4	Province	BKK

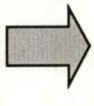
รูปที่ 3.9 แสดงการหมุนตารางข้อมูลต้นทางเป็นตาราง Local Heap

การสร้าง Cluster ใหม่ตาม ROCK Algorithm นั้น จะเกิดจากการทำ Join Operation ของ Item ต่างๆ ของแต่ละ point เกิดเป็นเซ็ตข้อมูลใหม่ การใช้ตารางในลักษณะนี้จะช่วยให้การเพิ่มเซ็ตข้อมูลของ Cluster ใหม่ทำได้ง่าย รวมถึงสามารถสรุปเซ็ตข้อมูลของแต่ละ Cluster ได้ง่ายขึ้นด้วย ตัวอย่างเช่นนำ point 3 รวมกับ point 4 เพื่อสร้าง Cluster ใหม่ สุดท้ายจะได้ตารางใหม่ดังรูปข้างล่าง

Local Heap

ID	Item	Value
1	Sex	Male
1	Occupation	Engineer
1	Position	Manager
1	Province	BKK
2	Sex	Male
2	Occupation	Engineer
2	Position	Technician
2	Province	BKK
3	Sex	Male
3	Occupation	Accountant
3	Position	Manager
3	Province	BKK
4	Sex	Female
4	Occupation	Dentist
4	Position	Manager
4	Province	BKK

Before merge



Local Heap

ID	Item	Value
1	Sex	Male
1	Occupation	Engineer
1	Position	Manager
1	Province	BKK
2	Sex	Male
2	Occupation	Engineer
2	Position	Technician
2	Province	BKK
34	Sex	Male
34	Sex	Female
34	Occupation	Accountant
34	Occupation	Dentist
34	Position	Manager
34	Province	BKK

After merge

รูปที่ 3.10 แสดงรายละเอียดเซ็ตข้อมูลหลังจากการสร้าง Cluster ใหม่

จากตารางข้างบน Cluster 34 ประกอบด้วยกลุ่มลูกค้าทั้งเพศชายและหญิง มีอาชีพนักบัญชี และทันตแพทย์ โดยทั้งหมดมีตำแหน่งเป็นผู้จัดการ และอาศัยอยู่ในกรุงเทพฯ ทั้งหมด

2) การแปลงค่า (Value of Item) ของตาราง Local_Heap

หลังจากหมุนข้อมูลในตาราง LoanDB แล้ว ขั้นตอนต่อไปคือการแทนค่าข้อมูลในตาราง Local_Heap ให้เป็นตัวเลขทั้งหมด เพื่อให้สามารถนำไปคำนวณหา Link โดยใช้ฟังก์ชั่นทางคณิตศาสตร์ได้ ซึ่งจะช่วยให้การหาค่า Link ของทั้งฐานข้อมูลเร็วขึ้น

ในการแทนค่าข้อมูลนี้ จะแทนค่า 2 ฟิลด์คือ Item และฟิลด์ Value โดยค่าใช้ตาราง Mapping Item และ Mapping Value of Item สุดท้ายจะได้ผลลัพธ์ตามรูปข้างล่างนี้

Local Heap

ID	Item	Value
1	Sex	Male
1	Occupation	Engineer
1	Position	Manager
1	Province	BKK
2	Sex	Male
2	Occupation	Engineer
2	Position	Technician
2	Province	BKK
3	Sex	Male
3	Occupation	Accountant
3	Position	Manager
3	Province	BKK
4	Sex	Female
4	Occupation	Dentist
4	Position	Manager
4	Province	BKK

Local Heap

ID	Item	Value
1	91	101
1	92	121
1	93	131
1	94	141
2	91	101
2	92	121
2	93	132
2	94	141
3	91	101
3	92	122
3	93	131
3	94	141
4	91	102
4	92	123
4	93	131
4	94	141

รูปที่ 3.11 แสดงการแปลงค่า Item และ Value ในตาราง Local_Heap

3) การเตรียมข้อมูลสำหรับตาราง Temp_Local_Heap

ตารางนี้เป็นตารางชั่วคราวที่ใช้สำหรับเก็บ point ต่างๆ ของ Cluster ใหม่เพื่อนำไปสร้างความสัมพันธ์ของแต่ละ Point ใน Cluster นี้กับ point ต่างๆ ในตาราง Local_Heap ความสัมพันธ์นี้ก็คือ Link ระหว่าง point ต่างๆ นั่นเอง

เนื่องจากเมื่อเริ่มการ Clustering นั้น จะถือว่าแต่ละ point ก็คือแต่ละ cluster ดังนั้นขั้นตอนสุดท้ายในการเตรียมข้อมูลก่อนดำเนินการ Clustering ตาม ROCK Algorithm จะเป็นการ Copy ข้อมูลของตาราง Local_Heap มาเก็บไว้ในตารางนี้ เพื่อสร้าง Link ของทุกๆ point ในฐานข้อมูล และใช้คำนวณหาค่า Goodness ที่จะใช้ในการสร้างกลุ่ม Cluster ต่อไป

ในลูปต่อๆ ไป ตาราง Temp_Local_Heap นี้จะเก็บเฉพาะ Item และ Value ของ Cluster ใหม่ที่เกิดขึ้นมา เพื่อหาค่า Goodness ที่สัมพันธ์กับ Cluster ใหม่นี้เท่านั้น

3.4.2 การดำเนินการตาม Algorithm

การทำงานของ ROCK ในการ Clustering ข้อมูลเป็นการทำงานแบบ Iterative คือ วนลูปการทำงานไปเรื่อยๆ ตราบใดที่ค่า Goodness ของ Cluster ยังมีค่าอยู่ในเกณฑ์ที่กำหนดไว้ ดังนั้นหน้าที่งานหลักในส่วนนี้สามารถแบ่งย่อยได้เป็น 3 ส่วนหลักๆ ได้แก่

3.4.2.1 การสร้าง และ Update Link Information ของแต่ละ Point

3.4.2.2 การคำนวณหาค่า Goodness ระหว่าง point และ Cluster

34.2.3 การรวม point เพื่อสร้าง Cluster ใหม่ (Merging)

3.4.2.1 การสร้าง และ Update Link Information ของแต่ละ Point

โดยทั่วไป เราสามารถใช้ Join Operation ในภาษา SQL ในการหาค่า Link ของแต่ละ point ได้ แต่วิธีการนี้ค่อนข้างใช้เวลานาน เพราะต้องวนลูปไปเรื่อยๆ จนกระทั่งทุกๆ point ในฐานข้อมูลถูก Join กันครบ ดังแสดงไว้ในรูปข้างล่าง

กำหนดให้ n คือจำนวน point (จำนวนเรคอร์ด) ทั้งหมดของฐานข้อมูล

```
Set       i = 2
WHILE     i <= n
SELECT    Count(*) as Link
FROM      Local_Heap A  INNER JOIN (Select Item, Value From Local_Heap Where ID = i)  B
          On A.Item = B.Item and A.Value = B.Value
WHERE ID = 1
SET       i = i + 1
Loop
```

รูปที่ 3.12 แสดงตัวอย่างการใช้ Join ในการหา Link ของทุกๆ point ต่อ point ที่ 1

จะเห็นว่าการใช้ JOIN Operation ในการสร้าง Link นั้นไม่เหมาะกับการทำงานของ ROCK Algorithm โดยเฉพาะกับฐานข้อมูลขนาดใหญ่ เนื่องจาก Operation นี้จำเป็นต้องใช้ขนาด

หน่วยความจำและเวลาในการประมวลผลนาน ดังนั้นในระบบงานนี้จะใช้วิธีการทำ Cartesian product operation ระหว่าง point ต่างๆ ในตาราง Local_Heap และ Temp_Local_Heap แทน

เมื่อทำ Cartesian product แล้ว จะทำให้ทุกๆ Item และทุกๆ Value เกิดการ Link กันในทุกทิศทาง ผลลัพธ์จากการทำ Cartesian Product จะเก็บไว้ในตาราง Link_Matrix ซึ่งต้องมีขั้นตอนการกำจัด Link ที่ไม่ถูกต้องอันเกิดจากการ Link ของ Item คนละประเภทกันออกไป เช่น Link ของ Item "Sex" กับ Item "Occupation" เป็นต้น

SEQ	ID_Linking	ID_Linked	Item_Linking	Item_Linked	Value_Linking	Value_Linked	DIFF_ID	DIFF_Item	DIFF_Value
1	1	3	91	93	101	131	-2	-2	-30
2	1	3	92	93	121	131	-2	-1	-10
3	1	3	93	93	131	131	-2	0	0
4	1	3	94	93	141	131	-2	1	10
5	2	3	91	93	101	131	-1	-2	-30
6	2	3	92	93	121	131	-1	-1	-10
7	2	3	93	93	132	131	-1	0	1
8	2	3	94	93	141	131	-1	1	10
9	3	3	91	93	101	131	0	-2	-30
10	3	3	92	93	122	131	0	-1	-9
11	3	3	93	93	131	131	0	0	0
12	3	3	94	93	141	131	0	1	10
13	4	3	91	93	102	131	1	-2	-29
14	4	3	92	93	123	131	1	-1	-8
15	4	3	93	93	131	131	1	0	0
16	4	3	94	93	141	131	1	1	10

3	1	3	93	93	131	131	-2	0	0
7	2	3	93	93	132	131	-1	0	1
11	3	3	93	93	131	131	0	0	0
15	4	3	93	93	131	131	1	0	0

ตารางที่ 3.3 แสดงผลลัพธ์จากการทำ Cartesian Product

Link ที่ไม่ถูกต้องเหล่านี้หาได้จากการนำ Value ของ Item ของแต่ละ Point มาลบกัน ถ้าได้ค่าเป็น 0 แสดงว่าเป็นการ Link ของ Item ประเภทเดียวกัน ส่วนเรคอร์ดที่มีค่าไม่เท่ากับ 0 แสดงว่าเป็นการ Link ของ Item คนละประเภทกัน ถือว่าเป็น Link ที่ไม่ถูกต้อง

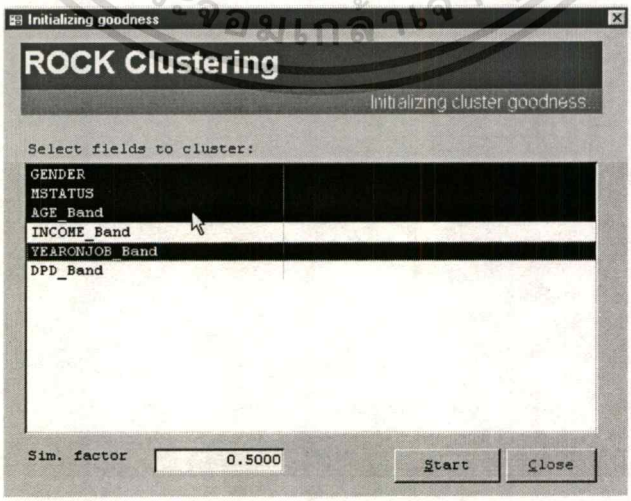
พิจารณาจากตารางที่ 3.3 เรคอร์ดที่ 1-4 เป็นการ Link ระหว่าง point ที่ 1 กับ 3 ระหว่าง Item Sex (91), Occupation (92), Position (93) และ Province (94) ของ Point 1 กับ Item Position (93) ของ point ที่ 3

จาก Matrix ที่ได้พบว่า เรคอร์ดที่ 1, 2 และ 4 เป็นการ Link ที่ไม่มีความหมายเพราะเกิดจากItem คนละประเภทกัน พิจารณาได้จากค่าฟิลด์ Diff_Item ที่มีค่าไม่เท่ากับ 0 โดยเรคอร์ดเหล่านี้จะถูกตัดทิ้ง

ค่า Diff_ID แสดงให้เห็นว่ามีการ Link เข้าหาตัวเองหรือไม่ แต่จะไม่ตัดทิ้งเพราะสามารถใช้คำนวณหาจำนวนเซ็ตข้อมูลของแต่ละ point ได้

ค่า Diff_Value แสดงให้เห็นว่า ถ้าเป็น Item ประเภทเดียวกัน Point ที่กำลังพิจารณาอยู่นั้นมีค่าตรงกันหรือไม่ ในที่นี้ เรคอร์ดที่ 3 เป็นการหา Link ของ Item Position ซึ่งทั้ง point 1 และ point 3 มีตำแหน่งงานเหมือนกัน คือ 131 (Manager) ซึ่งมีค่า Diff_value เท่ากับ 0

ในการหา Link ระหว่าง point นั้น ถ้าจำนวน item ที่ต้องการนำมาพิจารณามีมากเท่าใดก็ต้องใช้เนื้อที่ในการจัดเก็บข้อมูลที่จะเกิดจากการ Cross-Link มากขึ้นตามลำดับ ส่งผลให้ต้องใช้เวลาในการประมวลผลนานขึ้นด้วย อย่างไรก็ตามในระบบงานนี้ ผู้ใช้สามารถเลือกข้อมูลบาง Item มาพิจารณาทำ Clustering ได้



รูปที่ 3.13 แสดงหน้าจอการเลือก field ที่ต้องการนำมา Clustering

3.4.2.2 การคำนวณหาค่า Goodness ระหว่าง point และ Cluster

เมื่อทำการคำนวณ Link ระหว่าง point ต่างๆ แล้วในขั้นตอนนี้จะทำการคำนวณหาค่า Goodness ระหว่าง point ต่างๆ จากข้อมูลในตาราง Link_Matrix ผลลัพธ์ที่ได้จะเก็บไว้ในตาราง Global_Heap ซึ่งจะนำไปใช้ในการพิจารณาหา Cluster ที่เหมาะสมในการ Merge ต่อไป

จากฟังก์ชั่นการคำนวณหา Goodness Measure ในรูปที่ 2.8 ตัวแปรที่ต้องหาได้แก่

1) ค่า Link ระหว่าง point / cluster

ค่านี้หาได้จากผลรวมของค่า Diff_Value ของแต่ละ Point ในตาราง Local Heap ที่มีค่าเป็น 0 ซึ่งมีความหมายว่า point ทั้ง 2 นั้นเป็น Item และ Value เดียวกัน หรือมี Link ต่อกันนั่นเอง โดยใช้คำสั่ง SQL ดังนี้

```
SELECT      ID_Linking, IDLinked, Count(*) as numberOfLinks
FROM        [Link_Matrix]
WHERE       Diff_Value = 0
GROUP BY    ID_Linking, IDLinked
```

รูปที่ 3.14 แสดงคำสั่ง SQL ที่ใช้คำนวณหาค่า Link ระหว่าง point 1, 2, 3 และ 4

ผลลัพธ์ที่ได้จะมีลักษณะดังตารางต่อไปนี้

ID_Linking	IDLinked	NumberOfLinks
1	1	4
1	2	3
1	3	3
1	4	2
2	1	3
2	2	4
2	3	2
2	4	1
3	1	3
3	2	2
3	3	4

ID_Linking	IDLinked	NumberOfLinks
3	4	2
4	1	2
4	2	1
4	3	2
4	4	4

ตารางที่ 3.4 แสดงจำนวน Link ของแต่ละ point ในฐานข้อมูล

จากตารางที่ 3.4 เมื่อพิจารณาแต่ละ point พบว่า

Point 1 มีจำนวน Link กับ Point 2, 3 และ 4 เป็น 3, 3 และ 2 ตามลำดับ

Point 2 มีจำนวน Link กับ Point 3 และ 4 เป็น 2 และ 1 ตามลำดับ

Point 3 มีจำนวน Link กับ Point 4 เป็น 2

2) จำนวน item ของแต่ละ point

การหาจำนวน item ของแต่ละ point จะใช้การคำนวณลักษณะเดียวกับการหาค่า Link แต่กรณีนี้จะใช้จากการนับจำนวนเรคอร์ดที่มีค่า Diff_Item และ Diff_ID เป็น 0 โดยไม่สนใจว่า Diff_Value จะมีค่าเป็น 0 หรือไม่

```
SELECT      ID_Linking, IDLinked, Count(*) as numberOfItems
FROM        [Link_Matrix]
WHERE       Diff_Item = 0 And Diff_ID = 0
GROUP BY    ID_Linking, IDLinked
```

รูปที่ 3.15 แสดงคำสั่ง SQL ที่ใช้คำนวณหาจำนวน Item ของแต่ละ point

ผลลัพธ์ที่ได้จะมีลักษณะดังตารางต่อไปนี้

ID_Linking	numberOfItems
1	4
2	4

ID_Linking	numberOfItems
3	4
4	4

ตารางที่ 3.5 แสดงจำนวน Item ของแต่ละ point ในฐานข้อมูล

3) ค่าคงที่ในการกำหนดค่า Cross-link ที่เป็นไปได้

ค่านี้เป็นค่าคงที่ที่ใช้แทนค่าเฉลี่ยความเหมือนของฐานข้อมูลว่าแต่ละรายการมีระดับความเหมือนโดยเฉลี่ยมากน้อยเพียงใด โดยมีค่าระหว่าง 0-1

เมื่อหาตัวแปรทั้งสามได้แล้ว สุดท้ายจะเป็นการคำนวณค่า Goodness ตามสูตรแล้วเก็บผลลัพธ์ที่ได้ไว้ในตาราง Global_Heap

3.6.3 การรวม point เพื่อสร้าง Cluster ใหม่ (Merging)

1. พิจารณาค่า Goodness ของคู่ point จากตาราง Global_Heap ที่มีค่ามากที่สุดที่มากกว่าค่า Threshold ที่กำหนดไว้ จากนั้นจึงทำการ Merge point ของ Cluster คู่นั้น
2. ทำการ Merge Point คู่นั้น แล้วสร้าง Cluster ใหม่ขึ้นมา ตัวอย่างเช่น ต้องการสร้าง Cluster ระหว่าง point 3 และ point 4 (สมมติว่าค่า Goodness ระหว่าง point 3 และ 4 มีค่ามากที่สุด) โดยกำหนดเลขที่ของ Cluster ใหม่เป็น 999

```
SELECT    DISTINCT [999] as ClusterID, Item, Value
FROM      [Link_Matrix]
WHERE     ID in (3,4)
```

รูปที่ 3.16 แสดงคำสั่ง SQL ที่ใช้ในการ merge point ต่างๆ เข้าด้วยกัน

3. บันทึกข้อมูลของ Cluster ใหม่นี้เข้าไปในตาราง **[Local_Heap]** พร้อมทั้งลบเรคอร์ดที่ถูก Merge แล้วออกจากตารางนี้ด้วย (Link ของ Point 3 และ 4 นั่นเอง)
4. ทำการ Update ตาราง [Global_Heap] ใหม่โดยลบทุกเรคอร์ดที่ Link ไปยัง point ที่ถูก Merge
5. บันทึกข้อมูลของ Cluster ใหม่นี้เข้าไปในตาราง [Temp_Local_Heap]

6. วนลูปไปที่หน้าที่งาน "การคำนวณหาค่า Goodness ระหว่าง point และ Cluster" เพื่อคำนวณค่า Goodness ของทุก point กับ Cluster ใหม่นี้ แล้ว Update ค่า Goodness ในตาราง Global_Heap
7. การทำงานของหน้าที่งานนี้จะสิ้นสุดลงเมื่อไม่มีค่า Goodness ของ point คู่ใด ที่มากกว่าค่า Theshold ที่ตั้งไว้ ผลลัพธ์จากการทำงานจะเก็บไว้ในตาราง Local Heap

3.4.2.3 การจัดทำรายงานและสรุปผล

เมื่อดำเนินการ Clustering ตาม ROCK Algorithm จนเสร็จสมบูรณ์แล้ว จำนวนเรคอร์ดที่เหลือทั้งหมดในตาราง [Local_Heap] ก็คือจำนวน Cluster ที่ได้จากการดำเนินการนั่นเอง โดยหน้าที่งานในส่วนนี้เพียงทำการ Mapping ค่าต่างๆ กลับไปเป็นข้อมูลเดิม แล้วจัดรูปแบบรายงาน

ตัวอย่างข้อมูลที่ปรากฏในรายงานเป็นดังนี้

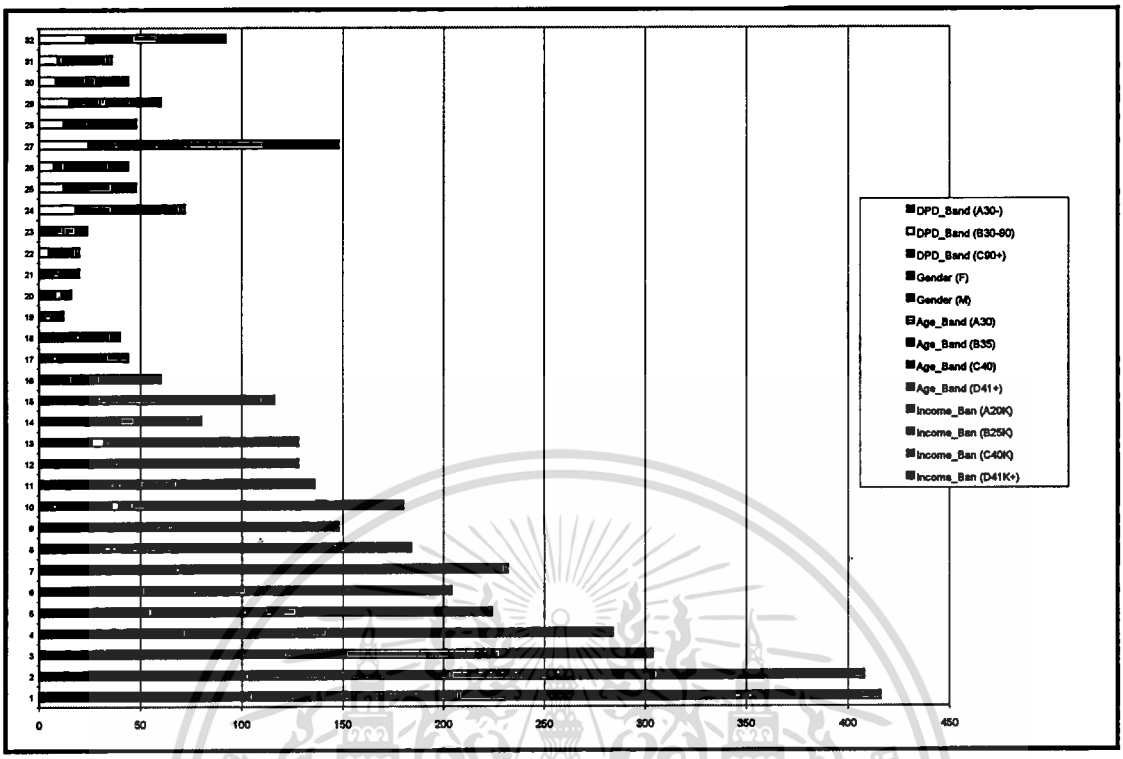
Seq	ClusterID	Sex	Occupation	Position	Province	DPD
1	1	{Male}	{Engineer}	{Manager}	{BKK}	30, 60
2	2	{Male}	{Engineer}	{Technician}	{BKK}	30
3	34	{Male, Female}	{Accountant, Dentist}	{Manager}	{BKK}	90,120,180

ตารางที่ 3.6 แสดงตัวอย่างผลลัพธ์จากการทำ clustering

Seq	ClusterID	Sex	Occupation	Position	Province	DPD
1	1	1000	1000	1000	1000	750, 250
2	2	2000	2000	2000	2000	2000
3	34	3000, 2500	3500, 2000	5500	5500	1000, 2000, 2500

ตารางที่ 3.7 แสดงตัวอย่างผลลัพธ์เป็นความถี่ข้อมูลของแต่ละ Cluster จากการทำ clustering

จากตัวอย่างผลลัพธ์ที่ได้ พบว่า ลูกค้ากลุ่มที่ 3 มีความเสี่ยงสูงสุด เพราะมีจำนวนลูกค้าที่ค้างชำระนานอยู่ในเกณฑ์ที่สูง ซึ่งประกอบด้วยกลุ่มอาชีพ นักบัญชี และทันตแพทย์ โดยมีตำแหน่ง ผู้จัดการ และอาศัยอยู่ในกรุงเทพฯ



รูปที่ 3.17 แสดงผลลัพธ์ในรูปของกราฟ จากการทดสอบ Clustering ข้อมูลสินเชื่อจำนวน 1000 เรคอร์ด

จากรูปที่ 3.17 แสดงให้เห็นถึงกลุ่มลูกค้าสินเชื่อจำนวน 32 กลุ่ม โดยส่วนใหญ่เป็นกลุ่มลูกค้าที่ดี คือมีระยะเวลาการค้างชำระหนี้ต่ำ ประกอบด้วยเพศชายและเพศหญิงในจำนวนที่ใกล้เคียงกัน แต่ใน 2 กลุ่มแรกที่ใหญ่ที่สุดนั้นส่วนใหญ่พบว่าเป็นเพศหญิง โดยกลุ่มแรกมีอายุอยู่ในช่วง30-41ปี ในขณะที่กลุ่มที่ 2 มีอายุอยู่ในช่วง 20-30 ปี

บทที่ 4
สรุปและข้อเสนอแนะ

การนำ Data Mining มาใช้จัดกลุ่มข้อมูลหรือการทำ Clustering นั้นจะทำให้กิจการทราบว่าความเสี่ยงที่เกิดจากลูกค้าแต่ละกลุ่มมีมากน้อยเพียงใด ทำให้สามารถปรับกลยุทธ์การดำเนินงานตลอดจนการกำหนดโปรแกรมส่งเสริมการขายให้ลูกค้าแต่ละกลุ่มได้อย่างเหมาะสม เช่น กลุ่มที่มีความเสี่ยงสูง ก็จำเป็นต้องกำหนดอัตราดอบเบี้ยเงินกู้ไว้สูงกว่ากลุ่มลูกค้าที่มีความเสี่ยงต่ำกว่า หรือลดคะแนนในการพิจารณาอนุมัติสินเชื่อสำหรับลูกค้ากลุ่มที่มีความเสี่ยงสูงเพื่อลดโอกาสในการรับลูกค้ากลุ่มนี้ให้อยู่ในระดับความเสี่ยงที่กิจการยอมรับได้

ข้อควรพิจารณาเพิ่มเติม

การทำงานตาม Algorithm นี้ จำเป็นต้องพื้นที่ในหน่วยความจำค่อนข้างมาก ในการเก็บค่า Link ทั้งหมดที่เป็นไปได้ของทุกๆ เรคอร์ดในฐานข้อมูล ยิ่งข้อมูลที่พิจารณามีจำนวน Item มากเท่าไร ก็จะเกิดจำนวน Link ที่เป็นไปได้มากขึ้นเท่านั้น ซึ่งก็ต้องใช้พื้นที่หน่วยความจำมากขึ้นตามไปด้วย

อย่างไรก็ตาม ปัญหานี้เป็นปัญหาทางด้านฮาร์ดแวร์ ซึ่งในอนาคตคาดว่าจะไม่ใช่ปัญหาหลักในการทำงานเพราะฮาร์ดแวร์จะมีสมรรถนะที่ดีขึ้น

นอกจากนี้ยังสามารถใช้เทคนิคการสุ่มตัวอย่างในการเลือกข้อมูลบางกลุ่มมาทำ Clustering แล้วสรุปผลลัพธ์เป็นตัวแทนของฐานข้อมูลนั้นได้

บรรณานุกรม

Anil, K.Jain and Richard, C.Dubes. 1998. **Algorithms for Clustering Data.** Prentice Hall. Englewood Cliffs: New Jersey.

Eric, Backer. 1995. **Computer-Assisted Reasoning in Cluster Analysis.** Prentice Hall. Englewood.

Kyuseok, Shim. et. al. 1998. **ROCK: A Robust Clustering Algorithm for Categorical Attributes.** [Online]

Sudipto, Guha. et. al. 1998. **A Clustering Algorithm for Categorical Attributes.** Bell Laboratories: Murray Hill.

ประวัติผู้เขียน

ชื่อ-นามสกุล	วรพงษ์ ริ้วเลิศธรรม
วันเดือนปีเกิด	4 มีนาคม 2517
ระดับการศึกษา	ปริญญาตรี บริหารธุรกิจบัณฑิต สาขา เทคโนโลยีสารสนเทศ มหาวิทยาลัยธรรมศาสตร์
การทำงาน	ตำแหน่งผู้ช่วยผู้จัดการ Programmer and Credit Analyst แผนกสินเชื่อส่วนบุคคล ธนาคารสแตนดาร์ดชาร์เตอร์ดนครธน