การพัฒนาเครื่องมือสำหรับไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บ

Development of Analysis Tool

for Mining Path Traversal Pattern In Web Environment

โดย

น.ส. อุบลพรรณ อุบลนุช

รหัส 42067034

อาจารย์ที่ปรึกษา

รศ. ดร. วิเชียร เปรมชัยสวัสดิ์

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2543

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาเครื่องมือสำหรับไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บ
นักศึกษา	นางสาวอุบลพรรณ อุบลนุช
อาจารย์ที่ปรึกษา	รศ. ดร. วิเชียร เปรมชัยสวัสดิ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

การไมนิ่งการใช้เว็บเป็นการค้นหารูปแบบการเข้าถึงของผู้ใช้จากเซิร์ฟเวอร์อย่างอัตโนมัติ องค์กรต่าง ๆ มีการจัดเก็บข้อมูลในการทำงานแต่ละวันเป็นจำนวนมาก ซึ่งข้อมูลเหล่านี้ได้ถูกสร้างขึ้นอย่างอัตโนมัติโดยเว็บเซิร์ฟเวอร์และจัดเก็บในล็อคการเข้าถึงของเซิร์ฟเวอร์ แหล่งของข้อมูลสารสนเทศแหล่งอื่นได้แก่ล็อคการอ้างอิงที่บรรจุข้อมูลสารสนเทศเกี่ยวกับการอ้างอิงเพจสำหรับแต่ละเพจ การวิเคราะห์ข้อมูลเหล่านี้ทำให้องค์กรสามารถกำหนดวางนโยบายและกลยุทธต่าง ๆ ได้มีประสิทธิผล รวมทั้งยังสามารถนำมาเป็นข้อมูลเพื่อใช้ในการปรับโครงสร้างของเว็บไซต์เพื่อสร้างการนำเสนอขององค์กรที่มีประสิทธิภาพมากขึ้น ดังนั้นเนื่องด้วยการมาบรรจบกันของคาต้าไมนิ่งและเทคโนโลยี www ทำให้ในปัจจุบันมีความต้องการที่จะค้นหาความรู้ที่ได้จากไฟล์ล็อคที่มีการจัดเก็บรวบรวมมาจากประวัติการเข้าถึงเพจ โดยใช้เทคนิคคาต้าไมนิ่งเข้ามาช่วยในการที่จะได้ความรู้นั้นมา โดยในโครงการพัฒนาระบบงานนี้จะศึกษาถึงการหารูปแบบเส้นทางการเดินทางภายในเว็บโดยสร้างเป็นเครื่องมือสำหรับไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บ ซึ่งผลที่ได้คือพฤติกรรมของผู้ใช้เว็บ

Title	Development of Analysis Tool for Mining Path Traversal Pattern In Web Environment
Student	Miss Ubonpan Ubonnut
Advisor	Assoc. Prof. Dr. Wichian Premchaiswadi
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

ABSTRACT

Web usage mining is the automatic discovery of user access patterns from Web server. Organizations collect large volumes of data in their daily operations, generated automatically by Web servers and collected in server access logs. Other sources of user information include referrer logs, which contain information about the referring pages for each page reference. Analyzing such data can help organizations determine the life time value of customers, cross marketing strategies across products, and effectiveness of promotional campaigns, among other thing. It can also provide information on how to restructure a Web site to create amore effective organizational presence. So, as a confluence of data mining and WWW technologies, it is now possible to perform data mining on web log records collected from the Internet web page access history. In this study, this project explores the problem of mining path traversal pattern and develops analysis tool. The behavior of the web page reader is imprinted in web log files.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานเรื่องการพัฒนาเครื่องมือไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บฉบับนี้ ผู้เขียนขอขอบพระคุณ รศ. ดร. วิเชียร เปรมชัยสวัสดิ์ อาจารย์ที่ปรึกษาที่ได้กรุณาให้คำปรึกษาและแนะนำ และขอขอบคุณผู้ที่เกี่ยวข้องทุก ๆ ท่านที่กรุณาให้คำปรึกษาและแนะนำวิธีการแก้ปัญหาต่าง ๆ ให้สามารถแก้ปัญหาให้ผ่านพ้นไปได้ ทำให้โครงการพัฒนาระบบงานนี้ได้สำเร็จลง

อุบลพรรณ อุบลนุช

กุมภาพันธ์ 2544

สารบัญ

หน้า

สารบัญตาราง

หน้า

ตารางที่

สารบัญรูป

หน้า

รูปที่

บทที่ 1
บทนำ

1.1 ความเป็นมา

เวิลด์ไวด์เว็บ (World Wide Web ,WWW) มีปริมาณการเจริญเติบโตด้วยอัตราที่รวดเร็วทั้งในเรื่องของปริมาณ ขนาดและความซับซ้อนของเว็บไซต์ ความซับซ้อนของงานเพิ่มขึ้นพร้อมกับการเจริญเติบโตของมัน เช่น การออกแบบเว็บไซต์ การออกแบบเว็บเซิร์ฟเวอร์ และความง่ายในการค้นหาผ่านเว็บไซต์ ข้อมูลเข้าที่สำคัญสำหรับการออกแบบงานเหล่านี้คือการวิเคราะห์ว่าเว็บไซต์ถูกใช้งานอย่างไร การวิเคราะห์การใช้งานเป็นไปได้ทั้งแบบสถิติตรงไปตรงมา เช่น ความถี่ในการเข้าถึงเพจ รวมทั้งรูปแบบของการวิเคราะห์ที่ซับซ้อนขึ้น เช่น การหาเส้นทางการเดินทางสามัญ (Common traversal path) ผ่านเว็บไซต์

เนื่องด้วยการมาบรรจบกันของคาต้าไมนิ่งและเทคโนโลยีเวิลด์ไวด์เว็บทำให้ในปัจจุบันมีความต้องการที่จะค้นหาความรู้ที่ได้จากไฟล์ล็อกที่มีการจัดเก็บรวบรวมมาจากประวัติการเข้าถึงเว็บเพจ โดยใช้เทคนิคคาต้าไมนิ่งเข้ามาช่วยในการที่จะได้ความรู้นั้นมา แอพพลิเคชันของเทคนิคคาต้าไมนิ่งที่ใช้กับเวิลด์ไวด์เว็บ ซึ่งก็คือการไมนิ่งเว็บ (Web mining) ได้รับความสนใจอย่างมากในงานวิจัยและสิ่งตีพิมพ์ทางการวิจัยในปัจจุบันอย่างมาก คำว่าเว็บไมนิ่งนั้นได้ถูกใช้ใน 2 แนวทาง แนวทางแรกคือการไมนิ่งเนื้อหาภายในเว็บ (Web content mining) ซึ่งเป็นกระบวนการค้นหาข้อมูลสารสนเทศจากเวิลด์ไวด์เว็บและในแนวทางที่สองคือการไมนิ่งการใช้งานเว็บ (Web usage mining) เป็นกระบวนการไมนิ่งเพื่อหาการค้นหาของผู้ใช้และรูปแบบการเข้าถึงเว็บ (Access pattern)

ในการศึกษานี้เราจะใช้ความสามารถของคาต้าไมนิ่งในการค้นหารูปแบบการเข้าถึงในสิ่งแวดล้อมที่มีการเตรียมข้อมูลสารสนเทศแบบกระจาย โดยที่บทความหรือออบเจ็กต์ถูกเชื่อมต่อเข้าด้วยกันเพื่อความสะดวกในการเข้าถึง ตัวอย่างเช่นในเวิลด์ไวด์เว็บและการบริการแบบออนไลน์ ซึ่งผู้ใช้ค้นหาสิ่งที่ต้องการโดยการเดินทางผ่านออบเจ็กต์ต่าง ๆ ที่ถูกเชื่อมต่อเข้าด้วยกัน ในการทำความเข้าใจรูปแบบการเข้าถึงของผู้ใช้ไม่เพียงแต่จะช่วยให้มีการแก้ไขออกแบบระบบได้ดีขึ้น (เช่น การเตรียมการเข้าถึงที่มีประสิทธิภาพระหว่างออบเจ็กต์ที่มีความสัมพันธ์กันสูง, การออกแบบเพจที่ดีขึ้น) แต่ยังสามารถนำไปสู่การตัดสินใจทางการตลาดได้ดีขึ้นด้วย (เช่น การใส่โฆษณาในที่ที่เหมาะสม , การแบ่งกลุ่มลูกค้าหรือผู้ใช้ และการวิเคราะห์พฤติกรรมของผู้ใช้) การดึงรูปแบบการเข้า

ถึงของผู้ใช้ในสิ่งแวดล้อมนี้ ในการศึกษาครั้งนี้จะหมายถึงการไมนิ่งรูปแบบการเดินทาง (Mining traversal pattern)

1.2 วัตถุประสงค์

1. พัฒนาเครื่องมือสำหรับหารูปแบบเส้นทางการเดินทางภายในเว็บ (Path traversal pattern) ซึ่งเป็นพฤติกรรมของผู้ใช้เว็บ
2. เพื่อเพิ่มความเข้าใจในการออกแบบและประยุกต์ใช้เทคโนโลยีเว็บไมนิ่ง
3. เพื่อเป็นแนวทางในการออกแบบและพัฒนาเครื่องมือสำหรับการทำเว็บไมนิ่งในรูปแบบอื่น ๆ ต่อไป
4. สามารถนำผลลัพธ์ที่ได้นำมาช่วยในการออกแบบระบบ หรือช่วยให้มีการตัดสินใจในทางการตลาดได้ดีขึ้น

1.3 แนวทางการศึกษา

1. ศึกษาแนวทางการนำดาต้าไมนิ่งมาประยุกต์ใช้กับการวิเคราะห์ข้อมูลที่ได้จากการเก็บประวัติการเข้าถึงเว็บ
2. กำหนดขอบเขตของการทำงาน
3. ศึกษาอัลกอริทึมในการทำเว็บไมนิ่ง วิเคราะห์ความเป็นไปได้และพิจารณาตัดสินใจเลือกอัลกอริทึมที่เหมาะสมในการทำงาน
4. พัฒนาโปรแกรมโดยใช้ Visual basic 6.0 , Microsoft SQL Server 7.0 และทดสอบ

1.4 ขอบเขตโครงการ

พัฒนาเครื่องมือสำหรับหารูปแบบเส้นทางการเดินทางภายในเว็บ โดยตัวโปรแกรมมีลักษณะดังนี้

1. จัดทำเครื่องมือสำหรับหารูปแบบเส้นทางการเดินทางภายในเว็บ
2. โปแกรมสามารถรับข้อมูลไฟล์ล็อกตามรูปแบบที่กำหนดไว้เท่านั้น
3. ผู้ใช้งานสามารถนำข้อมูลผลลัพธ์ที่ได้ไปวิเคราะห์เพื่อนำไปใช้ประโยชน์ต่อไปด้วยตนเอง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

เมื่อพัฒนาโปรแกรมสำเร็จลุล่วงตามเป้าหมายจะก่อให้เกิดประโยชน์ดังต่อไปนี้

1. ได้เครื่องมือสำหรับวิเคราะห์หารูปแบบเส้นทางการเดินภายภายในเว็บที่สามารถนำไปใช้งานได้จริง
2. สามารถนำเอาข้อมูลสารสนเทศที่มีอยู่แล้วในองค์กรมาใช้ประโยชน์ได้มากขึ้น
3. สามารถนำผลลัพธ์ที่ได้จากการวิเคราะห์มาเพิ่มประสิทธิภาพในการออกแบบระบบหรือวางนโยบายทางการตลาดได้ดีขึ้น

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ความรู้เบื้องต้น

เว็บไมนิ่ง (Web mining) สามารถจำกัดความอย่างกว้าง ๆ ได้ว่าเป็นการค้นหาและวิเคราะห์ข้อมูลสารสนเทศที่มีประโยชน์จาก WWW ซึ่งสามารถแบ่งการทำเว็บไมนิ่งได้เป็น 2 รูปแบบคือ การไมนิ่งเนื้อหาในเว็บ (Web content mining) คือการค้นหาทรัพยากรข้อมูลสารสนเทศอย่างอัตโนมัติแบบออนไลน์ และการไมนิ่งการใช้งานเว็บ (Web usage mining) คือการค้นหารูปแบบการเข้าถึงของผู้ใช้งานเว็บเซิร์ฟเวอร์

2.1.1 การไมนิ่งเนื้อหาในเว็บ

ข้อมูลสารสนเทศที่ได้จากเวิลด์ไวด์เว็บไม่มีโครงสร้างที่แน่นอนทำให้การค้นหาข้อมูลสารสนเทศที่ต้องการจากเว็บอย่างอัตโนมัติทำได้ยาก เครื่องมือค้นหา (Search engine) ที่มีในอดีตเช่น Lycos , Alta Vista , WebCrawler ,ALWEB , MetaCrawler และอื่น ๆ นั้นได้อำนวยความสะดวกให้กับผู้ใช้ในการค้นหา แต่ก็ไม่ได้เตรียมข้อมูลสารสนเทศที่เป็นโครงสร้างมากไปกว่าการจัดแบ่งกลุ่ม (Categorize) การกรอง (Filter) หรือการตีความเอกสาร ในการศึกษาในปัจจุบันได้มีการจัดเตรียมการสรุปและข้อมูลทางสถิติรวมทั้งมีการประเมินเปรียบเทียบกันระหว่างเครื่องมือค้นหาที่มีชื่อเสียงเกือบทั้งหมดไว้

ซึ่งจากปัจจัยที่ได้กล่าวมาทั้งหมดนี้ทำให้นักวิจัยมีความพร้อมที่จะพัฒนาเครื่องมือที่ฉลาดขึ้นได้เพื่อใช้ในการได้มาของข้อมูลสารสนเทศ เครื่องมือดังกล่าวเช่น ตัวแทนเว็บที่ฉลาด (Intelligent web agent) และเพื่อที่จะขยายเทคนิคคาต้าไมนิ่งให้ใช้กับการจัดการข้อมูลที่มีลักษณะกึ่งโครงสร้างที่มีอยู่ในเว็บ เราสามารถสรุปความพยายามที่จะทำในแต่ละวิธีได้ดังนี้

1. วิธีการพื้นฐานทางตัวแทน (Agent-based approach) สามารถแบ่งได้ 3 กลุ่มดังนี้
   - <u>ตัวแทนการค้นหาที่ฉลาด (Intelligent search agent)</u> ตัวแทนการค้นหาที่ฉลาดที่ได้พัฒนาขึ้นนั้นได้มีการค้นหาข้อมูลสารสนเทศที่ต้องการค้นโดยใช้คุณลักษณะของโดเมนและรูปแบบของผู้ใช้ (User profile) เพื่อจัดการและตีความข้อมูลสารสนเทศที่หาได้มา

- การกรองและจัดกลุ่มข้อมูลสารสนเทศ (Information filtering/categorization) มีตัวแทนเว็บหลายตัวที่ใช้เทคนิคการได้มาของข้อมูลสารสนเทศและคุณสมบัติการเปิดไฮเปอร์เทกซ์ของเอกสารเว็บมาใช้เพื่อดึง กรองและจัดกลุ่มข้อมูลสารสนเทศอย่างอัตโนมัติ
- ตัวแทนเว็บส่วนตัว (Personalized web agent) ตัวแทนเว็บในกลุ่มนี้จะเรียนรู้ความชอบของผู้ใช้และค้นหาแหล่งข้อมูลสารสนเทศโดยอยู่บนพื้นฐานของสิ่งที่ชอบเหล่านี้และปัจจัยอื่น ๆ ที่มีความสนใจโดยส่วนตัว

2. วิธีการทางฐานข้อมูล (Database approach)

วิธีการทางฐานข้อมูลของการไมนิ่งเว็บได้มุ่งความสนใจไปที่เทคนิคที่จะจัดการข้อมูลที่มีลักษณะกึ่งโครงสร้างที่มีอยู่ให้เว็บในมีโครงสร้างมากขึ้นโดยใช้กลไกการถาม (Query) ฐานข้อมูลที่เป็นมาตรฐานและใช้เทคนิคดาต้าไมนิ่งในการวิเคราะห์ผลลัพธ์ที่ได้มา

2.1.2 การไมนิ่งการใช้งานเว็บ

เวิลด์ไวด์เว็บมีการเจริญเติบโตด้วยอัตราที่น่าอัศจรรย์อย่างต่อเนื่องในทั้งปริมาณของการขนส่ง ขนาดและความซับซ้อนของเว็บไซต์ ความซับซ้อนของงานเช่น การออกแบบเว็บไซต์ การออกแบบเว็บเซิร์ฟเวอร์และความง่ายในการค้นหาผ่านเว็บไซต์ได้เพิ่มมากขึ้นพร้อมกับการเจริญเติบโตนี้ ข้อมูลเข้าที่สำคัญสำหรับงานในการออกแบบเหล่านี้คือการวิเคราะห์ว่าเว็บไซต์ถูกใช้อย่างไร การวิเคราะห์การใช้งานเป็นได้ทั้งทางสถิติอย่างตรงไปตรงมาเช่น ความถี่ในการเข้าถึงเพจ รวมถึงรูปแบบของการวิเคราะห์ที่ซับซ้อนขึ้นเช่น การหาเส้นทางการเดินทางสามัญผ่านเว็บไซต์ ข้อมูลสารสนเทศของการใช้งานสามารถถูกใช้ในการปรับโครงสร้างเว็บไซต์ใหม่เพื่อที่จะรองรับความต้องการของผู้ใช้ของเว็บไซต์ได้ดียิ่งขึ้น เส้นทางการเดินทางที่ผิดไปจากความเป็นจริงที่กำหนดไว้หรือเพจที่มีข้อมูลสารสนเทศที่สำคัญของไซต์ถูกใช้งานน้อยสามารถแสดงให้เห็นว่าลิงค์ของไซต์และข้อมูลสารสนเทศไม่ได้ถูกออกแบบที่ดี การออกแบบรูปแบบการแสดงข้อมูลหรือแบบแผนการกระจายของเว็บเซิร์ฟเวอร์สามารถถูกทำให้ดีขึ้นจากความรู้ที่ว่าผู้ใช้มีการเดินทางค้นหาผ่านเว็บไซต์อย่างไร ข้อมูลสารสนเทศของการใช้งานยังสามารถถูกใช้ในการช่วยเหลือการเดินทางค้นหาไซต์ได้โดยตรงโดยการเตรียมบัญชีรายชื่อของจุดปลายทางที่มีการเข้าถึงมากของเว็บเพจเฉพาะเจาะจง

ในปัจจุบันระบบและเทคนิคที่ซับซ้อนยิ่งขึ้นที่ใช้ในการค้นหาและวิเคราะห์รูปแบบได้มีการสร้างออกมา เครื่องมือเหล่านี้สามารถแบ่งออกได้เป็น 2 กลุ่มหลัก ๆ คือ

- เครื่องมือค้นหารูปแบบ (Pattern discovery tool)

- เครื่องมือวิเคราะห์รูปแบบ (Pattern analysis tool)

ในการศึกษาครั้งนี้จะศึกษาถึงการไมนิ่งการใช้งานเว็บ โดยมุ่งความสนใจที่เครื่องมือค้นหารูปแบบ

2.2 งานที่เกี่ยวข้อง

มีเครื่องมือสำหรับการวิเคราะห์เว็บเซิร์ฟเวอร์ล็อกทางการค้าหลายตัวซึ่งเตรียมกลไกที่มีข้อจำกัดสำหรับการรายงานกิจกรรมของผู้ใช้เช่น เครื่องมือนี้สามารถใช้หาจำนวนของการเข้าถึงในไฟล์หนึ่ง ๆ และจำนวนครั้งของการเข้าเยี่ยมชม อย่างไรก็ตามเครื่องมือเหล่านี้ไม่ได้ถูกออกแบบมาสำหรับเว็บเซิร์ฟเวอร์ที่มีอัตราการเข้าเยี่ยมชมสูงและโดยทั่วไปแล้วมีการเตรียมการวิเคราะห์ความสัมพันธ์ของข้อมูลระหว่างไฟล์ที่เข้าถึงเพียงเล็กน้อย ซึ่งเป็นจุดสำคัญในการนำใช้งานข้อมูลจากเซิร์ฟเวอร์ล็อกมาใช้งานให้ได้ประโยชน์อย่างเต็มที่ แนวคิดของการประยุกต์เทคนิคดาต้าไมนิ่งกับเว็บเซิร์ฟเวอร์ล็อกจึงได้ถูกเสนอขึ้นมา

2.3 โมเดลพฤติกรรมการค้นหา

ในบางแนวทางการไมนิ่งการใช้เว็บคือกระบวนการของการไกล่เกลี่ยมุมมองของผู้พัฒนาเว็บไซต์ว่าไซต์ควรจะถูกใช้ด้วยแนวทางที่ผู้ใช้ทำการค้นหาผ่านไซต์จริง ๆ ได้อย่างไร ดังนั้นจึงต้องการข้อมูลเข้าสำหรับกระบวนการไมนิ่งการใช้เว็บคือการตีความมุมมองพฤติกรรมของการค้นหาของผู้พัฒนาไซต์และการตีความพฤติกรรมของการค้นหาจริง ๆ ที่เกิดขึ้น ข้อมูลเข้าเหล่านี้หาได้จากไฟล์ของไซต์และล็อกของเซิร์ฟเวอร์ตามลำดับ

2.3.1 โมเดลของผู้พัฒนา

มุมมองของผู้พัฒนาเว็บไซต์มองว่าไซต์ควรจะถูกใช้ตามลำดับในโครงสร้างของไซต์เป็นอย่างไร แต่ละลิงค์ระหว่างเพจเกิดขึ้นเนื่องจากผู้พัฒนาเชื่อว่าเพจนั้น ๆ มีความสัมพันธ์กันอยู่ในบางแนวทาง ซึ่งเนื้อหาของเพจจะเป็นการเตรียมข้อมูลสารสนเทศเกี่ยวกับว่าผู้พัฒนาคาดหวังว่าไซต์จะถูกใช้อย่างไร ดังนั้นขั้นตอนโดยรวมของขั้นการเตรียมข้อมูลคือการแบ่งประเภทของไซต์เพจและการดึงลักษณะโครงสร้างของไซต์ออกจากไฟล์ HTML ที่ใช้ในการสร้างเว็บไซต์นั้นขึ้นมา ลักษณะโครงสร้างของเว็บไซต์สามารถหาอย่างง่าย ๆ โดยการหาไฮเปอร์เท็กซ์ลิงค์ทั้งหมดในแต่ละเพจที่สร้างจากไฟล์ HTML และจากนั้นทำการไล่ในแต่ละลิงค์จนกระทั่งทุก ๆ เพจของไซต์ถูกนำมาวางเป็นโครงสร้าง เพจถูกแยกประเภทโดยหลักมี 5 ประเภทดังนี้

- เพจหลัก (Head page) – เพจที่มีจุดประสงค์ให้เป็นเพจแรกที่ผู้ใช้จะเข้ามาเยี่ยมชม เช่น เพจ "home"
- เพจเนื้อหา (Content page) – เพจที่บรรจุส่วนของเนื้อหาข้อมูลสารสนเทศที่เว็บไซต์มีการจัดเตรียมไว้
- เพจนำทาง (Navigation page) – เพจที่มีจุดประสงค์คือเตรียมลิงค์เพื่อนำทางผู้ใช้ไปยังเพจเนื้อหา
- เพจค้นดู (Look-up page) – เพจที่ใช้ในการจัดเตรียมคำนิยามหรือการขยายของคำย่อ
- เพจส่วนตัว (Personal page) – เพจที่ใช้ในการแสดงข้อมูลสารสนเทศของประวัติบุคคลหรือลักษณะของบุคคลที่เกี่ยวข้องกับองค์กรที่ทำเว็บไซต์

แต่ละประเภทของเพจเหล่านี้ถูกคาดหวังให้แสดงคุณลักษณะทางกายภาพที่เห็นได้ชัดเจน ตัวอย่างเช่น เพจหลักถูกคาดหวังว่าต้องมีลิงค์ไปยังเพจอื่น ๆ มาก และบ่อยครั้งอยู่ที่หัว (root) ของโครงสร้างไฟล์ไซต์ ตารางที่ 2.1 แสดงคุณสมบัติทางกายภาพทั่วๆ ไปของแต่ละประเภทของเพจ แต่ตามความเป็นจริงแล้วสิ่งเหล่านี้เป็นเพียงแค่กฎตายตัวหยาบ ๆ และอาจจะมีเพจที่มีประเภทที่แน่นอนที่ไม่ได้เข้าชุดกับคุณสมบัติทางกายภาพนี้ เพจส่วนตัวไม่ได้ถูกคาดหวังให้แสดงคุณลักษณะทั่วไป แต่เพจส่วนตัวถูกคาดหวังให้เป็นที่หนึ่งของการรวมตัวของเพจประเภทอื่น ตัวอย่างเช่นเพจส่วนตัวบ่อยครั้งที่มีเนื้อหาในรูปแบบของข้อมูลสารสนเทศของประวัติบุคคลตามด้วยรายการของลิงค์ที่ชื่นชอบ ความแตกต่างที่สำคัญสำหรับเพจส่วนตัวคือเพจเหล่านี้ไม่ถูกควบคุมโดยผู้ออกแบบไซต์ และดังนั้นจึงไม่ถูกคาดหวังว่าจะเป็นตัวช่วยในการค้นหากฎ ที่เป็นเช่นนี้เพราะว่ามีความคาดหวังว่าการใช้งานของเพจส่วนตัวต่ำมากเมื่อเทียบกับการจราจรของไซต์ทั้งหมด ค่าเทรสโฮสต์เช่น ค่าสนับสนุนควรจะถูกคาดการณ์ว่าจะกรองกฎที่บรรจุเพจส่วนตัวออกไปได้ มีการรวมกันของเพจแต่ละประเภทที่สามารถนำมาประยุกต์กับเพจ ๆ หนึ่งได้ เช่น เพจหลัก-นำทาง หรือเพจเนื้อหา-นำทาง การแบ่งประเภทของเพจจะแสดงมุมมองของผู้พัฒนาเว็บไซต์ว่าแต่ละเพจจะถูกใช้งานอย่างไร การแบ่งแยกประเภทสามารถถูกกำหนดได้โดยที่ผู้ออกแบบไซต์ทำด้วยตนเอง หรือใช้แบบอัตโนมัติโดยใช้เทคนิคการเรียนรู้แบบมีผู้สอน (Supervised learning) เพื่อที่จะแบ่งประเภทของเพจไซต์ได้อย่างอัตโนมัติ คุณลักษณะทางกายภาพต้องสามารถถูกเรียนรู้โดยอัลกอริทึมการแบ่งประเภทเช่น C4.5 โดยใช้ชุดการฝึกสอนของเพจ วิธีที่น่าจะเป็นไปได้แบบอื่นเช่นมีการเพิ่มแทรกเทรกเพื่อบอกประเภทเข้าไปที่แต่ละเพจโดยผู้ออกแบบไซต์

2.3.2 โมเดลของผู้ใช้

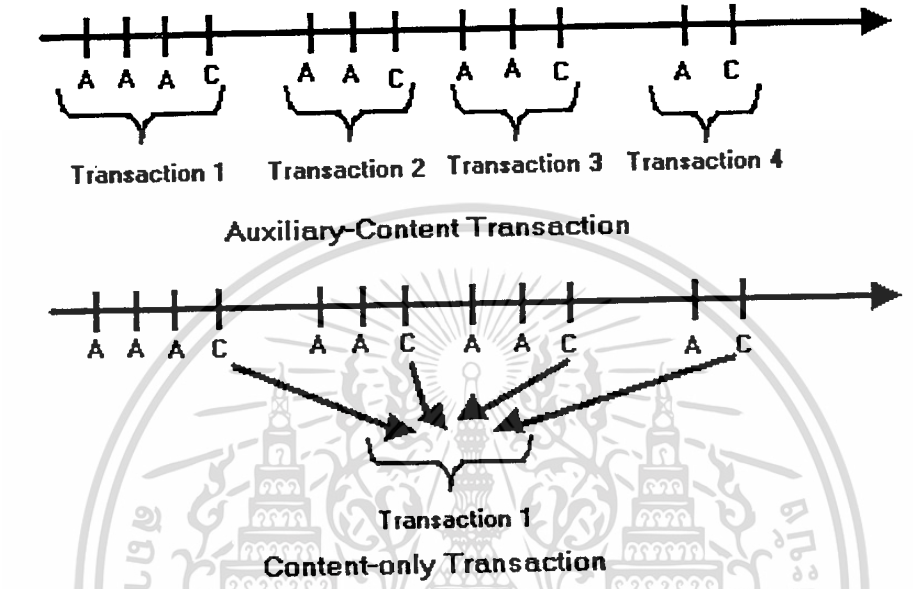
การเปรียบเทียบกันของแต่ละคุณลักษณะพิเศษทางกายภาพทั่ว ๆ ไปสำหรับประเภทเพจที่ต่างกันได้ถูกคาดหวังให้เป็นคุณลักษณะการใช้งานของผู้ใช้ที่แตกต่างกันดังที่ได้แสดงในตารางที่ 2.1 ความยาวของการอ้างอิงเพจ (Reference length) คือ จำนวนเวลาที่ผู้ใช้ใช้ในการดูเพจ ๆ หนึ่ง มีความท้าทายอย่างยิ่งในการคำนวณความยาวของการอ้างอิงเพจและการอ้างอิงไปข้างหน้าไกลที่สุด (Maximal forward reference) ซึ่งจะได้กล่าวถึงต่อไป

ประเภทเพจ	ลักษณะพิเศษทางกายภาพ	ลักษณะพิเศษทางการใช้งาน
เพจหลัก	• มีลิงค์ไปยังเพจอื่น ๆ มาก • อยู่ส่วนหัวของโครงสร้างไฟล์ไซต์	• เป็นเพจแรกในเซสชันของผู้ใช้
เพจเนื้อหา	• อัตราส่วนของบทความและรูปภาพต่อลิงค์มีสูง	• ค่าเฉลี่ยความยาวการอ้างอิงถึงยาว
เพจนำทาง	• อัตราส่วนของบทความและรูปภาพต่อลิงค์มีต่ำ	• ค่าเฉลี่ยความยาวการอ้างอิงถึงสั้น • ไม่เป็นเพจอ้างอิงที่ไกลที่สุด (Maximal forward reference)
เพจค้นดู	• จำนวนลิงค์เชื่อมต่อเข้ามาสูง • จำนวนลิงค์เชื่อมต่อออกไปต่ำหรือไม่มีเลย • มีเนื้อหาน้อยมา	• ค่าเฉลี่ยความยาวการอ้างอิงถึงสั้น • เป็นเพจอ้างอิงที่ไกลที่สุด
เพจส่วนตัว	• ไม่มีลักษณะพิเศษแน่นอน	• มีการใช้งานต่ำ

ตารางที่ 2.1 คุณลักษณะทั่วไปของเว็บเพจ

เพื่อที่จะรวมกลุ่มเว็บเพจที่อ้างอิงหนึ่ง ๆ ให้เป็นทรานส์แอ็กชันที่มีความหมายสำหรับการค้นหารูปแบบ เช่น กฎของสิ่งที่สัมพันธ์กัน จึงมีความจำเป็นที่ต้องเข้าใจแบบอย่างของพฤติกรรมการค้นหาของผู้ใช้ สำหรับจุดประสงค์ของการค้นหากฎของสิ่งที่สัมพันธ์กันจะให้ความสนใจที่การอ้างอิงเพจเนื้อหา สำหรับเพจประเภทอื่น ๆ เป็นเพียงแค่การอำนวยความสะดวกของการค้น

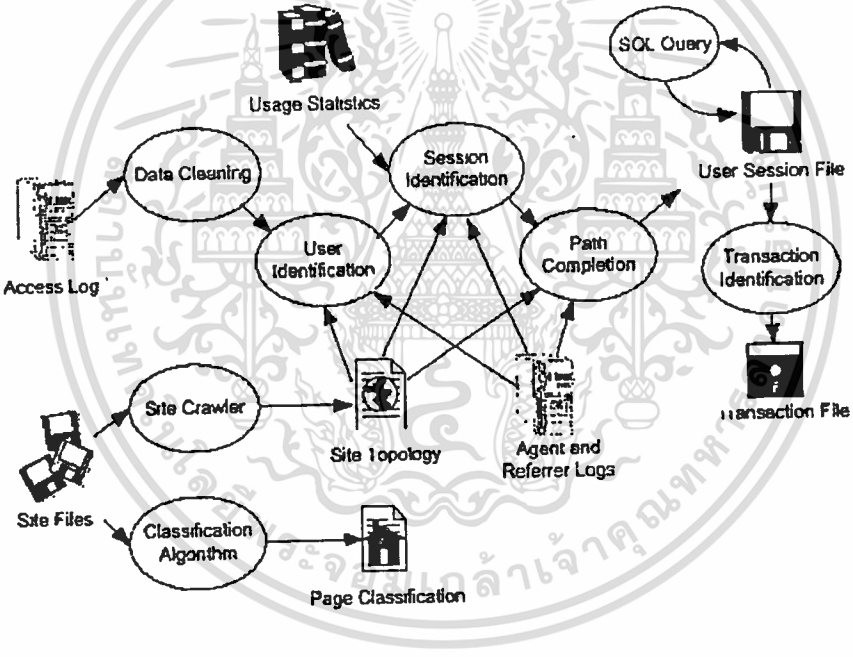
หาของผู้ใช้ในขณะการค้นหาข้อมูลสารสนเทศและอาจจะถูกอ้างอิงเป็นเพจสนับสนุน (Auxiliary page) ซึ่งเพจสนับสนุนสำหรับผู้ใช้คนหนึ่งอาจจะเป็นเพจเนื้อหาสำหรับอีกคนหนึ่งก็ได้ การระบุทรานส์แอ็กชันจึงจะทำได้เมื่อมีการตั้งสมมติฐานว่าเซสชันของผู้ใช้มีการระบุไว้เรียบร้อยแล้ว



รูปที่ 2.1 ประเภทของทรานส์แอ็กชัน : อ้างอิงเพจสนับสนุนและเพจเนื้อหา ไว้ที่แกนของเวลาด้วย A และ C ตามลำดับ

การใช้แนวคิดของการอ้างอิงเพจสนับสนุนและเพจเนื้อหามี 2 แนวทางในการกำหนดทรานส์แอ็กชันดังที่แสดงในรูปที่ 2.1 แนวทางแรกเป็นการกำหนดทรานส์แอ็กชันโดยรวมทุก ๆ การอ้างอิงเพจสนับสนุนขึ้นไปจนถึงการอ้างอิงเพจเนื้อหาสำหรับผู้ใช้คนหนึ่ง ๆ การไมนิ่งทรานส์แอ็กชันแบบเพจสนับสนุน-เนื้อหา (Auxiliary-Content) เหล่านี้เป็นการให้เส้นทางการเดินทางผ่านเว็บไซต์ไปยังเพจเนื้อหาเป็นจุดสำคัญ วิธีที่สองจะเป็นการกำหนด ทรานส์แอ็กชันที่เป็นทุก ๆ การอ้างอิงเพจเนื้อหาเท่านั้นสำหรับผู้ใช้หนึ่ง ๆ การไมนิ่งทรานส์แอ็กชันแบบเฉพาะเพจเนื้อหา (Content- only) จะให้ความสัมพันธ์ระหว่างเพจเนื้อหาของไซต์โดยไม่มีข้อมูลสารสนเทศที่เกี่ยวกับเส้นทางที่เกิดขึ้นระหว่างเพจ มีความสำคัญที่ต้องคำนึงถึงว่าผลลัพธ์ที่สร้างจากทรานส์แอ็กชันเฉพาะเพจเนื้อหาเป็นเพียงแค่การประยุกต์เมื่อเพจเหล่านั้นถูกใช้เป็นการอ้างอิงเนื้อหา ตัวอย่างเช่น กฎของสิ่งที่สัมพันธ์กัน $A \Rightarrow B$ มีความหมายโดยทั่วไปว่าเมื่อ A อยู่ในเซต ดังนั้น B ก็อยู่ในเซตด้วย อย่างไรก็ตามถ้ากฎนี้ถูกสร้างด้วยทรานส์แอ็กชันเฉพาะเพจเนื้อหาแล้วมันจะมีความหมายที่

เจาะจงมากกว่าคือ ชื่อ A จะอ้างไปถึง B เมื่อทั้ง A และ B ถูกใช้เป็นการอ้างถึงเนื้อหาเท่านั้น คุณสมบัตินี้ทำให้การทำคาด้าไมนิ่งกับทรานส์แอ็กชันเฉพาะเพจเนื้อหาจะได้ผลลัพธ์ของกฎที่อาจจะถูกทำให้หายไปโดยการรวมทุก ๆการอ้างอิงเพจในล็อก ถ้าผู้ใช้เข้าถึงเพจ A เป็นแบบเพจสนับสนุนไปต่อไปยังเพจ B แล้วการรวมการอ้างยิงเพจสนับสนุนเข้าไปในกระบวนการคาด้าไมนิ่งจะลดความเชื่อมั่นของกฎ $A \Rightarrow B$ ซึ่งมีความเป็นไปได้ที่กฎนี้จะไม่ถูกรายงานออกมา ขึ้นอยู่กับจุดมุ่งหมายของการวิเคราะห์โดยดูจากข้อดีข้อเสียที่ได้กล่าวไป จุดสำคัญคือทรานส์แอ็กชันเพจสนับสนุน-เนื้อหาสามารถถูกใช้เมื่อลักษณะดังที่กล่าวมานี้ไม่เป็นที่ต้องการ ความท้าทายของการระบุทรานส์แอ็กชันคือการหาตลอดเวลาว่าการอ้างอิงในล็อกของเซิร์ฟเวอร์อันใดเป็นเพจสนับสนุนและอันใดเป็นเพจเนื้อหา ได้มีการเสนอวิธีที่แตกต่างกัน 3 วิธีดังที่จะได้มีการนำเสนอในหัวข้อถัดไป



รูปที่ 2.2 รายละเอียดของการเตรียมข้อมูลสำหรับการไมนิ่งการใช้งานเว็บ

2.4 การเตรียมข้อมูล (Preprocessing)

รูปที่ 2.2 แสดงงานของการเตรียมข้อมูลของการไมนิ่งการใช้งานเว็บ ข้อมูลเข้าสำหรับขั้นตอนการเตรียมข้อมูลคือเซิร์ฟเวอร์ล็อก ไซต์ไฟล์และอาจจะมีสถิติการใช้งานจากการวิเคราะห์ก่อนหน้าหรือไม่ก็ได้ ข้อมูลผลลัพธ์คือไฟล์เซสชันของผู้ใช้ ไฟล์ทรานส์แอ็กชัน site topology และ page classification สิ่งกีดขวางที่สำคัญอย่างหนึ่งในการสร้างไฟล์เซสชันของผู้ใช้คือบราวเซอร์และ proxy server caching วิธีการในปัจจุบันในการเก็บข้อมูลสารสนเทศเกี่ยวกับ cache reference เป็น

การใช้คุกกี้ (cookies) และ cache busting ซึ่ง cache busting คือเป็นการปฏิบัติเพื่อป้องกันบราวเซอร์จากการใช้เพจเวอร์ชั่นที่ถูกเก็บไว้ในเครื่องของตนเองโดยการบังคับให้มีการเรียกเพจใหม่จากเซิร์ฟเวอร์ทุก ๆ ครั้งที่มีการเรียกดูเพจ แต่วิธีเหล่านี้ต่างมีข้อเสีย คุกกี้สามารถถูกลบโดยผู้ใช้และ cache busting ก็จะทำลายข้อได้เปรียบทางด้านความเร็วที่ caching ได้เตรียมไว้ให้และมีความเป็นไปได้ที่จะถูกยกเลิกโดยผู้ใช้ วิธีการอื่น ๆ ในการระบุผู้ใช้คือใช้การลงทะเบียนผู้ใช้ก่อนที่จะมีการเรียกดูเพจ ซึ่งการลงทะเบียนมีข้อดีคือสามารถเก็บข้อมูลสารสนเทศเพิ่มเติมได้มากกว่าที่เซิร์ฟเวอร์ล็อกได้เก็บไว้อย่างอัตโนมัติ ทำให้มีความง่ายในการระบุผู้ใช้และเซสชัน อย่างไรก็ตามประเด็นของความเป็นส่วนตัวทำให้ผู้ใช้หลายคนไม่เลือกที่จะเรียกดูไซต์ที่ต้องมีการลงทะเบียนและล็อกอิน หรือบางทีผู้ใช้อาจจะใส่ข้อมูลที่ไม่ถูกต้องลงไป วิธีการเตรียมข้อมูลที่ใช้ในระบบ WEBMINER จึงมีการใช้ข้อมูลสารสนเทศที่เอามาจากรูปแบบล็อกสามัญที่เป็นส่วนหนึ่งของโพรโตคอล HTTP เท่านั้น

2.4.1 การทำความสะอาดข้อมูล (Data Cleaning)

เทคนิคในการทำความสะอาดเซิร์ฟเวอร์ล็อกเพื่อกำจัดรายการที่ไม่เกี่ยวข้องออกไปเป็นส่วนสำคัญสำหรับทุก ๆ ประเภทของการวิเคราะห์เว็บล็อกไม่ใช่แค่งานคาค้าไมนิ่ง ความสัมพันธ์ที่ถูกค้นพบหรือรายงานทางสถิติจะเป็นประโยชน์ถ้าข้อมูลที่แสดงในเซิร์ฟเวอร์ล็อกให้ภาพที่ถูกต้องแม่นยำของการเข้าถึงเว็บไซต์ของผู้ใช้เท่านั้น โพรโตคอล HTTP ต้องการการเชื่อมต่อที่แยกออกจากกันสำหรับทุก ๆ ไฟล์ที่ถูกร้องขอจากเว็บเซิร์ฟเวอร์ ดังนั้นการร้องขอของผู้ใช้เพื่อที่จะดูเพจเฉพาะหนึ่ง ๆ บ่อยครั้งทำให้เกิดผลลัพธ์หลาย ๆ log entry เนื่องจากกราฟฟิกและสคริปต์ถูกดาวน์โหลดเป็นส่วนเพิ่มเติมจากไฟล์ html ส่วนมากแล้ว log entry ของไฟล์ HTML เพียงอย่างเดียวเท่านั้นที่ต้องการและควรจะถูกเก็บไว้ในไฟล์เซสชันของผู้ใช้ เนื่องจากโดยทั่วไปแล้วผู้ใช้ไม่ได้ร้องขอทุก ๆ กราฟฟิกบนเว็บเพจโดยตรง แต่พวกมันถูกดาวน์โหลดอัตโนมัติเนื่องมาจากเทรก HTML เนื่องจากจุดหลักของการไมนิ่งการใช้งานเว็บคือการหาพฤติกรรมของผู้ใช้ เพราะฉะนั้นจึงไม่จำเป็นเลยที่จะรวมไฟล์ที่ผู้ใช้ไม่ได้ร้องขอโดยตรงเข้าไปไว้ด้วย การกำจัดรายการที่ไม่เกี่ยวข้องสามารถทำได้โดยการตรวจสอบส่วนหลัง (suffix) ของชื่อ URL เช่น ทุก ๆ log entry ที่มีชื่อไฟล์ส่วนหลังเช่น GIF ,JPEG , jpg , JPG สามารถถูกลบออกไปได้ อีกทั้งพวกสคริปต่าง ๆ เช่น "count.cgi" ก็สามารถถูกลบออกไปด้วยได้ แต่สิ่งที่จะทำการลบออกไปเหล่านี้สามารถเปลี่ยนแปลงได้โดยขึ้นอยู่กับประเภทของไซต์ที่จะทำการวิเคราะห์ เช่น สำหรับเว็บไซต์ที่บรรจุรูปภาพเป็นสิ่งสำคัญการวิเคราะห์อาจจะไม่ต้องการให้มีการลบทุก ๆ ไฟล์ GIF หรือ JPEG ออกจากเซิร์ฟเวอร์ล็อกอย่างอัตโนมัติ ในกรณีนี้ log entry ของไฟล์รูปภาพอาจจะแสดงกิจกรรมโดยตรงของผู้ใช้ได้

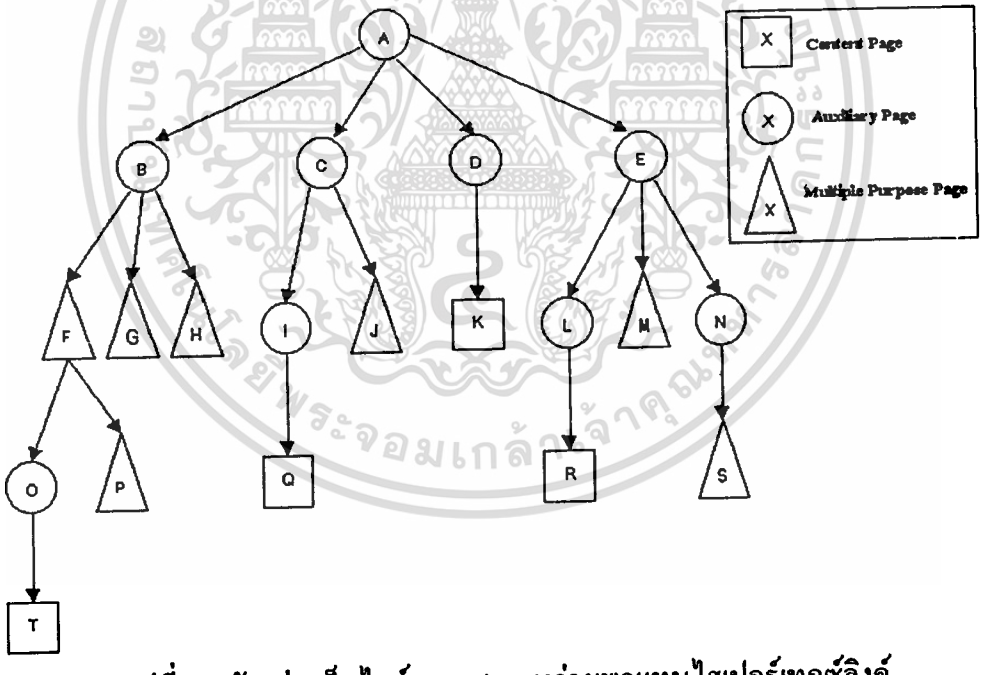
เป็นอย่างดีและควรจะเก็บไว้สำหรับการวิเคราะห์ เพื่อที่จะแบ่งแยกความแตกต่างระหว่าง log entry ที่เกี่ยวข้องหรือไม่เกี่ยวข้องเราสามารถใช้รายชื่อของชื่อไฟล์ที่ใช้งานจริงที่จะทำการลบออกหรือเก็บไว้แทนการใช้ไฟล์ส่วนหลังได้

2.4.2 การระบุผู้ใช้ (User Identification)

ต่อจากนั้นจะต้องมีการระบุผู้ใช้แต่ละคนได้ งานนี้ถูกทำให้ยุ่งยากมากขึ้นเนื่องจากมีการใช้ cache , firewall และ proxy server วิธีการไม่นิ่งการใช้งานเว็บที่อาศัยความร่วมมือของผู้ใช้เป็นแนวทางที่ง่ายที่สุดในการแก้ไขปัญหานี้ อย่างไรก็ตามวิธีการที่อยู่บนพื้นฐานของล็อกได้มีการเสนอไว้ว่า ในกรณีที่มี IP address เหมือนกัน ถ้าล็อกของตัวแทน (Agent log) มีการเปลี่ยนแปลงซอฟต์แวร์ค้นหา (Browser software) หรือ OS แล้วสามารถตั้งสมมติฐานได้อย่างสมเหตุสมผลได้ว่าแต่ละชนิดของตัวแทนที่ต่างกันของแต่ละ IP address จะเป็นการแสดงผู้ใช้ที่ต่างกัน เช่น พิจารณาเว็บไซต์ที่แสดงในรูปที่ 2.3 และตัวอย่างข้อมูลสารสนเทศที่เก็บจากการเข้าถึง, ตัวแทน และล็อกอ้างอิงดังแสดงในรูปที่ 2.4 ทุก ๆ log entry มี IP address เดียวกันและ user ID ไม่ได้ถูกบันทึก อย่างไรก็ตาม entry ตัวที่ 5,6,8 และ10 เป็นการเข้าถึงโดยการใช้ตัวแทนที่ต่างจาก entry อื่น ๆ จึงสามารถแนะนำได้ว่าล็อกนี้ได้แสดงอย่างน้อย 2 ผู้ใช้ หลักการต่อไปสำหรับการระบุผู้ใช้คือการใช้ล็อกเข้าถึงร่วมกับล็อกอ้างอิง และโครงสร้างของไซต์ (Site topology) เพื่อสร้างเส้นทางการเดินทางค้นหาสำหรับแต่ละผู้ใช้ ถ้าเพจที่ถูกร้องขอไม่สามารถเข้าถึงได้โดยตรงโดยไฮเปอร์ลิงค์จากเพจใด ๆ ที่เยี่ยมชมโดยผู้ใช้แล้ว หลักการนี้จะตั้งสมมติฐานว่ามีผู้ใช้คนอื่นที่มี IP address เดียวกันอยู่ จากตัวอย่างของล็อกดังรูปที่ 2.4 อีกครั้ง entry ตัวที่ 3 (เพจ L) ไม่ได้เข้าถึงตรงจากเพจ A,B อีกทั้งยัง entry ที่ 7 (เพจ R) สามารถไปถึงจากเพจ L แต่ไม่สามารถมาจาก log entry ใด ๆ ก่อนหน้านี้ ด้วยเหตุนี้จึงสามารถแนะนำได้ว่ามี 3 ผู้ใช้ใน IP address เดียวกัน ดังนั้นหลังจากขั้นตอนการระบุผู้ใช้ด้วยตัวอย่างของล็อกนี้สามารถระบุผู้ใช้ได้ 3 คน โดยมีเส้นทางการเดินทางค้นหาดังนี้คือ A-B-F-O-G-A-D , A-B-C-J และ L-R ตามลำดับ คิดไว้เสมอว่าหลักการเหล่านี้สำหรับระบุผู้ใช้เท่านั้น ผู้ใช้ 2 คนที่มี IP address เดียวกันที่ใช้บราวเซอร์ตัวเดียวกันบนเครื่องชนิดเดียวกันสามารถทำให้เกิดความสับสนได้ง่ายว่าทั้งสองเป็นผู้ใช้คนเดียวกันถ้าทั้งสองเข้าไปใช้ที่เพจชุดเดียวกัน ในทางกลับกันผู้ใช้คนเดียวที่ใช้ตัวค้นหาที่ต่างกัน 2 ตัวทำงานอยู่หรือผู้ที่พิมพ์ URL โดยตรงโดยไม่ใช้โครงสร้างของลิงค์ของไซต์สามารถทำให้เข้าใจผิดว่าเป็นผู้ใช้หลายคนได้

2.4.3 การระบุเซสชัน (Session Identification)

สำหรับล็อกที่เก็บในระยะที่ยาวมีความเป็นไปได้อย่างมากที่ผู้ใช้จะกลับเข้ามาเยี่ยมชมเว็บไซต์มากกว่า 1 ครั้ง จุดมุ่งหมายของการระบุเซสชันคือการแบ่งการเข้าถึงเพจของแต่ละผู้ใช้ไปเป็นเซสชัน วิธีที่ง่ายที่สุดคือใช้วิธีการค่าหมดเวลา (Timeout) โดยที่ถ้าเวลาระหว่างการร้องขอเพจเกินขอบเขตที่กำหนด ก็จะสามารถตั้งสมมติฐานได้ว่าผู้ใช้มีการเริ่มต้นเซสชันใหม่ หลาย ๆ ผลิตภัณฑ์ที่ขายอยู่ใช้ตั้งค่าหมดเวลาดีฟอลต์ไว้ที่ 30 นาที โดยอยู่บนพื้นฐานของการสังเกตข้อมูลเมื่อล็อกได้เคยถูกวิเคราะห์และได้หาสถิติการใช้งานค่าหมดเวลาที่เหมาะสมสำหรับเว็บไซต์เฉพาะสามารถถูกป้อนกลับเข้าไปในอัลกอริทึมที่ทำการระบุเซสชันได้ นี่คือเหตุผลที่ว่าทำไมสถิติการใช้งานจึงถูกแสดงเป็นข้อมูลเข้าให้กับขั้นตอนการระบุเซสชัน โดยการใช้ค่าหมดเวลา 30 นาที จากตัวอย่างเส้นทางสำหรับผู้ใช้คนที่หนึ่งสามารถถูกแตกออกเป็น 2 เซสชันที่แยกจากกันเนื่องจาก 2 การอ้างอิงสุดท้ายห่างจาก 5 การอ้างอิงแรกกว่าชั่วโมง ขั้นตอนการระบุเซสชันนี้ทำให้ได้ผลลัพธ์ 4 เซสชันของผู้ใช้ประกอบด้วย A-B-F-O-G, A-D, A-B-C-J และ L-R



รูปที่ 2.3 ตัวอย่างเว็บไซต์ – ลูกศรระหว่างเพจแทนไฮเปอร์เทกซ์ลิงค์

#	IP Address	Userid	Time	Method/URL/Protocal	Status	Size	Referel	Agent
1	123.456.78.9	-	[25/Apr/1998:03.04.41 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04(Win95,I)
2	123.456.78.9	-	[25/Apr/1998:03.05.34 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.04(Win95,I)
3	123.456.78.9	-	[25/Apr/1998:03.05.39 -0500]	"GET L.html HTTP/1.0"	200	4130	-	Mozilla/3.04(Win95,I)
4	123.456.78.9	-	[25/Apr/1998:03.06.02 -0500]	"GET F.html HTTP/1.0"	200	5096	B.html	Mozilla/3.04(Win95,I)
5	123.456.78.9	-	[25/Apr/1998:03.06.58 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04(X11,I,IPIX 6.2,IP22)
6	123.456.78.9	-	[25/Apr/1998:03.07.42 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.04(X11,I,IPIX 6.2,IP22)
7	123.456.78.9	-	[25/Apr/1998:03.07.55 -0500]	"GET R.html HTTP/1.0"	200	8140	L.html	Mozilla/3.04(Win95,I)
8	123.456.78.9	-	[25/Apr/1998:03.09.58 -0500]	"GET C.html HTTP/1.0"	200	1820	A.html	Mozilla/3.04(X11,I,IPIX 6.2,IP22)
9	123.456.78.9	-	[25/Apr/1998:03.10.02 -0500]	"GET O.html HTTP/1.0"	200	2270	F.html	Mozilla/3.04(Win95,I)
10	123.456.78.9	-	[25/Apr/1998:03.10.45-0500]	"GET J.html HTTP/1.0"	200	9430	C.html	Mozilla/3.04(X11,I,IPIX 6.2,IP22)
11	123.456.78.9	-	[25/Apr/1998:03.12.23 -0500]	"GET G.html HTTP/1.0"	200	7220	B.html	Mozilla/3.04(Win95,I)
12	123.456.78.9	-	[25/Apr/1998:05.05.22 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04(Win95,I)
13	123.456.78.9	-	[25/Apr/1998:05.06.03 -0500]	"GET D.html HTTP/1.0"	200	1680	A.html	Mozilla/3.04(Win95,I)

รูปที่ 2.4 ตัวอย่างข้อมูลสารสนเทศจากล็อกเข้าถึง , ล็อกอ้างอิงและเอเจนต์ล็อก

2.4.4 การทำเส้นทางให้สมบูรณ์ (Path Completion)

ปัญหาอื่น ๆ ในการระบุเซสชันของผู้ใช้ให้เชื่อถือได้คือการตัดสินใจถ้ามีการเข้าถึงที่สำคัญที่ไม่ได้ถูกบันทึกไว้ในล็อคการเข้าถึง (Access log) ปัญหานี้ถูกอ้างอิงเป็นขั้นตอนการทำเส้นทางให้สมบูรณ์วิธีการคล้ายกับที่ใช้ในการทำการระบุผู้ใช้ ถ้าการร้องขอเพจที่ถูกทำขึ้นไม่ได้ถูกลิงค์โดยตรงกับเพจสุดท้ายที่ผู้ใช้ร้องขอแล้ว referrer log สามารถใช้ในการตรวจสอบเพื่อดูการร้องขอเพจว่ามาจากไหน ถ้าเพจอยู่ในการร้องขอของผู้ใช้ที่เก็บไว้ในประวัติเมื่อเร็ว ๆ นี้แล้วสามารถตั้งสมมติฐานได้ว่าผู้ใช้ทำการย้อนกลับโดยใช้ปุ่ม "BACK" ที่มีบนตัวค้นหาส่วนมากโดยการเรียกเพจเวอร์ชันที่อยู่ใน cache ขึ้นมาจนกระทั่งเพจใหม่จะถูกร้องขอ ถ้า referrer log ไม่ชัดเจนสจมารถใช้ site topology มาวิเคราะห์ได้เช่นกัน ถ้ามีมากกว่า 1 เพจในประวัติของผู้ใช้ที่บรรจุลิงค์ไปยังเพจที่ร้องขอ มันสามารถถูกตั้งสมมติฐานว่าเพจที่ใกล้ที่สุดก่อนหน้าเพจที่ร้องขอคือต้นทางของการร้องขอครั้งใหม่ เพจอ้างอิงที่หายไปที่ถูกวินิจฉัยด้วยวิธีนี้จะถูกเพิ่มเข้าไปในไฟล์เซสชันของผู้ใช้ จากนั้นจะใช้อัลกอริทึมมาใช้ในการประมาณเวลาของแต่ละเพจอ้างอิงที่ถูกเพิ่มเข้าไป วิธีที่ง่ายในการเก็บ time-stamp คือตั้งสมมติฐานว่าการเข้าเยี่ยมชมเพจใด ๆ ที่เคยถูกเยี่ยมชมมาแล้วจะถือเอาเพจนั้นเป็นเพจสนับสนุนช่วงเวลาในการเข้าถึงเฉลี่ยของเพจสนับสนุนของไซต์สามารถถูกใช้ในการประมาณเวลาของการเข้าถึงสำหรับเพจที่หายไปนี้ได้ จากรูปที่ 2.3 และ 2.4 เพจ G ไม่สามารถเข้าถึงได้โดยตรงจากเพจ O ใน referrer log ของเพจ G ได้ร้องขอเพจ B จากเหตุการณ์นี้สามารถบอก

ได้ว่าผู้ใช้ได้ย้อนกลับไปยังเพจ B โดยใช้ปุ่ม "BACK" ก่อนที่จะร้องขอเพจ G ดังนั้นเพจ F และ B ควรจะถูกเพิ่มเติมเข้าไปในไฟล์เซสชันของผู้ใช้คนนี้ แต่ก็มีความเป็นไปได้ที่ผู้ใช้จะรู้ URL ของเพจ G และพิมพ์เข้าไปโดยตรงซึ่งเป็นเรื่องที่ไม่ต้องการให้เกิดขึ้น แต่จากการตั้งสมมติฐานว่าเหตุการณ์เช่นนี้ไม่ได้เกิดขึ้นบ่อยเพียงพอที่จะทำให้เกิดผลกระทบกับอัลกอริทึมไมนิ่ง ผลลัพธ์ของขั้นตอนการทำเส้นทางให้สมบูรณ์ จะได้เส้นทางของผู้ใช้ดังนี้คือ A-B-F-O-F-B-G, A-D, A-B-A-C-J และ L-R ผลลัพธ์ของแต่ละขั้นตอนของการเตรียมข้อมูลได้ถูกสรุปไว้ในตารางที่ 2.2

2.4.5 การจัดรูปแบบ (Formatting)

เมื่อได้ทำขั้นตอนที่เหมาะสมเพื่อทำการเตรียมข้อมูลเซิร์ฟเวอร์ล็อกแล้ว ส่วนสุดท้ายของการทำการเตรียมข้อมูลคือการเตรียมรูปแบบที่เหมาะสมของเซสชันหรือทรานส์แอ็กชันสำหรับแต่ชนิดของการทำดาต้าไมนิ่ง ตัวอย่างเช่นการเตรียมข้อมูลสำหรับการหากฎของสิ่งที่สัมพันธ์กันจะมีการดึงเวลาของแต่ละการเข้าถึงออกไปและทำการจัดรูปแบบของข้อมูลที่จำเป็นสำหรับอัลกอริทึมดาต้าไมนิ่งที่เราเจาะจงเลือกใช้

งาน	ผลลัพธ์
ทำความสะอาดล็อค	• A-B-L-F-A-B-R-C-O-J-G-A-D
ระบุผู้ใช้	• A-B-F-O-G-A-D • A-B-C-J • L-R
ระบุเซสชั่น	• A-B-G-O-G • A-D • A-B-C-J • L-R
ทำเส้นทางให้สมบูรณ์	• A-B-F-O-F-B-G • A-D • A-B-A-C-J • L-R

ตารางที่ 2.2 สรุปผลลัพธ์ของตัวอย่างการเตรียมข้อมูลล็อก

2.5 การระบุทรานส์แอ็กชัน

2.5.1 แบบอย่างโดยทั่วไป (General Model)

แต่ละเซสชันของผู้ใช้ในไฟล์เซสชันของผู้ใช้สามารถนำมาคิดได้ 2 แนวทางทั้งในแง่ว่าต่อทรานส์แอ็กชันมีการอ้างอิงเพจหลายเพจหรือชุดของทรานส์แอ็กชันที่แต่ละทรานส์แอ็กชันประกอบด้วยเพจอ้างอิง 1 เพจ จุดมุ่งหมายของการระบุทรานส์แอ็กชันคือการสร้างกลุ่มของการอ้างอิงที่มีความหมายสำหรับแต่ละผู้ใช้ เพราะฉะนั้นงานของการระบุทรานส์แอ็กชันคือเป็นได้ทั้งการแบ่ง (deviding) ทรานส์แอ็กชันขนาดใหญ่ให้เป็นทรานส์แอ็กชันขนาดเล็กลงหลาย ๆ ทรานส์แอ็กชัน หรือการรวม (merging) ทรานส์แอ็กชันเล็ก ๆ ให้เป็นทรานส์แอ็กชันที่ใหญ่ขึ้นหนึ่งทรานส์แอ็กชัน กระบวนการนี้สามารถถูกขยายเป็นหลาย ๆ ขั้นตอนของการรวมหรือการแบ่งเพื่อที่จะสร้างทรานส์แอ็กชันที่เหมาะสมสำหรับงานดาต้าไมนิ่ง วิธีการระบุทรานส์แอ็กชันสามารถถูกกำหนดเป็นได้ทั้งวิธีการรวมหรือการแบ่ง วิธีการทั้ง 2 ชนิดจะนำชุดของทรานส์แอ็กชันและตัวแปรที่เป็นไปได้บางตัวมาเป็นข้อมูลเข้าและผลลัพธ์ที่ออกมาจะได้เป็นชุดของทรานส์แอ็กชันที่ได้ถูกจัดการเรียบร้อยแล้ว โดยผลลัพธ์ที่ได้ออกมาจะมีรูปแบบเหมือนรูปแบบของข้อมูลเข้า การทำให้รูปแบบของ ทรานส์แอ็กชันที่เป็นข้อมูลเข้าเหมือนกับรูปแบบของทรานส์แอ็กชันที่เป็นผลลัพธ์นี้ทำให้สามารถรวมหลาย ๆ วิธีการเข้าด้วยกันเป็นลำดับของขั้นตอนการทำได้เพื่อให้การวิเคราะห์ข้อมูลออกมาได้เหมาะสม ให้ L คือชุดของเซสชันของผู้ใช้ สมาชิกของเซสชัน $l \in L$ ประกอบด้วย IP address ของไคลเอ็นต์ $l.ip$, id ผู้ใช้ของไคลเอ็นต์ $l.uid$, URL ของเพจที่เข้าถึง $l.url$, และเวลาของการเข้าถึง $l.time$ ยังมีข้อมูลอื่น ๆ ในสมาชิกของไฟล์เซสชันของผู้ใช้ เช่น รูปแบบของการร้องขอใช้ (เช่น POST หรือ GET) และขนาดของไฟล์ที่ทำการส่ง อย่างไรก็ตามข้อมูลเหล่านี้ไม่ได้ถูกใช้ในแบบอย่างของทรานส์แอ็กชัน ทรานส์แอ็กชัน t โดยทั่วไปมี 3 ส่วน ดังที่แสดงใน (1)

$$t = \left\langle ip_t, uid_t, \left\{ \left(l_1^t.url, l_1^t.time\right), \ldots, \left(l_m^t.url, l_m^t.time\right) \right\} \right\rangle$$
$$\text{โดย สำหรับ } 1 \le k \le m,\ l_k^t \in L,\ l_k^t.ip = ip_t,\ l_k^t.uid = uid_t \qquad (1)$$

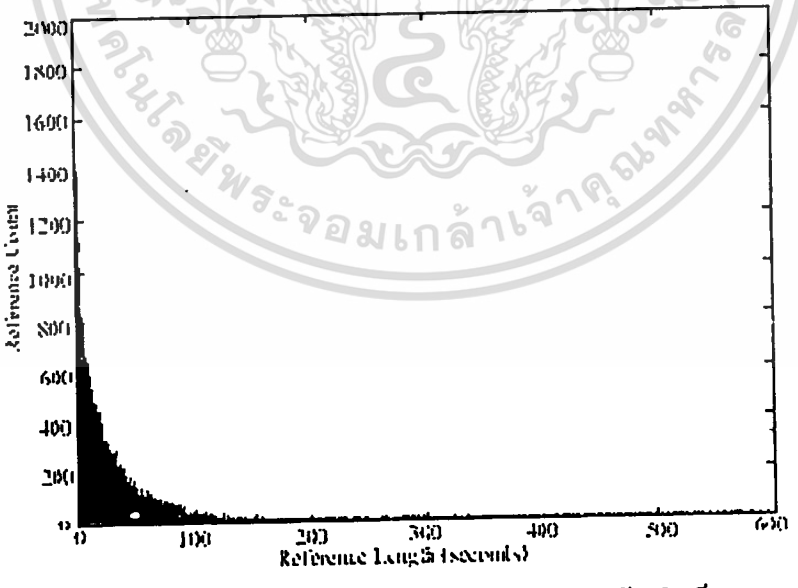
เมื่อข้อมูลเข้าเริ่มต้นของกระบวนการระบุทรานส์แอ็กชันประกอบด้วยทุก ๆ เพจที่เข้าถึงสำหรับเซสชันของผู้ใช้ที่กำหนดให้ ในขั้นตอนแรกในกระบวนการระบุทรานส์แอ็กชันจะเป็นการทำงานเกี่ยวกับการใช้วิธีการแบ่งส่วนเสมอ ในส่วนถัดไปจะอธิบายวิธีการแบ่งส่วนเพื่อระบุทรานส์แอ็กชัน 3 วิธี โดย 2 วิธีการแรกคือ การอ้างอิงความยาว (Reference length) และการอ้างอิงไปข้างหน้ายาวที่สุด (Maximal forward reference) ซึ่งเป็นวิธีการที่พยายามที่จะระบุทรานส์แอ็กชันที่มีความหมาย ส่วนวิธีการที่ 3 คือช่วงเวลา (Time window) ซึ่งไม่ได้อยู่บนพื้นฐานของแบบอย่างของ

การค้นหาใด ๆ และถูกให้เป็นตัวเปรียบเทียบหลักเพื่อเปรียบเทียบกับอีก 2 อัลกอริทึม ผลลัพธ์ของการใช้ 3 วิธีการที่แตกต่างกันนี้กับตัวอย่างในรูปที่ 2.4 ได้แสดงไว้ในตารางที่ 2.3

Approach	Transaction	
	Content-only	Auxiliary-Content
Reference Length	F-G, D, L-R , J	A-B-F, O-F-B-G, A-D L, R, A-B-A-C-J
Maximal Forward Reference	O-G, R, B-J ,D	A-B-F-O, A-B-G, L-R A-B, A-C-J, A-D
Time Window	A-B-F, O-F-B-G, A-D ,L-R, A-B-A-C-J	

ตารางที่ 2.3 สรุปผลลัพธ์ของตัวอย่างการระบุทรานส์แอ็กชัน
(ประเภทของทรานส์แอ็กชันไม่ได้ถูกระบุสำหรับวิธีการช่วงเวลา)

2.5.2 การระบุทรานส์แอ็กชันโดยใช้การอ้างอิงความยาว



รูปที่ 2.5 ฮิสโทรแกรมของความยาวของการอ้างอิงเว็บเพจ

วิธีการระบุทรานส์แอ็กชันโดยการอ้างอิงความยาวอยู่บนพื้นฐานของสมมติฐานที่ว่าจำนวนเวลาที่ผู้ใช้ใช้ไปกับเพจมีความสัมพันธ์กับว่าเพจควรจะถูกแบ่งเป็นเพจสนับสนุน (auxiliary page) หรือเพจเนื้อหา (content page) สำหรับผู้ใช้คนนั้น ๆ รูปที่ 2.5 แสดงฮิสโทรแกรมของความยาวของการอ้างอิงเพจระหว่าง 0-600 วินาทีจากล็อกของเซิร์ฟเวอร์หนึ่ง

การวิเคราะห์คุณลักษณะของล็อกของเซิร์ฟเวอร์อื่นใดโดยทั่วไปแล้วจะเหมือนในรูปที่ 2.5 รูปร่างของฮิสโทรแกรมมีลักษณะเอ็กซ์โปแนนเชียลขนาดใหญ่ มันถูกคาดไว้ว่าความแปรปรวนของเวลาที่ใช้ไปในเพจสนับสนุนจะน้อยมากและการอ้างอิงเพจสนับสนุนจะทำให้เกิดส่วนล่างทางขวาของส่วนโค้ง ความยาวของการอ้างอิงเพจเนื้อหาถูกคาดไว้ว่ามีความแปรปรวนที่กว้างและทำให้เกิดส่วนหางบนที่ขยายไปจนถึงการอ้างอิงที่ยาวนานที่สุด ถ้าตั้งสมมติฐานเกี่ยวกับเปอร์เซ็นต์ของการอ้างอิงเพจสนับสนุนในล็อก ความยาวของการอ้างอิงสามารถถูกคำนวณได้เพื่อประมาณค่าแบ่งแยก (Cutoff) ระหว่างการอ้างอิงเพจสนับสนุนและเพจเนื้อหา โดยเฉพาะอย่างยิ่งถ้ากำหนดเปอร์เซ็นต์ของเพจสนับสนุน วิธีการอ้างอิงความยาวนี้ได้ใช้ maximum likelihood ในการประมาณเพื่อคำนวณความยาวของเวลา t ดังที่แสดงใน (2)

$$t = \frac{-\ln(1-\gamma)}{\lambda}$$

โดย γ = % ของการอ้างอิงเพจสนับสนุน

λ = ค่าเฉลี่ยของความยาวของการอ้างอิงที่สังเกตได้จากทั้งสองฝ่าย (2)

คำนิยามที่ (2) มาจากการรวมสูตรสำหรับการกระจายเอกซ์โปรแนนเชียลจาก γ ถึง 0 maximum likelihood ที่ประมาณสำหรับการกระจายเอกซ์โปรแนนเชียลคือค่าเฉลี่ยที่สังเกตได้ ความยาวของเวลา t สามารถคำนวณได้โดยตรงโดยการเรียงลำดับทุก ๆ ความยาวของการอ้างอิงจากล็อกและจากนั้นทำการเลือกความยาวของการอ้างอิงที่วางอยู่ที่ตำแหน่ง $\gamma \times$ ขนาดของล็อก อย่างไรก็ตามการทำเช่นนี้เป็นการเพิ่มความซับซ้อนของอัลกอริทึมจากแบบเชิงเส้นไปเป็น O(nlogn) ในขณะที่ไม่ได้เพิ่มความแม่นยำของการคำนวณมากนักเนื่องจากค่าของ γ เป็นเพียงค่าประมาณเท่านั้น แม้ว่าการกระจายเอกซ์โปรแนนเชียลจะไม่เหมาะสมกับฮิสโทรแกรมของข้อมูลล็อกของเซิร์ฟเวอร์พอดี แต่มันก็ได้มีการเตรียมการประมาณค่าอ้างอิงความยาวของค่าแบ่งแยกที่มีเหตุมีผล

การนิยามทรานส์แอ็กชันภายใต้วิธีการอ้างอิงความยาวมีองค์ประกอบ 4 ส่วนดังแสดงใน (3) ซึ่งมีโครงสร้างเหมือน (1) ที่มีการเพิ่มความยาวการอ้างอิงเข้าไปที่แต่ละเพจ

$$t_{n'} = \left\langle ip_{t_{n'}}, uid_{t_{n'}}, \left\{ \left(l_1^{t_{n'}}.url, l_1^{t_{n'}}.time, l_1^{t_{n'}}.length \right), \ldots, \left(l_m^{t_{n'}}.url, l_m^{t_{n'}}.time, l_m^{t_{n'}}.length \right) \right\} \right\rangle$$

โดย, สำหรับ $1 \le k \le m,\ l_k^{t_{n'}} \in L,\ l_k^{t_{n'}}.ip = ip_{t_{n'}},\ l_k^{t_{n'}}.uid = uid_{t_{n'}}$ (3)

ความยาวของแต่ละการอ้างอิงถูกประมาณโดยผลต่างของเวลาระหว่างเวลาการอ้างอิงครั้งถัดไปและการอ้างอิงปัจจุบัน จะเห็นว่าการอ้างอิงสุดท้ายในแต่ละทรานส์แอ็กชันจะไม่มีเวลาครั้งถัดไปที่จะใช้ในการประมาณความยาวของการอ้างอิง วิธีการอ้างอิงความยาวมีการสร้างสมมติฐานที่ว่าทุก ๆ การอ้างอิงตัวสุดท้ายคือการอ้างอิงเนื้อหาและละทิ้งเพจนี้ในขณะที่คำนวณค่าเวลาแบ่งแยก สมมติฐานนี้สามารถทำให้เกิดข้อผิดพลาดได้ถ้าเพจสนับสนุนเฉพาะที่ถูกใช้โดยทั่วไปเป็นเพจที่ใช้ออกจากเว็บไซต์ ในขณะที่การขัดจังหวะเช่น โทรศัพท์มาหรือพักรับประทานอาหารกลางวันสามารถทำให้เกิดการแบ่งการอ้างอิงเพจสนับสนุนผิดเป็นการอ้างอิงเพจเนื้อหา ซึ่งจะเป็นข้อผิดพลาดอย่างมากถ้าเหตุการณ์นี้เกิดขึ้นที่เพจเดียวกัน ซึ่งเป็นเหตุผลที่มีการตั้งค่าเทรสโฮลท์สนับสนุน (Support threshold) น้อย ๆ ระหว่างการทำดาต้าไมนิ่งเพื่อที่จะแยกข้อผิดพลาดเหล่านี้ออกมาได้

เมื่อค่าเวลาแบ่งแยกถูกคำนวณออกมา ทรานส์แอ็กชันทั้งสองประเภทนี้สามารถถูกทำขึ้นมาได้โดยการเปรียบเทียบแต่ละค่าความยาวของการอ้างอิงเทียบกับค่าเวลาแบ่งแยก โดยขึ้นอยู่กับจุดมุ่งหมายในการวิเคราะห์ว่าจะใช้ทรานส์แอ็กชันสนับสนุน-เนื้อหา (Auxiliary-content transaction) หรือทรานส์แอ็กชันเฉพาะเนื้อหา (Content-only transaction) ถ้า C คือค่าเวลาแบ่งแยก สำหรับทรานส์แอ็กชันสนับสนุน-เนื้อหามีเงื่อนไขดังนี้

$$\text{สำหรับ } 1 \le k \le (m-1) : l_k^{t_{n'}}.length \le C \text{ และ } k = m : l_k^{t_{n'}}.length > C$$

จะถูกเพิ่มเข้าไปใน (3) และสำหรับทรานส์แอ็กชันเฉพาะเนื้อหาจะมีเงื่อนไข

$$\text{สำหรับ } 1 \le k \le m : l_k^{t_{n'}}.length > C$$

ถูกเพิ่มเข้าไปใน (3) ที่แสดงในหัวข้อที่ 4 ด้วยสมมติฐานที่ว่าเพจที่มีหลายจุดมุ่งหมายถูกใช้เป็นเหมือนเพจเนื้อหาครึ่งหนึ่งของเวลาที่มีการเข้าถึง ค่าเวลาแบ่งแยกที่คำนวณได้คือ 78.4 ผลลัพธ์ที่เป็นทรานส์แอ็กชันเฉพาะเนื้อหาที่ได้คือ F-G, D, L-R และ L ตามที่แสดงในตารางที่ 2.7 จะเห็นว่าเพจ L ถูกแบ่งเป็นเพจเนื้อหาแทนที่จะเป็นเพจสนับสนุนซึ่งเกิดขึ้นเนื่องมาจากเหตุผลที่กล่าวไว้ข้างต้น

ตัวแปรหนึ่งที่วิธีการอ้างอิงความยาวต้องการคือการประมาณเปอร์เซนต์ทั้งหมดของการอ้างอิงที่เป็นเพจสนับสนุน การประมาณค่าเปอร์เซนต์นี้สามารถทำได้โดยอยู่บนพื้นฐานของโครงสร้างและเนื้อหาของไซต์หรือประสบการณ์ของการวิเคราะห์ข้อมูลด้วยล็อกของเซิร์ฟเวอร์อื่น ๆ ซึ่งผลของวิธีนี้นับว่าเชื่อถือได้พอสมควรและช่วงที่กว้างของเปอร์เซนต์ของเพจสนับสนุนจะให้ผลของชุดของกฎของสิ่งที่มีความสัมพันธ์กันที่สมเหตุสมผล

2.5.3 การระบุทรานส์แอ็กชันโดยการอ้างอิงไปข้างหน้าไกลที่สุด (Maximal forward reference)

วิธีการระบุทรานส์แอ็กชันโดยใช้การอ้างอิงไปข้างหน้ายาวที่สุดจะใช้หลักการว่าแทนที่จะใช้เวลาที่ใช้ในแต่ละเพจเหมือนวิธีการอ้างอิงความยาว แต่ทรานส์แอ็กชันจะถูกกำหนดให้เป็นชุดของเพจในเส้นทางจากเพจแรกในเซสชันของผู้ใช้ไปจนถึงเพจก่อนที่จะการทำการอ้างอิงย้อนกลับขึ้น (Backward reference) การอ้างอิงไปข้างหน้า (Forward reference) ถูกนิยามว่าเป็นเพจที่ยังไม่เคยอยู่ในชุดของเพจสำหรับทรานส์แอ็กชันปัจจุบัน ซึ่งคล้ายกับการอ้างอิงย้อนกลับซึ่งถูกนิยามว่าคือเพจที่บรรจุอยู่ในชุดของเพจสำหรับทรานส์แอ็กชันปัจจุบันเรียบร้อยแล้ว ทรานส์แอ็กชันใหม่จะถูกเริ่มต้นเมื่อการอ้างอิงไปข้างหน้าครั้งถัดไปเกิดขึ้น เพจการอ้างอิงไปข้างหน้าที่ยาวที่สุดคือเพจเนื้อหาและเพจที่นำไปสู่แต่ละเพจการอ้างอิงไปข้างหน้าที่ยาวที่สุดคือเพจสนับสนุน วิธีนี้เหมือนกับวิธีการอ้างอิงความยาวที่ทรานส์แอ็กชันมี 2 ชุด คือ ทรานส์แอ็กชันเพจสนับสนุน-เพจเนื้อหาหรือทรานส์แอ็กชันเฉพาะเพจเนื้อหา เราสามารถนำเอานิยามของทรานส์แอ็กชันโดยทั่วไปที่เสนอไว้ใน (1) มาใช้กับวิธีนี้ จากการใช้ตัวอย่างในหัวข้อที่ 4 ทรานส์แอ็กชันเพจสนับสนุน-เพจเนื้อหาถูกสร้างขึ้นได้เป็น A-B-F-O, A-B-G, L-R, A-B, A-C-J และ A-D ส่วนทรานส์แอ็กชันเฉพาะเพจเนื้อหาคือ O-G, R, B-J และ D วิธีการอ้างอิงไปข้างหน้าที่ยาวที่สุดมีข้อดีกว่าวิธีการอ้างอิงความยาวที่วิธีนี้ไม่ต้องการพารามิเตอร์เป็นข้อมูลเข้า

2.5.4 การระบุทรานส์แอ็กชันโดยใช้ช่วงเวลา (Time window)

วิธีการระบุทรานส์แอ็กชันโดยใช้ช่วงเวลามีการแบ่งเซสชันของผู้ใช้ตามคาบของเวลา วิธีการนี้ไม่ได้พยายามที่จะระบุทรานส์แอ็กชันโดยอยู่บนพื้นฐานของเพจสนับสนุนหรือเพจเนื้อหา แต่จะตั้งสมมติฐานว่าทรานส์แอ็กชันที่มีความหมายมีค่าเฉลี่ยของความยาวทั้งหมดที่สัมพันธ์กับพวกมัน สำหรับค่าที่กว้างเพียงพอของช่วงเวลา (Time window) ทำให้แต่ละทรานส์แอ็กชันบรรจุทั้งเซสชันของผู้ใช้ เนื่องจากวิธีใช้ช่วงเวลาไม่ได้อยู่บนพื้นฐานของแบบอย่างที่แสดงไว้ในหัวข้อที่ 3 ดังนั้นทำให้ไม่มีความเป็นไปได้ที่จะแยกออกเป็น 2 ทรานส์แอ็กชันได้ การอ้างอิงครั้งสุดท้ายของแต่ละทรานส์แอ็กชันไม่ได้มีลักษณะเป็นเพจอ้างอิงเนื้อหา ซึ่งแตกต่างจากแนวทางของทรานส์แอ็กชัน

เพจสนับสนุน-เพจเนื้อหาที่หาได้จากวิธีการอ้างอิงความยาวและวิธีการอ้างอิงไปข้างหน้ายาวที่สุด ถ้า W คือความกว้างของช่วงเวลา นิยามที่ (1)ได้ถูกประยุกต์สำหรับทรานส์แอ็กชันที่นิยามด้วยวิธีช่วงเวลาจะมีการเพิ่มเงื่อนไข

$$\left(l_m^t.time - l_1^t.time\right) \leq W$$

เนื่องจากบางความแปรปรวนที่สัมพันธ์กับความยาวของแต่ละทรานส์แอ็กชันจริง ๆ ไม่น่าจะเป็นไปได้ที่จะให้ช่วงเวลาที่คงที่มาใช้ในการแตกล็อกได้เหมาะสม อย่างไรก็ตามวิธีช่วงเวลายังสามารถถูกใช้เป็นเหมือนวิธีการรวม (Merge) ในการเชื่อมต่อกับวิธีการแบ่ง (Devide) ตัวอื่น ตัวอย่างเช่น หลังจากที่ทำวิธีการอ้างอิงความยาวแล้ว วิธีการรวมโดยใช้ช่วงเวลาที่มีพารามิเตอร์เป็นข้อมูลเข้ามีค่า 10 นาที สามารถถูกนำมาใช้เพื่อให้แน่ใจว่าแต่ละ ทรานส์แอ็กชันมีค่าความยาวที่น้อยที่สุดของทั้งหมดเท่าใด

2.6 เทคนิคที่ใช้ค้นหาความรู้จากเว็บทรานส์แอ็กชัน

การไมนิ่งการใช้งานเว็บ (Web usage mining) คือแอปพริเคชั่นของเทคนิคดาต้าไมนิ่งที่ใช้กับที่เก็บข้อมูลเว็บขนาดใหญ่เพื่อที่จะผลิตผลลัพธ์ที่สามารถถูกใช้ในงานออกแบบ อัลกอริทึมดาต้าไมนิ่งบางตัวที่ถูกใช้โดยทั่วไปในการไมนิ่งการใช้งานเว็บคือการวิเคราะห์เส้นทาง (Path analysis) การหากฎของสิ่งที่สัมพันธ์กัน (Association rule) การหารูปแบบตามลำดับ (Sequential pattern) และทำการแบ่งกลุ่ม (Clustering)

มีกราฟหลากหลายประเภทที่สามารถถูกสร้างขึ้นมาสำหรับการทำการวิเคราะห์เส้นทาง เนื่องจากกราฟเป็นสิ่งที่เป็นตัวแทนความสัมพันธ์ระหว่างเว็บเพจ กราฟโดยทั่วไปส่วนใหญ่เป็นการแสดงโครงสร้างทางกายภาพของเว็บไซต์ที่มีโหนดแทนเว็บเพจและกิ่งที่เชื่อมต่อระหว่างเพจคือไฮเปอร์เทกซ์ลิงค์ กราฟชนิดอื่นได้ถูกสร้างโดยอยู่บนพื้นฐานของประเภทของเว็บเพจซึ่งมีกิ่งแสดงความเหมือนกันระหว่างเพจหรือการสร้างกิ่งที่มีการกำหนดจำนวนของผู้ใช้ที่เดินทางจากเพจหนึ่งไปอีกเพจไว้ งานวิจัยในปัจจุบันมีความสนใจที่จะค้นหารูปแบบการเดินทางที่เกิดขึ้นบ่อย (Frequent traversal pattern) หรือลำดับการอ้างอิงที่มากที่สุด (Large reference sequence) จากโครงสร้างทางกายภาพของกราฟ การวิเคราะห์เส้นทางสามารถถูกใช้ในการตัดสินหาเส้นทางที่มีการเข้าเยี่ยมชมมากที่สุดในเว็บไซต์ ตัวอย่างของข้อมูลสารสนเทศที่สามารถค้นพบได้โดยใช้การวิเคราะห์เส้นทางเป็นดังนี้

- 70 % ของไคลเอ็นต์ที่เข้าถึง /company/product2 จะเริ่มต้นการเข้าถึงที่ /company ก่อนจากนั้นผ่าน /company/new , /company/products, และ /company/product1

- 80 % ของไคลเอ็นต์ที่เข้าถึงไซต์จะเริ่มต้นจาก /company/products
- 65 % ของไคลเอ็นต์ออกจากไซต์หลังจากที่มีการอ้างอิงเพจ 4 เพจหรือน้อยกว่านั้น

ในกฎแรกสามารถบอกได้ว่ามีข้อมูลสารสนเทศที่มีประโยชน์ใน /company/product2 แต่เนื่องจากผู้ใช้มีแนวโน้มที่จะเส้นเดินอ้อม ๆ ไปยังเพจทำให้แสดงว่าเพจนี้ยังไม่ได้ถูกทำให้ไปถึงอย่างเด่นชัด กฎที่สองบอกได้ว่าผู้ใช้มีการเข้าถึงไซต์โดยผ่านเพจอื่นที่ไม่ใช่เพจหลัก (เช่นสมมติว่าเป็นเพจ /company) ซึ่งทำให้มีแนวความคิดที่ว่าจะทำการรวมข้อมูลสารสนเทศลงไปบนเพจนี้เลย และในกฎสุดท้ายแสดงให้เห็นถึงอัตราการใช้งานของไซต์ เนื่องจากผู้ใช้ไม่มีการค้นหามากเกินกว่า 4 เพจในไซต์เพราะฉะนั้นเราจะต้องทำให้แน่ใจว่าข้อมูลสารสนเทศที่สำคัญได้ถูกบรรจุอยู่ในเพจทั้ง 4 ของไซต์ที่ผู้ใช้มักเข้าถึง

เทคนิคการไมนิ่งกฎของสิ่งที่สัมพันธ์กันจะค้นหาความสัมพันธ์ที่ไม่เรียงลำดับระหว่างรายการสิ่งของที่พบในฐานข้อมูลทรานส์แอ็กชัน ในสิ่งแวดล้อมของการไมนิ่งการใช้งานเว็บ ทรานส์แอ็กชันคือกลุ่มของเว็บเพจที่เข้าถึง ซึ่งรายการสิ่งของคือเพจที่เข้าถึงตัวหนึ่ง ๆ ตัวอย่างของกฎของสิ่งที่สัมพันธ์กันที่ค้นพบเป็นดังนี้

- 40% ของไคล์เอ็นต์ที่เข้าถึงเว็บเพจด้วย URL /company/product1 ยังมีการเข้าถึง /company/product2
- 30 % ของไคล์เอ็นต์ที่เข้าถึง /company/special จะมีการสั่งสินค้าแบบออนไลน์ใน /company/product1 ด้วย

รายงานเปอร์เซ็นต์ในตัวอย่างนี้หมายถึงค่าความเชื่อมั่น (Confidence) ค่าความเชื่อมั่นคือจำนวนของทรานส์แอ็กชันที่บรรจุทุก ๆ รายการในกฎหารด้วยจำนวนของทรานส์แอ็กชันที่บรรจุ rule antecedent (antecedent คือ /company/product1 สำหรับตัวอย่างแรก และ /company/special สำหรับตัวอย่างที่สอง)

ปัญหาของการค้นพบรูปแบบตามลำดับคือการหารูปแบบทรานส์แอ็กชันระหว่างทรานเซคชั่น (Inter-transaction pattern) หรือรูปแบบตามลำดับเป็นการแสดงถึงชุดของรายการสิ่งของที่ถูกติดตามโดยรายการอื่นในลำดับตามค่าเวลาในชุดทรานส์แอ็กชัน โดยการวิเคราะห์ข้อมูลสารสนเทศนี้การไมนิ่งการใช้งานเว็บสามารถหาความสัมพันธ์ชั่วคราวของรายการข้อมูล เช่นตามตัวอย่างดังนี้

- 30 % ของไคล์เอ็นต์ที่เข้าเยี่ยมชม /company/products ได้มีการทำการค้นหาใน Yahoo ภายในอาทิตย์ที่ผ่านมาด้วยคำค้น *w*
- 60 % ของไคล์เอ็นต์ผู้ที่สั่งสินค้าแบบออนไลน์ใน /company/product1 ยังมีการสั่งสินค้าออนไลน์ใน /company/product4 ภายใน 15 วัน

รายงานเปอร์เซ็นต์ในตัวอย่างหมายถึงค่าสนับสนุน(Support) ค่าสนับสนุนคือเปอร์เซ็นต์ของทรานส์แอ็กชันที่บรรจุรูปแบบที่ให้ไว้ ทั้งค่าความเชื่อมั่นและค่าสนับสนุนโดยทั่วไปแล้วถูกใช้เป็นเหมือนเทรสโฮสต์เพื่อที่จะจำกัดจำนวนของกฎที่ค้นพบและรายงานออกมา ตัวอย่างเช่น ถ้าให้เทรสโฮสต์ค่าสนับสนุนเท่ากับ 50% แล้วตัวอย่างที่หนึ่งของตัวอย่างการหารูปแบบตามลำดับจะไม่ถูกรายงานออกมา

การวิเคราะห์การจับกลุ่ม (Clustering Analysis) เป็นการจับกลุ่มเข้าด้วยกันของผู้ใช้หรือรายการข้อมูลที่มีคุณลักษณะเหมือนกันเข้าไว้ด้วยกัน การจับกลุ่มของข้อมูลสารสนเทศของผู้ใช้หรือข้อมูลจากเว็บเซิร์ฟเวอร์ล็อกสามารถนำมาอำนวยความสะดวกในการพัฒนาและสร้างกลยุทธทางการตลาดในอนาคตทั้งในแบบออนไลน์และออฟไลน์เช่น การส่งเมลกลับอัตโนมัติไปยังผู้เข้าเยี่ยมชมที่ตกอยู่ในกลุ่มที่แน่นอน หรือการเปลี่ยนแปลงไซต์อย่างไดนามิคตามผู้เยี่ยมชมที่กลับเข้ามาโดยอยู่บนพื้นฐานของการจัดกลุ่มในอดีตของผู้เข้าเยี่ยมชมคนนี้

ตามตัวอย่างที่แสดงไว้ การไมนิ่งสำหรับการหาความรู้จากข้อมูลเว็บล็อกมีศักยภาพในการหาข้อมูลสารสนเทศในปริมาณที่มาก ในขณะที่ความแม่นยำขึ้นอยู่กับอัลกอริทึมดาต้าไมนิ่งที่มีอยู่แล้วด้วย เช่น การค้นพบกฎของสิ่งที่สัมพันธ์กันหรือรูปแบบตามลำดับ งานทั้งหมดนี้มีความยากในการปรับอัลกอริทึมที่มีอยู่กับข้อมูลใหม่ที่เข้ามา ในทางอุดมคติข้อมูลเข้าสำหรับกระบวนการไมนิ่งการใช้งานเว็บคือไฟล์ที่เป็นลักษณะเซสชันไฟล์ของผู้ใช้โดยไฟล์นี้ได้มาจากการบันทึกว่าใครเข้าถึงมาที่เว็บไซต์ เพจใดถูกร้องขอและร้องขอในลำดับอะไร และแต่ละเพจถูกแสดงนานเท่าใด เซสชันของผู้ใช้ถูกพิจารณาเป็นทุก ๆ การเข้าถึงเพจที่เกิดขึ้นระหว่างการเข้าเยี่ยมชมต่อครั้งของเว็บไซต์ ข้อมูลสารสนเทศที่บรรจุในเว็บเซิร์ฟเวอร์ล็อกดิบไม่ได้แสดงถึงไฟล์เซสชันของผู้ใช้จริง เนื่องด้วยเหตุผลต่าง ๆ ที่จะกล่าวถึงก่อนหน้านี้ โดยเฉพาะอย่างยิ่งมีความยุ่งยากหลายอย่างในการจัดการเซิร์ฟเวอร์ล็อกดิบเพื่อกำจัดข้อมูลที่ไม่เกี่ยวข้องออกไป การระบุผู้ใช้และการระบุเซสชันของผู้ใช้ภายในเซิร์ฟเวอร์ และการระบุทรานส์แอ็กชันที่มีความหมายที่จะนำไปใช้ได้จากเซสชันของผู้ใช้

บทที่ 3
ทฤษฎีที่นำมาใช้

การศึกษาถึงปัญหาของการไมนิ่งรูปแบบการเดินทาง การดำเนินการแก้ปัญหาประกอบด้วย 2 ขั้นตอนหลัก ๆ คือ ขั้นแรกจะใช้อัลกอริทึม MF (Standing for maximal forward reference) ทำการแปลงรูปแบบของลำดับของข้อมูลล็อก (Log data) ให้เป็นชุดของลำดับการเดินทาง (Traversal subsequence) โดยแต่ละลำดับการเดินทางจะแสดงถึงการค้นหาไปข้างหน้าที่ไกลที่สุดจากจุดเริ่มต้นของการเข้าถึงของผู้ใช้ (Maximal forward reference) ซึ่งในขั้นตอนของการแปลงนี้จะทำการกรองผลกระทบของการค้นหาย้อนกลับออกไป ซึ่งทำให้การเดินทางมีความง่ายขึ้นและทำให้เราสามารถมุ่งความสนใจไปที่ลำดับของการเข้าถึงของผู้ใช้ที่มีความหมายจริง ๆ ได้ ขั้นตอนที่สองจะใช้อัลกอริทึมมาช่วยหารูปแบบการเดินทางที่มีความถี่ที่ยอมรับได้จาก Maximal forward reference ที่ได้จากขั้นตอนแรก ซึ่งเราจะเรียกรูปแบบการเดินทางนี้ว่า Large reference sequence โดยที่ Large reference sequence คือลำดับการค้นหาที่มีจำนวนครั้งของการปรากฏในฐานข้อมูลเพียงพอกับที่กำหนดไว้ ปัญหาของการหา Large reference sequence จะคล้ายกับการหา Large itemset สำหรับกฎของสิ่งที่สัมพันธ์กัน (Association rule) โดยที่ Large itemset คือชุดของสิ่งของที่มีการปรากฏในทรานส์แอ็กชันตามจำนวนที่กำหนดไว้ อย่างไรก็ตามมีความแตกต่างระหว่าง 2 ปัญหานี้ในลักษณะที่ลำดับการเข้าถึงในการไมนิ่งรูปแบบการเดินทางมีการเรียงลำดับการเข้าถึงใน Maximal forward reference แต่ในขณะที่ Large itemset ในการไมนิ่งกฎของสิ่งที่สัมพันธ์กันเป็นเพียงแค่การรวมของสิ่งของเข้าไว้ด้วยกันในทรานส์แอ็กชัน ด้วยความแตกต่างนี้ทำให้ต้องออกแบบอัลกอริทึมใหม่เพื่อใช้ในการหา Large reference sequence

3.1 ลักษณะของปัญหา

ในสิ่งแวดล้อมที่มีการใช้หาข้อมูลสารสนเทศ ออบเจคจะถูกเชื่อมเข้าด้วยกันโดยผู้ใช้มักจะเดินทางไปยังออบเจคก่อนหน้าและถัดไปด้วยลิคค์และไอคอนที่จัดเตรียมไว้ให้ ทำให้มีผลว่าบางโหนดจะถูกเข้าถึงซ้ำเนื่องจากตำแหน่งที่ตั้งของมันมากกว่าจะเป็นสิ่งที่บรรจุอยู่ข้าง เช่นในเวิลด์ไวด์เว็บผู้ใช้มักจะย้อนกลับไปยังโหนดก่อนหน้าก่อนแล้วจึงเดินทางต่อไปยังโหนดอื่น แทนที่จะเปิด URL ใหม่ ด้วยเหตุนี้การดึงความหมายที่ใช้ได้จากรูปแบบการเข้าถึงของผู้ใช้จากฐานข้อมูลล็อกจึงทำได้ยากยิ่งขึ้น โดยทั่วไปเรามีความต้องการที่จะศึกษาถึงผลกระทบของการเดินทางย้อน

กลับและค้นหารูปแบบการเข้าถึงจริง ๆ ในมุมมองนี้เราตั้งสมมติฐานว่าการเข้าถึงย้อนกลับถูกทำขึ้นเพื่อความง่ายของการเดินทางไม่ใช่สำหรับการค้นหา (Browse) เราจึงมุ่งไปให้ความสนใจในการค้นหารูปแบบการเข้าถึงไปข้างหน้า โดยที่การเข้าถึงย้อนกลับหมายถึงการเข้าถึงออบเจคซ้ำอีกครั้งโดยการเข้าถึงของผู้ใช้คนเดียวกัน เมื่อการเข้าถึงย้อนกลับเกิดขึ้นเส้นทางของการเข้าถึงไปข้างหน้าจะสิ้นสุดลง ผลลัพธ์ของเส้นทางการเข้าถึงไปข้างหน้านี้จะอยู่ในรูปของ Maximal forward reference เมื่อได้รับ Maximal forward reference แล้วเราจะกลับไปที่จุดเริ่มต้นของการเข้าถึงไปข้างหน้าอีกครั้งและสืบค้นเส้นทางของการเข้าถึงไปข้างหน้าเส้นทางอื่นใหม่ และถ้ามีโหนดที่มีจุดเริ่มต้นเป็น null เกิดขึ้นสามารถบ่งชี้ว่าเป็นการสิ้นสุดเส้นทางของการเข้าถึงไปข้างหน้าที่กำลังทำอยู่และเป็นจุดเริ่มต้นของเส้นทางอื่น

จะมีการนำเสนอและอธิบายรูปแบบของอัลกอริทึมที่ใช้ในการหา Maximal forward reference (อัลกอริทึก MF) ในส่วนถัดไป ซึ่งในที่นี้จะมีการแสดงตัวอย่างสำหรับการหา Maximal forward reference โดยสมมติว่าล็อกของการเดินทางบรรจุเส้นทางการเดินทางของผู้ใช้คนหนึ่งดังต่อไปนี้: {A, B,C, D, C, B, E, G, H, G, W, A, O, U, O, V} ดังแสดงในรูปที่ 3.1 จากนั้นใช้อัลกอริทึม MF ในการหาชุดของ Maximal forward reference สำหรับผู้ใช้คนนี้ ซึ่งจะได้เป็น {ABCD , ABEGH , ABEGW , AOU , AOV} หลังจากที่หา Maximal forward reference สำหรับผู้ใช้ทุก ๆ คนแล้วเราจะทำการหาความถี่ของการปรากฏของชุดลำดับในทุก ๆ Maximal forward reference โดย Large reference sequence คือลำดับของการเข้าถึงในชุดของ Maximal forward reference ที่มีจำนวนครั้งของการปรากฏเพียงพอ ซึ่งจำนวนครั้งที่ลำดับของการเข้าถึงปรากฏที่ใช้ในการหา Large reference sequence นี้ถูกเรียกว่าค่าสนับสนุนที่น้อยที่สุด (Minimal support) และ Large k-reference คือ Large reference sequence ที่มี k อิลิเมนต์ เราจะใช้ L_k เป็นเครื่องหมายแสดงถึงชุดของ Large k-reference และ C_k คือชุดของตัวเลือก (Candidate set) ของมัน จากที่ได้กล่าวมาแล้วก่อนหน้านี้ว่ามีความแตกต่างกันอย่างมากระหว่างการไมนิ่งรูปแบบการเดินทางที่การเข้าถึงเป็นลำดับตามกันมาใน *Maximal forward reference แต่ในขณะที่ Large itemset ในการไมนิ่งกฎของสิ่ง* ที่สัมพันธ์กันเป็นเพียงแค่ชุดของสิ่งของในทรานส์แอ็กชัน ด้วยเหตุผลนี้จึงมีความจำเป็นที่ต้องสร้างอัลกอริทึมใหม่สำหรับใช้ในการหา Large reference sequence

ในการอ้างถึง Maximal reference sequence หลังจากได้รับ Large reference sequence สามารถทำได้อย่างตรงไปตรงมา โดย Maximal reference sequence คือ Large reference sequence ที่ไม่ได้ถูกบรรจุใน Maximal reference sequence อื่น ตัวอย่างเช่น สมมติว่า {AB, BE, AD, CG, GH , BG} เป็นชุดของ Large 2-reference (L_2) และ {ABE, CGH} คือชุดของ Large 3-reference (L_3) ผลลัพธ์ของ Maximal reference sequence จะคือ AD , BG , ABE และ CGH ซึ่ง Maximal reference

sequence มีความสัมพันธ์กับรูปแบบการเข้าถึงที่มีผู้ใช้มาก ("Hot" access pattern) ในบริการการจัดหาข้อมูลสารสนเทศ จากที่ได้กล่าวมาทั้งหมดสามารถสรุปโพรซีเจอร์ทั้งหมดสำหรับการไมนิ่งรูปแบบการเดินทางได้ดังนี้

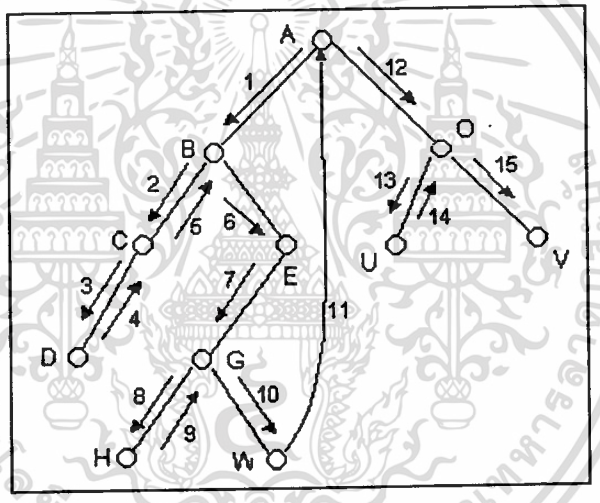
<u>โพรซีเจอร์สำหรับไมนิ่งรูปแบบการเดินทาง</u>

ขั้นที่ 1 : หา Maximal forward reference จากข้อมูลล็อกที่ผ่านการเตรียมรูปแบบข้อมูลแล้ว

ขั้นที่ 2 : หา Large reference sequence (L_k ,$k \geq 1$) จากชุดของ Maximal forward reference

ขั้นที่ 3 : หา Maximal reference sequence จาก Large reference sequence

เนื่องจากการดึง Maximal reference sequence จาก Large reference sequence (ในขั้นที่ 3) สามารถทำได้อย่างตรงไปตรงมา ดังนั้นนับจากนี้ไปเราจะสนใจให้ความสำคัญกับขั้นที่ 1 และ 2 และหาอัลกอริทึมสำหรับการหา large reference sequence ที่มีประสิทธิภาพ



รูปที่ 3.1 แสดงตัวอย่างของเส้นทางการเดินทาง

3.2 อัลกอริทึมสำหรับหารูปแบบเส้นทางการเดินทาง

ในหัวข้อต่อไปนี้จะอธิบายอัลกอริทึม MF ที่ใช้ในการแปลงลำดับของการเดินทางเดิมให้เป็นชุดของ maximal forward reference จากนั้นจะทำการหา large reference sequence โดยจะเสนออัลกอริทึม full-scan (FS) ซึ่งใช้หลักการของอัลกอริทึม DHP

3.2.1 การหา Maximal forward reference

หลังจากการทำการเตรียมข้อมูลแล้วฐานข้อมูลล็อกของการเดินทางจะบรรจุคู่ของ (ต้นทาง, ปลายทาง) ของแต่ละการเดินทางที่เชื่อมต่อกัน สำหรับการเริ่มต้นของเส้นทางใหม่ซึ่งไม่ได้ถูกเชื่อมต่อกับการเดินทางก่อนหน้า โหนดต้นทางจะเป็น null โดยให้ลำดับของการเดินทาง $\{(s_1,d_1), (s_2,d_2), \ldots, (s_n,d_n)\}$ ของผู้ใช้ เราจะทำการจับมันไปเป็นชุดลำดับหลาย ๆ ชุดลำดับ โดยแต่ละชุดลำดับจะแสดงถึง maximal forward reference อัลกอริทึมสำหรับการหา Maximal forward reference ทุก ๆ ตัวจากลำดับของการเดินทางมีการทำงานดังนี้คือ ขั้นแรกฐานข้อมูลล็อกของการเดินทางจะถูกทำการกรองข้อมูลที่ไม่เกี่ยวข้องออก (Data cleaning), ระบุผู้ใช้ (User identification) ,ระบุเซสชัน (Session identification), ทำเส้นทางการเดินทางให้สมบูรณ์ (Path completion) และการแปลงรูปแบบข้อมูลให้เหมาะสม (Formatting) ผลลัพธ์ที่ได้จะได้เส้นทางการเดินทาง $\{(s_1,d_1), (s_2,d_2), \ldots, (s_n,d_n)\}$ สำหรับแต่ละผู้ใช้ โดยที่คู่ของ (s_i,d_i) ถูกเรียงลำดับตามเวลา จากนั้นอัลกอริทึม MF จะถูกประยุกต์ใช้กับแต่ละเส้นทางของผู้ใช้ในการหา Maximal forward reference ทุก ๆ ตัวของมัน และผลลัพธ์ที่ได้จะนำไปเก็บไว้ใน D_F ซึ่งก็คือฐานข้อมูลที่ใช้เก็บทุก ๆ Maximal forward reference

ตัวอย่างจากการเดินทางในรูปที่ 3.1 สามารถพบได้ว่าการเข้าถึงย้อนกลับที่เกิดขึ้นครั้งแรกพบในการเคลื่อนที่ครั้งที่ 4 (จาก D ไปยัง C) ณ จุดนี้ Maximal forward reference ABCD ถูกเขียนลงใน D_F (โดยสเต็ป 3) ในการเคลื่อนที่ถัดมา (จาก C ไปยัง B) แม้ว่าเงื่อนไขแรกในสเต็ป 3 จะเป็นจริง แต่จะไม่มีการเขียนลงใน D_F เนื่องจาก flag = 0 ซึ่งหมายถึงเป็นการเดินทางย้อนกลับ การเข้าถึงไปข้างหน้าที่ตามมาจะใส่ ABEGH ลงใน string Y และจากนั้นจะถูกเขียนลงใน D_F เมื่อมีการพบการเข้าถึงย้อนกลับ (จาก Hไปยัง G) รูปแบบของการทำงานโดยอัลกอริทึม MF โดยมีข้อมูลในรูปที่ 3.1 เป็นอินพุต แสดงไว้ในตารางที่ 3.1

Step 1: Set i = 1 and string Y to null for initialization , where string Y is used to store the current forward reference path. Also, set the flag F=1 to indicate a forward traversal.

Step 2: Let A = s_i and B = d_i .

If A is equal to null then

/* this is the beginning of a new traversal */

begin

Write out the current string Y (if not null) to the database D_F ;

Set string Y = B;

Go to Step 5.

end

Step 3: If B is equal to some reference (say the j-th reference) in string Y then

/* this is a cross-referencing back to previous reference */

begin

If F is equal to 1 then write out string Y to database D_F ;

Discard all the references after the j-th one in string Y;

set F=0

Go to Step 5.

end

Step 4: Otherwise, append B to the end of string Y.

/* we are continuing a forward traversal */

If F is equal to 0 , set F = 1.

Step 5: Set i = i+1 . If the sequence is not completed scanned then go to Step 2.

รูปที่ 3.2 อัลกอริทึม MF

move	string Y	output to D_F
1	*AB*	-
2	*ABC*	-
3	*ABCD*	-
4	*ABC*	*ABCD*
5	*AB*	-
6	*ABE*	-
7	*ABEG*	-
8	*ABEGH*	-
9	*ABEG*	*ABEGH*
10	*ABEGW*	-
11	*A*	*ABEGW*
12	*AO*	-
13	*AOU*	-
14	*AO*	*AOU*
15	*AOV*	*AOV*(end)

ตารางที่ 3.1 ตัวอย่างการทำงานโดยอัลกอริทึม MF

3.2.2 การหา Large reference sequence

เมื่อฐานข้อมูลที่บรรจุทุก ๆ Maximal forward reference ของผู้ใช้ทุกคน (D_F) ถูกสร้างขึ้น เราสามารถหารูปแบบการเดินทางที่เกิดขึ้นบ่อยโดยการกำหนดความถี่ของการปรากฏลำดับการเข้าถึงใน D_F ลำดับ $s_1,\ldots,s_n$ สามารถบอกได้ว่าบรรจุลำดับที่ตามกันมา $r_1,\ldots,r_n$ ถ้ามีการเกิดของ i ที่ทำให้ $s_{i+j} = r_j$ สำหรับ $1 \leq j \leq k$ ถ้ามีจำนวนของ Maximal forward reference ใน D_F ที่บรรจุ $r_1,\ldots,r_n$ มีจำนวนเพียงพอแล้วลำดับของ k-reference $(r_1,\ldots,r_n)$ สามารถถูกเรียกได้ว่า Large reference sequence

ในส่วนต่อไปจะทำการอธิบายอัลกอริทึม Full-Scan (FS) ซึ่งใช้สำหรับการไมนิ่งรูปแบบการเดินทาง โดยอัลกอริทึมนี้จะใช้หลักการของ DHP (standing for Direct Hashing and Pruning) ดังนั้นก่อนที่จะมีการอธิบายอัลกอริทึม MF เราจะอธิบายแนวคิดของอัลกอริทึม DHP ก่อน

3.2.2.1 อัลกอริทึม DHP

DHP มีคุณลักษณะที่สำคัญ 2 ประการในการหากฎของสิ่งที่สัมพันธ์กัน ประการแรกคือมีการสร้างวิวัฒนาการของ Large itemset ที่มีประสิทธิภาพ และอีกประการหนึ่งคือมีการลดขนาดของฐานข้อมูลหลังจากการแสกนในแต่ละครั้งอย่างมีประสิทธิภาพ โดยใช้เทคนิคแฮชชิง (Hashing) ทำให้ DHP มีประสิทธิภาพในการวิวัฒนาการของ Candidate itemset อย่างมาก โดยเฉพาะอย่างยิ่งสำหรับ Large 2-itemset ทำให้เพิ่มประสิทธิภาพของทั้งกระบวนการทำงาน ยิ่งไปกว่านั้น DHP ยังใช้เทคนิคพรุนนิง (Pruning) ในการลดขนาดของฐานข้อมูล

การทำงานในแต่ละรอบเราใช้ชุดของ Large itemset (L_i)ในการทำชุดของ Candidate set C_{i+1} โดยการจอย L_i ด้วย L_i ($L_i * L_i$) จากนั้นเราจะแสกนฐานข้อมูลและนับค่าสนับสนุนของแต่ละ itemset ใน C_{i+1} เพื่อค้นหา L_{i+1} โดยทั่วไปแล้ว itemset ใน C_i ยิ่งมีมากเท่าใดแล้วทำให้มีการใช้เวลาการทำงานมากขึ้นในการหา L_i ในอัลกอริทึม Apriori ขั้นตอนในการหา L_2 จาก C_2 โดยการแสกนทั้งฐานข้อมูล การเทียบแต่ละทรานส์แอ็กชันกับแฮชทรี (Hash tree) ที่สร้างโดย C_2 จะกินเวลามาก โดยการสร้าง C_2 ที่เล็กลงแล้ว DHP สามารถสร้าง D_3 ที่เล็กลงมากเพื่อได้มาซึ่ง C_3 ถ้า C_2 ไม่เล็กลงแล้วฐานข้อมูลไม่สามารถถูกตบแต่งได้อย่างมีประสิทธิภาพ หลังจากผ่านขั้นนี้ไปแล้วขนาดของ L_i จะลดลงอย่างรวดเร็วตาม i ที่เพิ่มขึ้น เมื่อ L_i เล็กลงทำให้ C_{i+1} เล็กลง ดังนั้นทำให้เวลาของการทำงานที่สัมพันธ์กันลดลงด้วย

จุดสำคัญของ DHP ที่แสดงดังรูปที่ 3.3 มีการใช้เทคนิคของการแฮชชิงในการกรอง itemset ที่ไม่สำคัญสำหรับการสร้าง Candidate itemset ถัดไปทิ้ง เมื่อค่าสนับสนุนของ Candidate k-itemset ถูกนับโดยการแสกนฐานข้อมูล ในขณะเดียวกัน DHP ก็จะเก็บสะสมข้อมูลข่าวสารที่เกี่ยวกับ Candidate (k+1)-itemset ล่วงหน้า โดยใช้แนวทางที่ทุก ๆ (k+1)-itemset ที่เป็นไปได้ของแต่ละทรานส์แอ็กชันที่ผ่านการพรุนจะถูกแฮชไปยังตารางแฮช (Hash table) แต่ละบักเก็ต (butcket) ในตารางแฮชจะประกอบด้วยตัวเลขที่แสดงถึงว่ามี itemset จำนวนเท่าใดที่เคยถูกแฮชลงในบักเก็ตนี้ โดยผลของตารางแฮชจะสามารถสร้างบิทเวกเตอร์ออกมาได้ โดยบิท ๆ หนึ่งจะกำหนดค่าให้เป็น 1 ก็ต่อเมื่อจำนวนสมาชิกที่สัมพันธ์กันของตารางแฮชมีค่ามากกว่าหรือเท่ากับค่าสนับสนุน (s) ซึ่งเราจะได้เห็นต่อไปว่าบิทเวกเตอร์นี้สามารถถูกใช้ในการลดจำนวนของ itemset ใน C_i

ในรูปที่ 3.3 เป็นการนำเสนออัลกอริทึม DHP โดยเพื่อความง่ายในการนำเสนอจะมีการแบ่งออกเป็น 3 ส่วน

- <u>ส่วนที่ 1</u> สร้างชุดของ Large 1-itemset และสร้างตารางแฮช (H_2) สำหรับ 2-itemset

- <u>ส่วนที่ 2</u> สร้างชุดของ Candidate itemset C_k โดยอยู่บนพื้นฐานของตารางแฮช (H_k) ที่ทำการสร้างไว้ในรอบก่อนหน้านี้, หาชุดของ Large k-itemset (L_k) , ลดขนาดของฐานข้อมูลสำหรับ Large itemset ถัดมา, และทำตารางแฮชสำหรับ Candidate large (k+1)-itemset

- <u>ส่วนที่ 3</u> พื้นฐานเหมือนในส่วนที่ 2 ยกเว้นในส่วนนี้ไม่ใช้ตารางแฮชเนื่องจาก DHP มีประสิทธิภาพในการหา Large itemset ในช่วงต้น ๆ ได้ดีมาก ขนาดของ C_k จะลดลงอย่างมากในช่วงหลัง ดังนั้นไม่จำเป็นต้องมีการกรองก็ได้ ซึ่งเป็นเหตุผลสำคัญที่เราจะใช้ส่วนที่ 2 สำหรับการทำรอบแรก ๆ และใช้ส่วนที่ 3 สำหรับการทำรอบหลัง ๆ ที่จำนวนบักเก็ตของตารางแฮชที่มีค่ามากกว่าหรือเท่ากับ s ($\left|\{x|H_k[x] \geq s\}\right|$ ในส่วนที่ 2) มีจำนวนบักเก็ตน้อยกว่าเทรสโฮล LARGE ที่กำหนดไว้ล่วงหน้า ในส่วนที่ 3 โพรซีเจอร์ apriori_gen ที่ใช้ในการสร้าง C_{k+1} จาก L_k วิธีที่ใช้จะเหมือนที่ใช้ในอัลกอริทึม Apriori

หลังจากที่ผ่านการกำหนดในส่วนที่ 1 แล้ว ส่วนที่ 2 ประกอบด้วย 2 เฟส

- เฟสแรกทำการสร้างชุดของ Candidate k-itemset (C_k) โดยอยู่บนพื้นฐานของตารางแฮช H_k โดยใช้โพรซีเจอร์ gen_candidate การสร้างจะทำเหมือนใน Apriori ซึ่งสร้าง k-itemset โดยใช้ L_{k-1} แต่ที่ต่างกันคือ DHP จะใช้บิทเวกเตอร์ที่ทำการสร้างไว้ในรอบก่อนหน้าในการทดสอบการใช้ได้ของแต่ละ k-itemset คือแทนที่จะใช้ทุก ๆ k-itemset ที่สร้างมาจาก L_{k-1}*L_{k-1} กำหนดให้ C_k แต่ DHP จะเพิ่ม k-itemset ให้ C_k ก็ต่อเมื่อ k-itemset ได้ผ่านการกรองของแฮช ซึ่งการทำเช่นนี้สามารถลดขนาดของ C_k ได้อย่างมาก เมื่อทุก ๆ k-itemset ที่ได้ผ่านการกรองของแฮชแล้วจะถูกรวมเข้าไปใน C_k และเก็บไว้ในแฮชทรี โดยแฮชทรีนี้จะถูกใช้โดยแต่ละทรานส์แอ็กชันภายหลังเมื่อฐานข้อมูลถูกแสกนและนับค่าสนับสนุนของแต่ละ Candidate itemset

- เฟสที่สองของส่วนที่ 2 คือการนับค่าสนับสนุนของ Candidate itemset และลดขนาดของแต่ละทรานส์แอ็กชันโดยใช้โพรซีเจอร์ count_support การนับค่าสนับสนุนจะใช้ฟังก์ชัน subset ในการหาทุก ๆ Candidate itemset ที่บรรจุอยู่ในแต่ละทรานส์แอ็กชัน โดยทรานส์แอ็กชันในฐานข้อมูล (ที่ถูกลดทอนหลังจาก D_2) จะถูกแสกนครั้งละหนึ่งทรานส์แอ็กชันผลที่ได้คือ k-subset ของแต่ละทรานส์แอ็กชัน จากนั้นใช้ k-subset นั้นนับหาค่าสนับสนุนของ itemset ใน C_k

```
/*Part 1*/
s = a minimum support;
set all the buckets of H_2 to zero;                    /* hash table*/
forall transaction t ∈ D do begin
        insert and count 1-items occurrences in a hash tree;
        forall 2-subsets x of t do
                        H_2[h_2(x)]++;
end
L_1 = {c | c.count ≥ s, c exists in the leaf node of the hash tree} ;

/*Part 2*/
k=2;
D_k = D;                                   /* database for large k-itemset */
while (|{x | H_k[x] ≥ s}| ≥ LARGE) {              /* make a hash table */
        gen_candidate(L_{k-1}, H_k, C_k);
        set all the buckets of H_{k+1} to zero;
        D_{k+1} = φ;
                forall transactions t ∈ D_k do begin
                        count_support(t, C_k, k, f̂) ;        /* f̂ ⊆ t*/
                        if (|f̂| > k) then do begin
                                make_hasht (f̂, H_k, k, H_{k+1}, t') ;
                                if (|f̂| > k) then D_{k+1} = D_{k+1} ∪ {f̂} ;
                        end
                end
        L_k = {c ∈ C_k | c.count ≥ s};
        k++;
}
/* Part 3 */
gen_candidate (L_{k-1}, H_k, C_k) ;
while (|C_k| > 0) {
        D_{k+1} = φ;
        forall transactions t ∈ D_k do begin
                count_support (t, C_k, k, f̂) ;                /* f̂ ⊆ t */
                if (|f̂| > k) then D_{k+1} = D_{k+1} ∪ {f̂} ;
        end
        L_k = {c ∈ C_k | c.count ≥ s} ;
        if (|D_{k+1} = 0|) then break ;
        C_{k+1} = apriori_gen(L_k) ;
        k++ ;
}
```

รูปที่ 3.3 อัลกอริทึม DHP

Procedure gen_candidate (L_{k-1}, H_k, C_k)

$C_k = \phi$;

forall $c = c_p[1] \cdot \ldots \cdot c_p[k-2] \cdot c_p[k-1] \cdot c_q[k-1], c_p, c_q \in L_{k-1}, |c_p \cap c_q| = k-2$ **do**

if ($H_k[h_k(c)] \geq s$) **then**

$C_k = C_k \cup \{c\}$; /* insert c into hash-tree */

end Procedure

Procedure count_support ($t, C_k, k, \hat{t}$)

forall c such that $c \in C_k$ and c $(= t_{i1} .. t_{ik}) \in t$ **do begin**

$c.count++$;

for ($j = 1; j \leq k; j++$) $a[i_j]++$;

end

for ($i = 0, j = 0; i < |t|; i++$)

if ($a[i] \geq k$) **then do begin** $\hat{t}_j = t_i$; $j++$; **end**

end Procedure

Procedure make_hasht ($\hat{t}, H_k, k, H_{k+1}, \hat{t}$)

forall $(k+1)$-subsets $x(= \hat{t}_{i1} .. \hat{t}_{ik})$ of $\hat{t}$ **do**

if (for all k -subsets y of x , $H_k[h_k(y)] \geq s$) **then do begin**

$H_{k+1}[h_{k+1}(x)]++$;

for ($j = 1; j \leq k+1; j++$) $a[i_j]++$;

end

for ($i = 0, j = 0; i < |\hat{t}|; i++$)

if ($a[i] > 0$) **then do begin** $\hat{t}_j = \hat{t}_i$; $j++$; **end**

end Procedure

รูปที่ 3.4 โพรซีเจอร์ย่อยสำหรับอัลกอริทึม DHP

ตัวอย่างการสร้าง Candidate itemset โดย DHP แสดงไว้ในรูปที่ 3.6 ให้ Candidate set ของ Large 1-itemset $C_1 = \{\{A\},\{B\},\{C\},\{D\},\{E\}$ ทุก ๆ ทรานส์แอ็กชันของฐานข้อมูลถูกแสกนเพื่อนับค่าสนับสนุนของ C_1 ในขั้นตอนนี้แฮชทรีสำหรับ C_1 ถูกสร้างขึ้นมาลอย ๆ เพื่อจุดประสงค์ให้มีการนับที่มีประสิทธิภาพ DHP จะทดสอบว่าแต่ละไอเท็มมีในแฮชทรีแล้วหรือยัง ถ้ามีมันจะเพิ่มการนับของไอเท็มนั้นขึ้น 1 แต่ถ้ายังไม่มีจะทำการแทรกเพิ่มไอเท็มนี้ลงไปและใส่การนับเท่ากับ 1 ลงไปในแฮชทรีสำหรับแต่ละทรานส์แอ็กชัน หลังจากที่การนับการปรากฏของทุก ๆ i-subset แล้ว ทุก ๆ 2-subset ของทรานส์แอ็กชันนี้จะถูกสร้างและถูกแฮชไปในตารางแฮช H_2 โดยแต่ละ subset ที่ใส่ไปยังบักเก็ต i ทำให้ค่าของบักเก็ตนี้เพิ่มขึ้นครั้งละ 1 หลังจากฐานข้อมูลถูกแสกนแล้ว แต่ละบักเก็ตของตารางแฮชจะมีจำนวนของ 2-subset ที่อยู่ในบักเก็ตใน H_2 ดังรูปที่ 3.6 ถ้าให้ค่าสนับสนุนที่ต่ำที่สุดเท่ากับ 2 ฉะนั้นจะได้รับค่าบิทเวกเตอร์ {1,0,1,0,1;0,1} ซึ่งเราใช้บิทเวกเตอร์นี้ในการกรอง 2-

itemset ที่ได้จาก $L_1 * L_1$ ออก จะได้ $C_2 = \{\{AC\},\{BC\},\{BE\},\{CE\}\}$ แทนที่จะได้ $C_2 = \{\{AB\},\{AC\},\{AE\},\{BC\},\{BE\},\{CE\}\}$

การลดขนาดของฐานข้อมูล

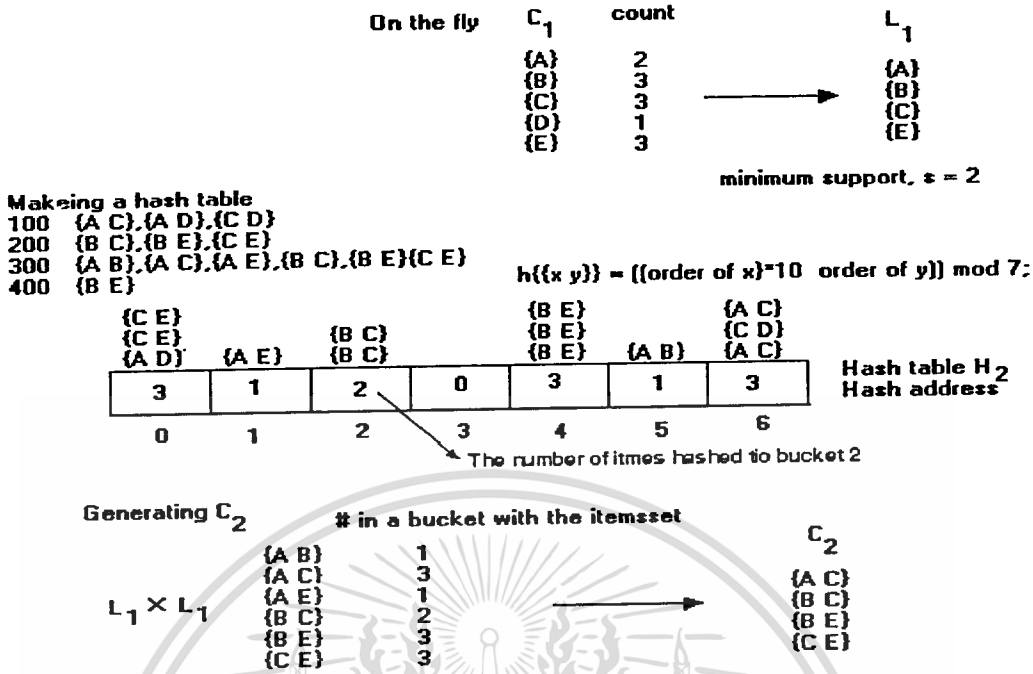
DHP สามารถลดขนาดของฐานข้อมูลลงโดยไม่เพียงแต่ใช้การตบแต่งแต่ละทรานส์แอ็กชัน แต่ยังรวมถึงการพรุนจำนวนของทรานส์แอ็กชันในฐานข้อมูลด้วย จากทฤษฎีที่ว่า subset ใด ๆ ของ Large itemset จะต้องเป็น Large itemset ด้วยตัวมันเอง ซึ่งนั่นก็คือถ้า $\{B,C,D\} \in L_3$ หมายความว่า $\{B,C\} \in L_2, \{B,D\} \in L_2$ และ $\{C,D\} \in L_2$ ด้วย จากข้อเท็จจริงนี้สามารถบอกได้ว่าทรานส์แอ็กชันที่ถูกใช้ในการหาชุดของ Large (k+1)-itemset ก็คือทรานส์แอ็กชันที่ประกอบด้วย Large k itemset ในรอบก่อนหน้านี้เท่านั้น ในการมองลักษณะนี้เราจะสามารถรู้ได้ว่าทรานส์แอ็กชันนี้พบเงื่อนไขของการบรรจุ Large (k+1)-itemset หรือไม่โดยดูจาก k-subset ของแต่ละทรานส์แอ็กชันที่ถูกนับตาม Candidate k-itemset จากการตบแต่งทรานส์แอ็กชันอย่างมีประสิทธิภาพและการลดจำนวนของทรานส์แอ็กชันโดยการกำจัดไอเท็มที่พบว่าไม่มีประโยชน์สำหรับ การหา Large item รุ่นต่อมา ทำให้มีจำนวนของ Candidate itemset ใกล้เคียงกับ Large itemset ของมันเมื่อนับ k-subset แล้ว

ถ้าทรานส์แอ็กชันบรรจุ Large (k+1)-itemset แล้วไอเท็มใด ๆ ที่บรรจุใน (k+1)-itemset นี้จะปรากฏอย่างน้อย k ของ Candidate k-itemset ใน C_k ด้วยผลลัพธ์นี้ไอเท็มในทรานส์แอ็กชัน t สามารถถูกตบแต่งออกไปถ้าไม่เป็นไปตามเงื่อนไขนี้ แนวคิดนี้ถูกใช้ในโพรซีเจอร์ count_support เพื่อตบแต่งขนาดของฐานข้อมูล ซึ่งจากที่กล่าวมาแล้วนั้นเป็นเพียงเงื่อนไขหนึ่งเท่านั้น ไม่ใช่เงื่อนไขทั้งหมด ในโพรซีเจอร์ make_hasht เราจะตรวจสอบอีกครั้งว่าแต่ละไอเท็มในทรานส์แอ็กชันถูกครอบคลุมโดย (k+1)-itemset (ของทรานส์แอ็กชัน) ที่ทุก ๆ k-itemset ของ (k+1)-itemset ถูกบรรจุใน C_k

Database D

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

รูปที่ 3.5 ตัวอย่างฐานข้อมูลทรานส์แอ็กชัน

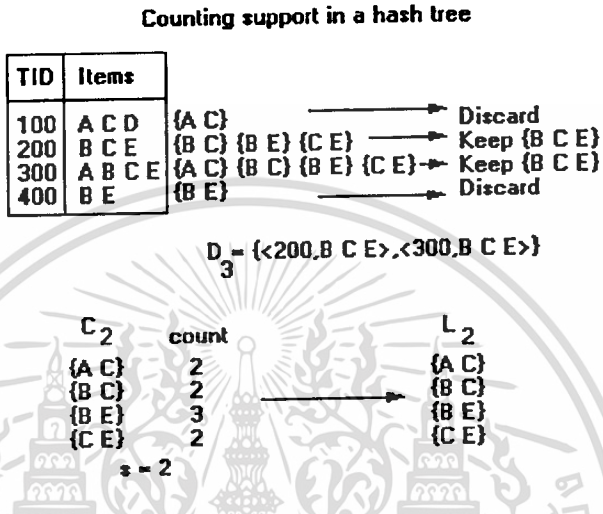


รูปที่ 3.6 ตัวอย่าง hash table และการวิวัฒนาการของ C_2

ตัวอย่างในการตบแต่งและลดทรานส์แอ็กชันถูกแสดงไว้ในรูปที่ 3.7 จากที่ทราบแล้วว่าค่าสนับสนุนของ k-itemset จะเพิ่มขึ้นตราบใดที่มันเป็น subset ของทรานส์แอ็กชัน t และยังเป็นสมาชิกของ C_k ด้วยการอธิบายโพรซีเจอร์ count_support ตัวแปร a[i] ถูกใช้ในการเก็บความถี่ในการปรากฏของแต่ละไอเท็มตัวที่ i ของทรานส์แอ็กชัน t เมื่อ k-subset ที่บรรจุไอเท็มตัวที่ i เป็นสมาชิกของ C_k เราจะเพิ่ม a[i] ขึ้น 1 (เช่น ในทรานส์แอ็กชัน 100 ,a[0] มีความสัมพันธ์กับ A, a[1] มีความสัมพันธ์กับ C และ a[2] มีความสัมพันธ์กับ D) จากในโพรซีเจอร์ make_hasht ก่อนการแฮชของ (k+1)-subset ของทรานส์แอ็กชัน $\hat{t}$ จะมีการทดสอบทุก ๆ k-subset ของ $\hat{t}$ โดยการตรวจสอบค่าของบักเก็ตในตารางแฮช H_k ที่มีความเกี่ยวข้อง ในการลดขนาดของทรานส์แอ็กชันจะมีการตรวจสอบแต่ละไอเท็ม $\hat{t}_i$ ใน $\hat{t}$ เพื่อดูว่า $\hat{t}_i$ ที่ถูกรวมเข้าไปใน (k+1)-subset สมควรสำหรับการแฮชจาก $\hat{t}$ หรือไม่ โดย $\hat{t}_i$ จะถูกละทิ้งไปถ้าไม่เป็นไปตามเงื่อนไข

ตัวอย่างเช่นในรูปที่ 3.7 ทรานส์แอ็กชัน 100 มีเพียง AC เป็น Candidate itemset ความถี่ของการปรากฏของทุก ๆ ไอเท็มคือ a[0]=1 ,a[1]=1 และ a[2]=0 โดยเรากำหนดให้ a[i] ที่มีค่าน้อยกว่า 2 ไม่มีประโยชน์สำหรับการสร้าง Large 3-itemset ดังนั้นจะตัดทรานส์แอ็กชันนี้ทิ้งไป ในทางกลับกันในทรานส์แอ็กชัน 300 มี 4 Candidate 2- itemset และความถี่ในการปรากฏของไอเท็มคือ a[0]=1, a[1]=2, a[2]=3 และ a[3]=2 ดังนั้นเราจะเก็บ BCE และตัด A ทิ้ง

ดังนั้นระหว่างการแสกนทรานส์แอ็กชันจะมีการตบแต่งทรานส์แอ็กชันหรือไม่ก็ลบทรานส์แอ็กชันนั้นออกไป โดยจะมีเพียงทรานส์แอ็กชันที่ประกอบด้วยสมาชิกที่สำคัญสำหรับการหา Large itemset ถัดไปเท่านั้นที่จะถูกเก็บไว้ใน D_{k+1} การทำเช่นนี้ทำให้ขนาดของฐานข้อมูลลดลงในแต่ละรอบ ซึ่งด้วยเหตุผลนี้เป็นสาเหตุให้ DHP มีเวลาในการทำงานที่สั้นกว่าอัลกอริทึมอื่น



รูปที่ 3.7 ตัวอย่างของ L_2 และ D_3

3.2.2.2 อัลการิทึม Full Scan (FS)

FS ใช้แนวคิดของ DHP ในการทำงานโดยให้ L_k แสดงถึงชุดของ large k-reference ทั้งหมดและ C_k เป็น superset ของ L_k โดยการแสกนทั้งฐานข้อมูล D_F, FS จะได้ L_1 และสร้างตารางแฮช (H_2) เพื่อนับจำนวนการปรากฏของแต่ละ 2-reference เริ่มที่ k=2 เหมือนใน DHP FS จะสร้าง C_k โดยอยู่บนพื้นฐานของตารางแฮชที่บรรจุการนับในรอบก่อนหน้า, หาชุดของ Large k-reference, ลดขนาดของฐานข้อมูลสำหรับรอบถัดไป และสร้างตารางแฮชเพื่อตัดสินหา Candidate (k+1)-reference ตามลำดับ จากการไม่นิ่งกฎของสิ่งที่สัมพันธ์กัน การทำชุดของ Candidate reference (C_k) สามารถถูกสร้างจากการจอย L_{k-1} ด้วยตัวมันเอง ซึ่งสามารถเขียนได้เป็น L_{k-1}*L_{k-1} อย่างไรก็ตาม เนื่องจากความแตกต่างระหว่างรูปแบบการเดินทางและกฎของสิ่งที่สัมพันธ์กัน ในอัลกอริทึมนี้ได้ทำการเปลี่ยนแปลงการสร้าง Candidate reference ดังนี้คือ สำหรับลำดับการเข้าถึง 2 ตัวใด ๆ ใน L_{k-1} ($r_1,\ldots,r_{k-1}$ และ $s_1,\ldots s_{k-1}$) จะมีการจอยทั้งสองลำดับการเข้าถึงเข้าด้วยกันเพื่อสร้าง k-reference ก็ต่อเมื่อ $r_1,\ldots,r_{k-1}$ บรรจุ $s_1,\ldots,s_{k-2}$ หรือ $s_1,\ldots,s_{k-1}$ บรรจุ $r_1,\ldots,r_{k-2}$ (เช่น หลังจากตัดอิลิเมนต์ตัวแรกในชุดลำดับแรกออกไปและตัดอิลิเมนต์สุดท้ายในอีกชุดลำดับออกแล้ว ผลลัพธ์ที่ได้คือ (k-2)-

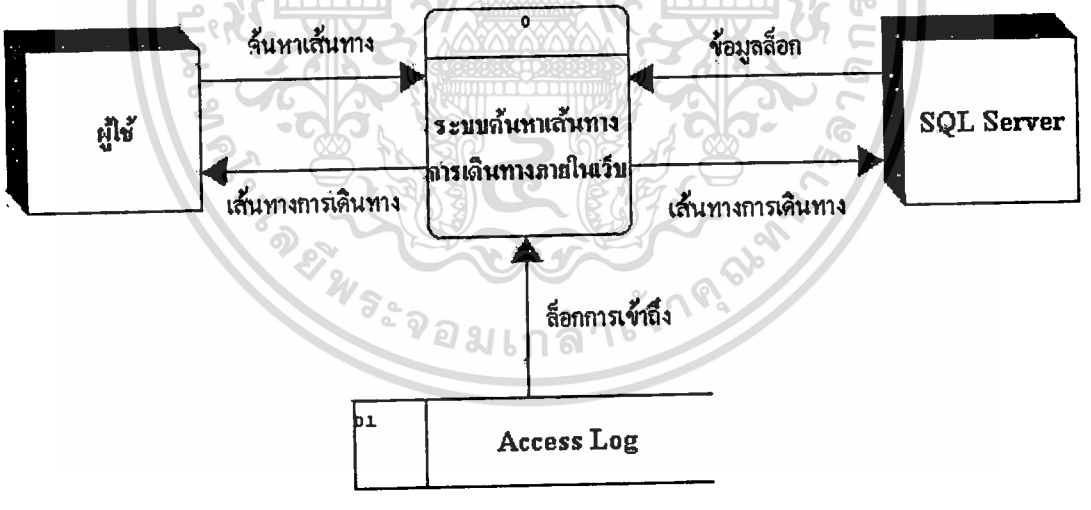
reference 2 ชุดลำดับ ที่เหมือนกัน) จากที่ทราบแล้วว่าเมื่อ k น้อย ๆ (โดยเฉพาะอย่างยิ่งในกรณี k=2) การหา C_k โดยการจอย L_{k-1} ด้วยตัวมันเองจะทำให้เกิด Candidate reference จำนวนมาก ดังนี้จึงใช้เทคนิคแฮชชิงดังที่อธิบายไว้อัลกอริทึม DHP เข้ามาช่วย แต่เมื่อ k เพิ่มขึ้น ขนาดของ $L_{k-1}*L_{k-1}$ จะมีจำนวนลดลงอย่างมาก เพราะฉะนั้นจึงสามารถสร้าง C_k ได้โดยตรงจาก $L_{k-1}*L_{k-1}$ โดยไม่ต้องมีการแฮชชิง หลังจากถึงรอบที่กำหนดซึ่งเป็นหลักการที่ใช้ใน DHP ด้วยเช่นกัน

ในการนับการปรากฏของ k-reference ใน C_k เพื่อตัดสินหา L_k จะต้องมีการแสกนผ่านฐานข้อมูล D_F ที่ถูกตบแต่งแล้ว หลังจากที่แสกนทั้งฐานข้อมูลแล้ว k-reference ใดใน C_k ที่มีการนับที่เกินค่าเทรสโฮลที่กำหนดจะกลายมาเป็น L_k ถ้า L_k ยังมีอยู่จะต้องทำต่อยังรอบถัดไป (รอบ k+1) ซึ่งหลักการจะเหมือนใน DHP ที่ทุก ๆ ครั้งที่ฐานข้อมูลถูกแสกนฐานข้อมูลจะถูกตบแต่งโดย FS เพื่อเพิ่มประสิทธิภาพของการแสกนในอนาคต

บทที่ 4
การออกแบบโปรแกรมและฐานข้อมูล

4.1 การออกแบบโปรแกรม

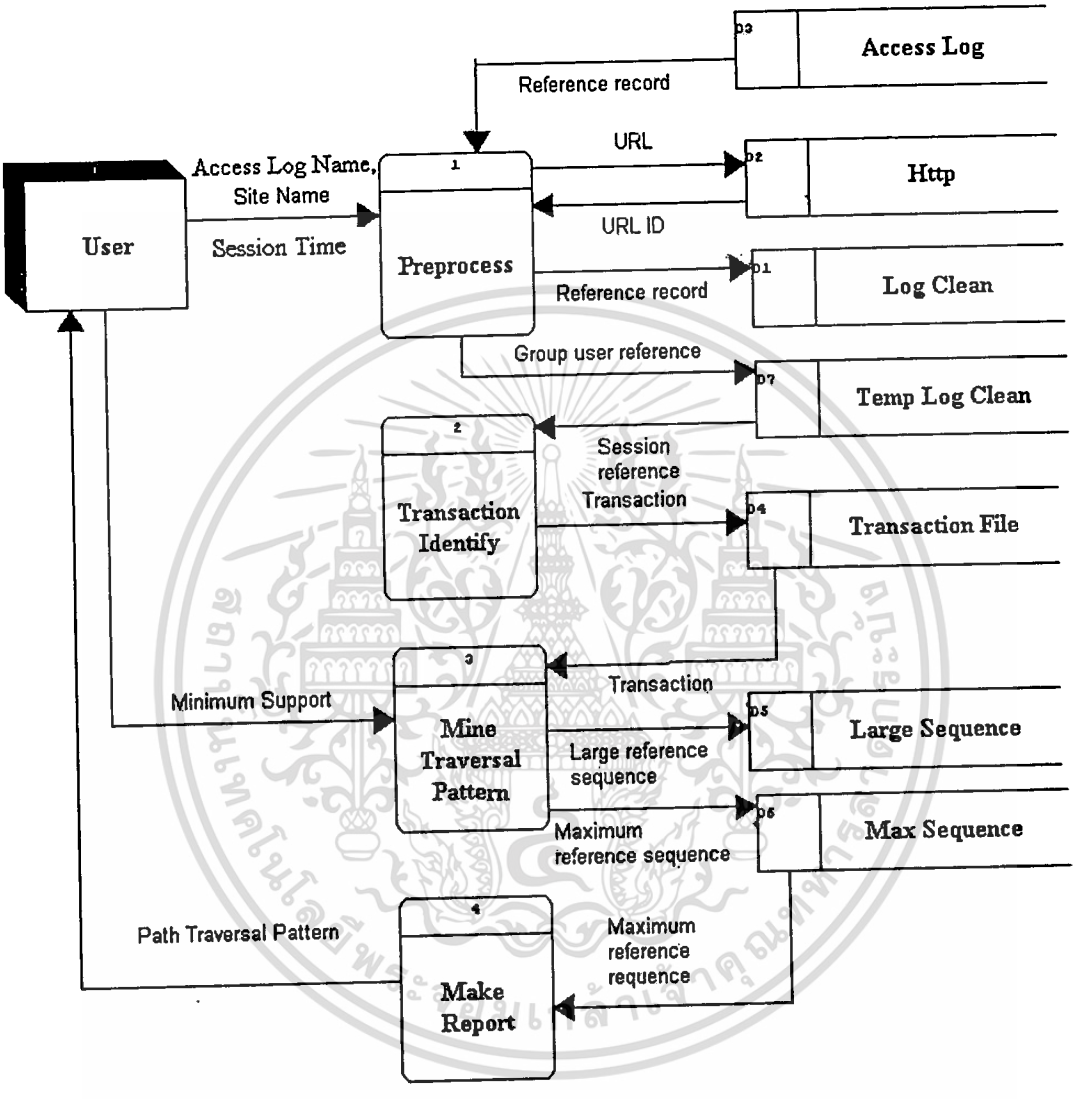
จากการศึกษา วิเคราะห์วิธีการหาเส้นทางการเดินทางภายในเว็บ จึงได้ออกแบบโปรแกรมให้มีการทำงานโดยออกแบบลักษณะการทำงานของโปรแกรมออกแบบ 2 ส่วน คือส่วนที่ใช้สำหรับเตรียมข้อมูล Access log ที่จะใช้เป็นข้อมูลอินพุตสำหรับการหาเส้นทางการเดินทางภายในเว็บ โดยข้อมูลที่ผ่านการจัดเตรียมเรียบร้อยนี้จะถูกเก็บไว้ในฐานข้อมูลเพื่อใช้สำหรับส่วนถัดไป อีกส่วนหนึ่งคือส่วนการทำการค้นหาเส้นทางการเดินทางภายในเว็บโดยใช้อัลกอริทึมที่ได้ศึกษามาเครื่องมือสำหรับไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บที่ได้ออกแบบนี้สามารถเขียนเป็น context diagram และ data flow diagram ได้ดังแสดงในรูปที่ 4.1 และรูปที่ 4.2 ตามลำดับ



รูปที่ 4.1 Context Diagram ของเครื่องมือสำหรับไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บ

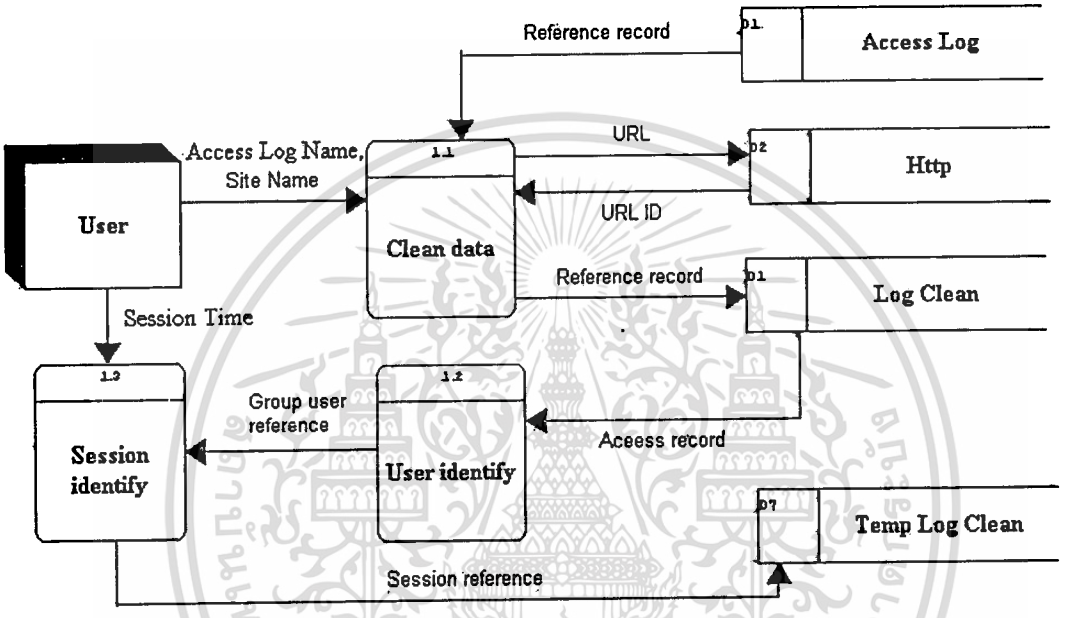
จากรูปผู้ใช้จะเป็นผู้กำหนดไฟล์ Access log ที่จะนำมาใช้หาเส้นทางการเดินทาง กำหนดเว็บไซต์ที่สนใจและกำหนดค่า minimum support ที่ใช้ในการกรองเส้นทางการเดินทางที่น่าสนใจเท่านั้น โดยข้อมูลที่นำมาใช้ในการหาเส้นทางการเดินทางจะถูกจัดเตรียมก่อนเพื่อให้มีรูปแบบที่

เหมาะสม ข้อมูลเหล่านี้ถูกจัดเก็บไว้ในฐานข้อมูลเพื่อความง่ายในการ query งานย่อยซึ่งมีอยู่ในระบบสามารถแตกออกได้ดังรูปที่ 4.2



รูปที่ 4.2 Data Flow Diagram Level 1 ของเครื่องมือสำหรับไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บ

จากรูปที่ 4.2 งานย่อยที่แตกออกมาประกอบไปด้วย process ต่าง ๆ 4 ส่วน ในโปรแกรมส่วนที่เป็น process หมายเลข 1, 2 เป็นงานย่อยของการทำการจัดเตรียมข้อมูลก่อนการเข้าไปสู่กระบวนการไมนิ่ง ส่วน process หมายเลข 3 เป็น process ของการทำไมนิ่งเพื่อหาเส้นทางการเดินทางภายในเว็บ และส่วนสุดท้าย process หมายเลข 4 เป็นส่วนการนำเสนอผลลัพธ์แก่ผู้ใช้



รูปที่ 4.3 Data Flow Diagram Level 2 ของ Process 1.0 Preprocess

process ที่ 1 เป็นการเตรียมข้อมูล รายละเอียดของ process นี้ได้แสดงไว้ในรูปที่ 4.3 โดยแตกออกเป็นงานย่อยได้ 3 งานคืองานทำความสะอาดข้อมูลคือการตัดข้อมูลที่ไม่เกี่ยวข้องกับการหาเส้นทางการเดินทางภายในเว็บทิ้งไป งานการระบุผู้ใช้ และงานระบุเซสชันโดยงานนี้จะต้องได้รับค่าเวลาเซสชันจากผู้ใช้เพื่อที่จะใช้ในการตัดหาเซสชันต่อไป ผลลัพธ์ของ process นี้จะถูกเก็บไว้ในตาราง "Http", "Log Clean" และ "Temp Log Clean"

process ที่ 2 การระบุทรานแอ็กชันโดยใช้อัลกอริทึมที่ได้ศึกษามา ผลลัพธ์ถูกเก็บไว้ใน Transaction file

process ที่ 3 เป็นการค้นหาเส้นทางการเดินทางภายในเว็บโดยใช้อัลกอริทึม FS ซึ่งต้องมีการรับค่าสนับสนุนที่น้อยที่สุดจากผู้ใช้เพื่อใช้ในการกรองเฉพาะผลลัพธ์ที่สนใจเท่านั้น ผลลัพธ์ถูกเก็บไว้ในตาราง "Large Sequence" และ "Max Sequence"

process ที่ 4 นำผลลัพธ์นำเสนอแก่ผู้ใช้

4.2 ข้อมูลอินพุต

ในการค้นหาเส้นทางการเดินทางภายในเว็บนั้นสามารถนำข้อมูลทั้งจากเว็บเซิร์ฟเวอร์หรือพร็อกซีเซิร์ฟเวอร์มาใช้เป็นข้อมูลอินพุตให้แก่ระบบได้ แต่ในการศึกษาครั้งนี้เราจะนำเอาข้อมูลจากพร็อกซีเซิร์ฟเวอร์มาใช้เนื่องจากข้อมูลนี้ทางสถาบันได้ทำการจัดเก็บไว้อยู่แล้ว

การทำงานของระบบที่มีพร็อกซีเซิร์ฟเวอร์ติดตั้งอยู่จะใช้หลักการของไคลเอนต์/เซิร์ฟเวอร์ (Client/Server) ที่มีตัวเครื่องพร็อกซีเซิร์ฟเวอร์เป็นผู้ให้บริการแก่ไคลเอนต์ซึ่งในที่นี้คือ HTTP ไคลเอนต์ โดยบริการในที่นี้หมายถึงการเป็นตัวแทนของไคลเอนต์เหล่านั้นในการไปเรียกข้อมูลจากเครื่องเว็บเซิร์ฟเวอร์ (Web Server) มาให้แก่ไคลเอนต์ตามที่ได้รับการร้องขอมา ทั้งนี้พร็อกซีเซิร์ฟเวอร์ยังสามารถทำการจัดเก็บข้อมูลที่ได้มาเหล่านั้นไว้ชั่วระยะเวลาหนึ่ง (Cache) ซึ่งถ้ามีการร้องขอข้อมูล (ออบเจกต์) เดียวกันเข้ามา พร็อกซีเซิร์ฟเวอร์ก็สามารถนำออบเจกต์ที่จัดเก็บไว้ส่งให้ไคลเอนต์ได้เลยโดยไม่ต้องไปดึงมาจากภายนอกอีก ในกรณีนี้จะเรียกว่าพบ (HIT) ออบเจกต์ในแคช (Cache) แต่ถ้าออบเจกต์ที่ไคลเอนต์ร้องขอมาไม่มีอยู่ในแคชหรือกรายเป็นออบเจกต์ใหม่ (เกิดจากกรณีที่ออบเจกต์มีการเปลี่ยนแปลงขนาดและ/หรือวันที่สร้าง) ก็จะเกิดกรณีไม่พบ (MISS) และตัวพร็อกซีเซิร์ฟเวอร์ก็จะต้องไปดึงข้อมูลจากภายนอกเข้ามาให้กับไคล์เอนต์

เมื่อพร็อกซีเซิร์ฟเวอร์ทำงานตามการร้องขอจากไคล์เอนต์แต่ละรายการเสร็จก็จะทำการบันทึกผลการทำงานของการร้องขอนั้นลงสู่ Log file เพื่อแสดงรายละเอียดต่าง ๆ ซึ่งในโครงการพัฒนาระบบงานนี้จะใช้ไฟล์ชื่อ access.log ของ Squid เวอร์ชัน 2.3 สเตเบิล 2 ซึ่งทำงานเป็นพร็อกซีเซิร์ฟเวอร์ของ proxy.kmitl.ac.th ทำงานบนเครื่อง HP รุ่น LH4Plus Dual Pentium III 500 MHz โดยในไฟล์ access.log 1 บรรทัดจะประกอบด้วย 10 ฟิลด์ดังต่อไปนี้

Timestamp Elapsed Client-Addr Log-Tag/HTTP-Code

Size Req-Method URL ref931 Hierarchy/Hostname Content-Type

ในที่นี้จะขออธิบายความหมายเฉพาะฟิลด์ที่เกี่ยวข้องและนำมาใช้ในโครงการพัฒนาระบบงานนี้เท่านั้น ได้แก่

Timestamp คือค่าเวลาปัจจุบันที่ Squid จะบันทึกไว้เมื่อเสร็จสิ้นการร้องขอนั้น

Client-Address คือ IP Address ของผู้ใช้ที่ทำการร้องขอ

Req-Method คือวิธีการของการร้องขอเพื่อจะกระทำกับออบเจกต์ เช่น GET, POST เป็นต้น

URL คือ URL ที่ไคลเอนต์ร้องขอ

Content-Type คือประเภทของ URL ที่ทำการร้องขอ เช่น /text/html, /image/gif เป็นต้น

ตัวอย่างของไฟล์ access.log เป็นดังนี้

```
977387445.078 277 161.246.45.31 TCP_MISS/000 0 GET http://adforce.imgis.com/? - DIRECT/adforce.imgis.com -
977387445.079 1852 161.246.59.62 TCP_MISS/200 363 GET http://216.35.185.221/creatives_3.7.cgi? –
              DIRECT/216.35.185.221 text/html
977387445.084  33  161.246.51.68 TCP_MEM_HIT/200 481 GET http://chat.sanook.com/images/insertpic.gif –
              NONE/- image/gif
977387445.090  28 161.246.48.168 TCP_IMS_HIT/304 210 GET http://www.paidforsurf.com/images/shop.gif –
              NONE/- image/gif
```

รูปที่4.4 ตัวอย่างไฟล์ access.log

4.3 การออกแบบฐานข้อมูล

ฐานข้อมูลของระบบงานนี้ประกอบด้วยตารางหลัก ๆ 5 ตาราง ได้แก่ ตาราง "Http" และตาราง "Log_Clean" ทั้งสองตารางนี้จัดเก็บข้อมูลจากไฟล์ล็อกที่ผ่านการจัดเตรียมข้อมูลมาเรียบร้อยแล้ว โดยตาราง "http" จะเป็นตารางที่ใช้เปลี่ยน URL ที่เป็นลักษณะไฮเปอร์เทกซ์ให้เป็น ID ที่จะสามารถนำไปเข้ากระบวนการดาต้าไมนิ่งได้เร็วยิ่งขึ้น

ตาราง "Temp_LogClean" เป็นตารางที่ทำการกรองข้อมูลจากตาราง "Log_Clean" เฉพาะเงื่อนไขที่ผู้ใช้ต้องการเท่านั้นแล้วนำข้อมูลจากตารางนี้มาใช้ในการทำทรานแอ็กชันเพื่อจัดเตรียมเข้าสู่การทำไมนิ่ง ตาราง "Large_Sequence" และตาราง "Max_Sequence" ใช้จัดเก็บผลลัพธ์เส้นทางการเดินทางที่ได้จากการไมนิ่ง

รายละเอียดและความสัมพันธ์ของแต่ละฐานข้อมูลมีดังนี้

1. **Http** ใช้เก็บ ID ของแต่ละ URL มีรายละเอียดดังนี้

Field	Type	Description
Http_ID	Auto Number	หมายเลข ID ของ URL
Http_Name	Text(900)	URL

2. **Log_Clean** ใช้เก็บ log entry ที่ผ่านการทำความสะอาดจากไฟล์ล็อก มีรายละเอียดดังนี้

Field	Type	Description
Row	Auto Number	หมายเลข record
Timestamp	Text(25)	ค่าเวลาปัจจุบันที่ Squid บันทึกไว้เมื่อเสร็จสิ้นการร้องขอ
Address	Text(16)	หมายเลข IP Address ของแต่ละการร้องขอ
Http_ID	Text(100)	หมายเลข ID ของ URL

3. **Large_Sequence** ใช้เก็บผลลัพธ์เส้นทางการเดินทางที่ได้จากการทำไมนิ่ง (Large reference sequence) มีรายละเอียดดังนี้

Field	Type	Description
Large_Sequence	Text(1000)	เส้นทางการเดินทางภายในเว็บ
Large_Support	Number	ค่าสนับสนุนของเส้นทางการเดินทางนั้น

4. **Max_Sequence** ใช้เก็บ Maximum reference sequence ("Hot" access pattern) ที่ได้จากการวิเคราะห์ Large reference sequence

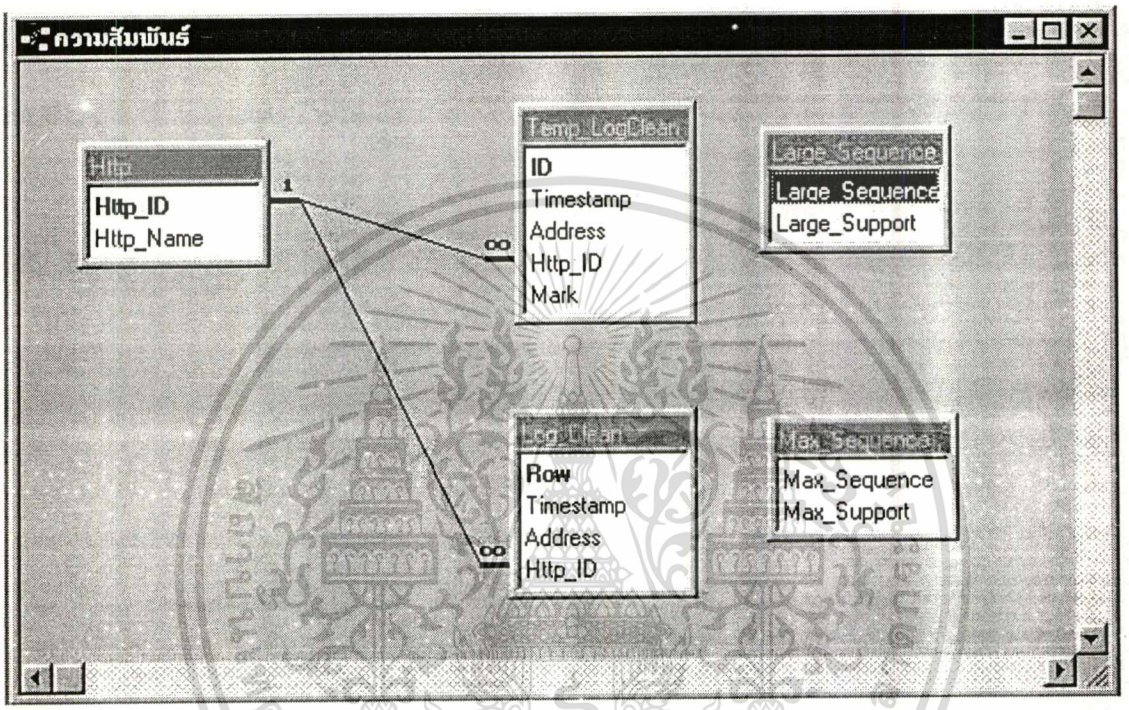
Field	Type	Description
Max_Sequence	Text(1000)	เส้นทางการเดินทางภายในเว็บ
Max_Support	Number	ค่าสนับสนุนของเส้นทางการเดินทางนั้น

5. **Temp_LogClean** ใช้เก็บข้อมูลที่จากตารางจง "Log_Clean" เฉพาะเงื่อนไขที่ผู้ใช้ต้องการเท่านั้น

Field	Type	Description
Timestamp	Text(25)	ค่าเวลาปัจจุบันที่ Squid บันทึกไว้เมื่อเสร็จสิ้นการร้องขอ
Address	Text(16)	หมายเลข IP Address ของแต่ละการร้องขอ
Http_ID	Text(100)	หมายเลข ID ของ URL
Mark	Yes/No	flag ในการ query

4.4 ความสัมพันธ์ของฐานข้อมูล

ความสัมพันธ์ระหว่างตาราง “Http” กับ “Log_Clean” และ “Temp_LogClean” เป็นความสัมพันธ์แบบ one to many ส่วนตาราง “Large_Sequence” และ “Max_Sequence” ไม่ได้สัมพันธ์กับตารางใด ๆ เนื่องจากเป็นตารางที่ใช้เก็บผลลัพธ์เท่านั้น ดังแสดงในรูปที่ 4.5



รูปที่ 4.5 ความสัมพันธ์ระหว่างตารางต่าง ๆ ในฐานข้อมูล

บทที่ 5
การพัฒนาโปรแกรม

5.1 หลักการทำงานของโปรแกรม

หลังจากที่ได้ทำการออกแบบโปรแกรมเรียบร้อยแล้ว จึงพิจารณาแบ่งส่วนการทำงานของโปรแกรมออกเป็น 2 ส่วนหลัก ๆ คือ ส่วนของการเตรียมข้อมูล และส่วนของการทำการไมนิ่งหารูปแบบเส้นทางการเดินทางภายในเว็บ ในการพัฒนาเครื่องมือสำหรับไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บของการศึกษานี้ได้ใช้ Visual Basic version 6.0 และ Microsoft SQL Server version 7.0 ในการพัฒนาโปรแกรม

5.1.1 ส่วนการเตรียมข้อมูล

ในส่วนแรกจะเป็นส่วนที่นำเอาข้อมูลดิบที่ได้จากไฟล์ access.log ของพร็อกซีเซร์ฟเวอร์มาทำการแปลงให้อยู่ในรูปแบบที่เหมาะสมแล้วจัดเก็บไว้ในฐานข้อมูลโดยการศึกษาครั้งนี้ได้ใช้ Microsoft SQL Server เวอร์ชัน 7.0 เป็นตัวจัดการฐานข้อมูล ส่วนการเตรียมข้อมูลมีขั้นตอนดังต่อไปนี้

- การทำความสะอาดข้อมูล (Data Cleaning) จะทำการกำจัดไอเท็มที่ไม่เกี่ยวข้องออกไปเพื่อให้ได้ภาพที่ถูกต้องแม่นยำของการเข้าถึงของผู้ใช้ที่แท้จริงเท่านั้น เนื่องจากโปรโตคอล HTTP มีความต้องการการเชื่อมต่อที่แยกออกจากกันของแต่ละไฟล์ที่ถูกร้องขอจากเซร์ฟเวอร์ ดังนั้นการร้องขอของผู้ใช้เพื่อที่จะดูเพจหนึ่ง ๆ จะทำให้เกิดผลลัพธ์ในล็อกหลาย entry เนื่องจากมีการดาวน์โหลดพวกกราฟฟิกหรือสคริปต์ต่าง ๆ เข้ามาเพิ่มเติมจากเพจที่ผู้ใช้ต้องการจริง ๆ เนื่องจากในที่นี้เราต้องการที่จะค้นหาพฤติกรรมการเดินทางของผู้ใช้จริง ๆ ซึ่งก็คือการร้องขอเพจที่เกิดจากผู้ใช้ไม่ใช่เกิดจากเทรกในเพจ ดังนั้นเราจึงกำจัดไอเท็มที่ไม่เกี่ยวข้องออกไปโดยการตรวจสอบที่ฟิลด์ Content-Type และ Req.-Mehod ของไฟล์ access.log ร่วมกับการตรวจสอบส่วนต่อท้าย (suffix) ของชื่อ URL ซึ่งในการทำงานของโปรแกรมนี้จะเลือกเฉพาะไอเท็มที่ Content-Type เป็นชนิด /text/... , Req.-Method เป็น "GET" และมีส่วนต่อท้ายที่ไม่ใช่พวกกราฟฟิกหรือสคิปต์ไฟล์เท่านั้นมาใช้งาน ผลลัพธ์จากการทำความสะอาดข้อมูลดิบในรูปที่ 5.1 ได้ถูกแสดงไว้ในรูปที่ 5.2

- การระบุผู้ใช้ (User Identification) เนื่องจากในการศึกษาครั้งนี้ไม่มีการคิดผลกระทบที่เกิดจากแคช (Cache) , ไฟร์วอลล์ (Firewall) และพร็อกซีเซิร์ฟเวอร์ (Proxy Server) เพราะฉะนั้นเราจึงทำการระบุผู้ใช้โดยการดูจากฟิลด์ Client-Addr ซึ่งบันทึก IP Address เท่านั้น

- การระบุเซสชัน (Session Identification) เนื่องจาก access log ของเรามีการเก็บไฟล์ละ 1 วันซึ่งเป็นคาบเวลาที่ยาว มีความเป็นไปได้อย่างมากที่ผู้ใช้จะกลับเข้ามาเยี่ยมชมเว็บไซต์มากกว่า 1 ครั้ง โดยจุดมุ่งหมายของการระบุเซสชันคือต้องการแบ่งการเข้าถึงเพจของแต่ละผู้ใช้ไปเป็นเซสชัน โดยการระบุเซสชันเราจะใช้วิธี timeout ซึ่งในการศึกษาครั้งนี้จะให้ผู้ใช้เป็นผู้ระบุ timeout ที่ใช้ในการแบ่งเซสชัน

- การทำเส้นทางให้สมบูรณ์ (Path Completion) เนื่องจากเราไม่คิดผลกระทบที่เกิดจากการกดปุ่ม "Back" บนบราวเซอร์ ดังนั้นจึงตั้งสมมติฐานว่าในไฟล์ access.log จะเก็บทุก ๆ การร้องขอของผู้ใช้ เพราะฉะนั้นจึงไม่มีการทำงานในส่วนนี้

- การจัดรูปแบบ (Formatting) ส่วนสุดท้ายของการเตรียมข้อมูลคือการจัดรูปแบบให้เหมาะสมโดยในที่นี้เราจะใช้อัลกอริทึม MF ในสร้างทรานส์แอ็กชันที่บรรจุเส้นทางการเดินทางไปข้างหน้าที่ไกลที่สุด (Maximum forward reference) โดยทรานส์แอ็กชันนี้เป็นทรานแอ็กชันแบบเพจเนื้อหา-เพจสนับสนุน (Content-Navigation Transaction)

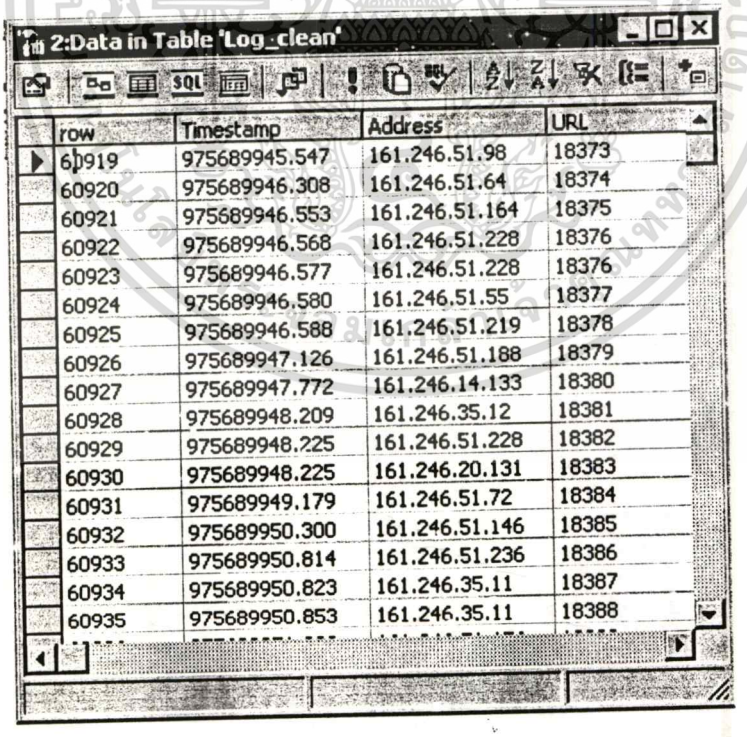
```
977387445.079  161.246.59.62  GET http://216.35.185.221/creatives_3.7.cgi? - DIRECT/216.35.185.221
text/html
977387445.084  161.246.51.68  GET http://chat.sanook.com/images/insertpic.gif - NONE/- image/gif
977387445.085  161.246.45.31 GET http://adforce.imgis.com/? - DIRECT/adforce.imgis.com -
977387445.090  161.246.48.168  GET http://www.paidforsurf.com/images/shop.gif - NONE/- image/gif
977387445.141  161.246.27.233  GET http://pantip.inet.co.th/ads/cafe/banner_ratchada.shtml -
DIRECT/pantip.inet.co.th text/html
977387445.142  161.246.37.28 GET http://thai.to/sonya/Gallery_Ploy/dol04_small.jpg - DIRECT/thai.to
image/jpeg
977387445.153  161.246.59.62 GET http://www.paidforsurf.com/images/shop.gif - NONE/- image/gif
977387445.174  161.246.51.228  GET http://master.cpe.ku.ac.th/~g4365017/cs10/chaten/chat.cgi? -
DIRECT/master.cpe.ku.ac.th text/html
```

รูปที่ 5.1 ตัวอย่างข้อมูลล็อก

```
977387445.079  161.246.59.62  GET http://216.35.185.221/creatives_3.7.cgi? - DIRECT/216.35.185.221
text/html
977387445.085  161.246.45.31 GET http://adforce.imgis.com/? - DIRECT/adforce.imgis.com -
977387445.141  161.246.27.233  GET http://pantip.inet.co.th/ads/cafe/banner_ratchada.shtml -
DIRECT/pantip.inet.co.th text/html
977387445.174  161.246.51.228  GET http://master.cpe.ku.ac.th/~g4365017/cs10/chaten/chat.cgi? -
DIRECT/master.cpe.ku.ac.th text/html
```

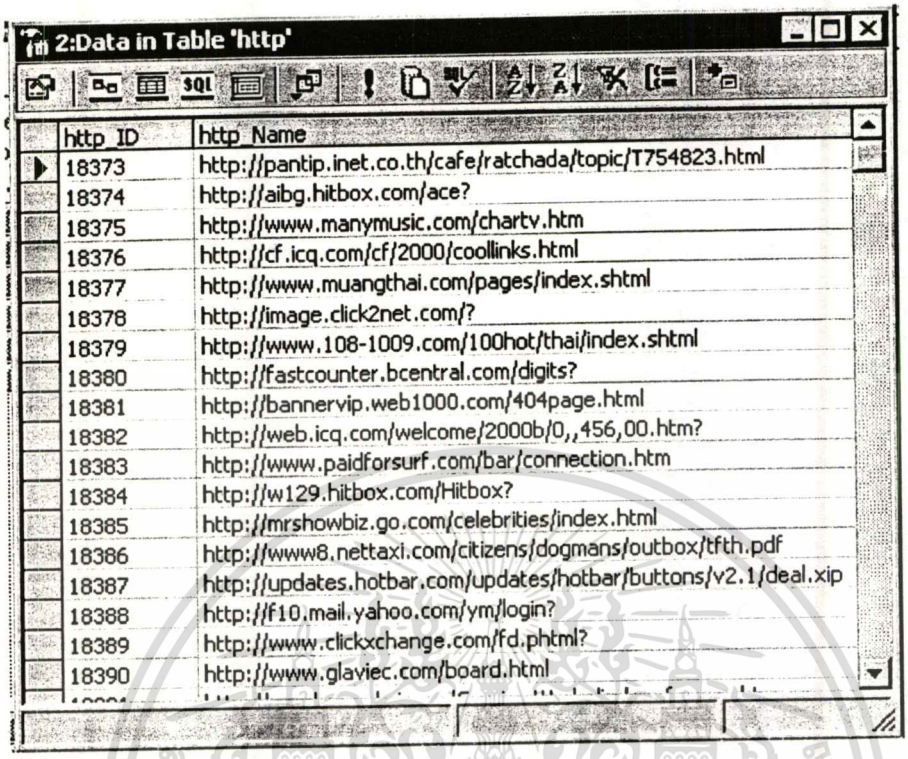
รูปที่ 5.2 ข้อมูลล็อกหลังจากการทำความสะอาด

การเตรียมข้อมูลจะทำโดยการอ่านข้อมูลทั้งจาก access.log แล้วนำมาทำความสะอาดจากนั้นเก็บผลลัพธ์ไว้ในฐานข้อมูล โดยจะทำการเก็บ URL เป็น ID เพื่อความง่ายในการแปลงข้อมูลเข้าสู่การทำในส่วนต่อไปตัวอย่างตาราง "Log_Clean" และ "Http" ที่ใช้เก็บผลลัพธ์ในส่วนนี้ได้แสดงไว้ในรูปที่ 5.3 และ 5.4 ตามลำดับ



row	Timestamp	Address	URL
60919	975689945.547	161.246.51.98	18373
60920	975689946.308	161.246.51.64	18374
60921	975689946.553	161.246.51.164	18375
60922	975689946.568	161.246.51.228	18376
60923	975689946.577	161.246.51.228	18376
60924	975689946.580	161.246.51.55	18377
60925	975689946.588	161.246.51.219	18378
60926	975689947.126	161.246.51.188	18379
60927	975689947.772	161.246.14.133	18380
60928	975689948.209	161.246.35.12	18381
60929	975689948.225	161.246.51.228	18382
60930	975689948.225	161.246.20.131	18383
60931	975689949.179	161.246.51.72	18384
60932	975689950.300	161.246.51.146	18385
60933	975689950.814	161.246.51.236	18386
60934	975689950.823	161.246.35.11	18387
60935	975689950.853	161.246.35.11	18388

รูปที่ 5.3 ตัวอย่างข้อมูลตาราง "Log Clean"



2:Data in Table 'http'

http_ID	http_Name
18373	http://pantip.inet.co.th/cafe/ratchada/topic/T754823.html
18374	http://aibg.hitbox.com/ace?
18375	http://www.manymusic.com/chartv.htm
18376	http://cf.icq.com/cf/2000/coollinks.html
18377	http://www.muangthai.com/pages/index.shtml
18378	http://image.click2net.com/?
18379	http://www.108-1009.com/100hot/thai/index.shtml
18380	http://fastcounter.bcentral.com/digits?
18381	http://bannervip.web1000.com/404page.html
18382	http://web.icq.com/welcome/2000b/0,,456,00.htm?
18383	http://www.paidforsurf.com/bar/connection.htm
18384	http://w129.hitbox.com/Hitbox?
18385	http://mrshowbiz.go.com/celebrities/index.html
18386	http://www8.nettaxi.com/citizens/dogmans/outbox/tfth.pdf
18387	http://updates.hotbar.com/updates/hotbar/buttons/v2.1/deal.xip
18388	http://f10.mail.yahoo.com/ym/login?
18389	http://www.clickxchange.com/fd.phtml?
18390	http://www.glaviec.com/board.html

รูปที่ 5.4 ตัวอย่างข้อมูลตาราง “Http”

จากนั้นนำข้อมูลตาราง “Log_Clean” มาทำการระบุผู้ใช้และระบุเซสชัน ผลลัพธ์คือเซสชันทั้งหมดของแต่ละผู้ใช้เก็บไว้ในเซสชันไฟล์ ตัวอย่างเซสชันไฟล์แสดงไว้ดังรูปที่ 5.5 โดยแบ่งเซสชันละ 30 นาที แต่ละบรรทัดแสดงเซสชันของผู้ใช้ โดยผู้ใช้คนหนึ่งมีได้หลายเซสชัน

```
,5178,14993,4352,4351,16086,3602,5178,5178,5178,17132,5178
,5178,5178,5178,3783,4195,5178,5178,5178,18792,4256,6103,5178
,5178,20519,3584,3778,3802,5178,5292,3602,5178,5178,8153,5178,3602,21220,5661,5178,5178
,5178,8127,5178,26054,8127,5178,3602,3602,5178,5178,25460,25461,25472,5178,5178
```

รูปที่ 5.5 ตัวอย่างเซสชันไฟล์

เมื่อได้ไฟล์เซสชันแล้วจะนำข้อมูลที่ได้จากไฟล์นี้มาทำการจัดรูปแบบเป็นขั้นตอนสุดท้าย โดยใช้อัลกอริทึม MF สามารถนำมาจัดรูปแบบเพื่อหา Maximum forward reference เก็บไว้ในไฟล์

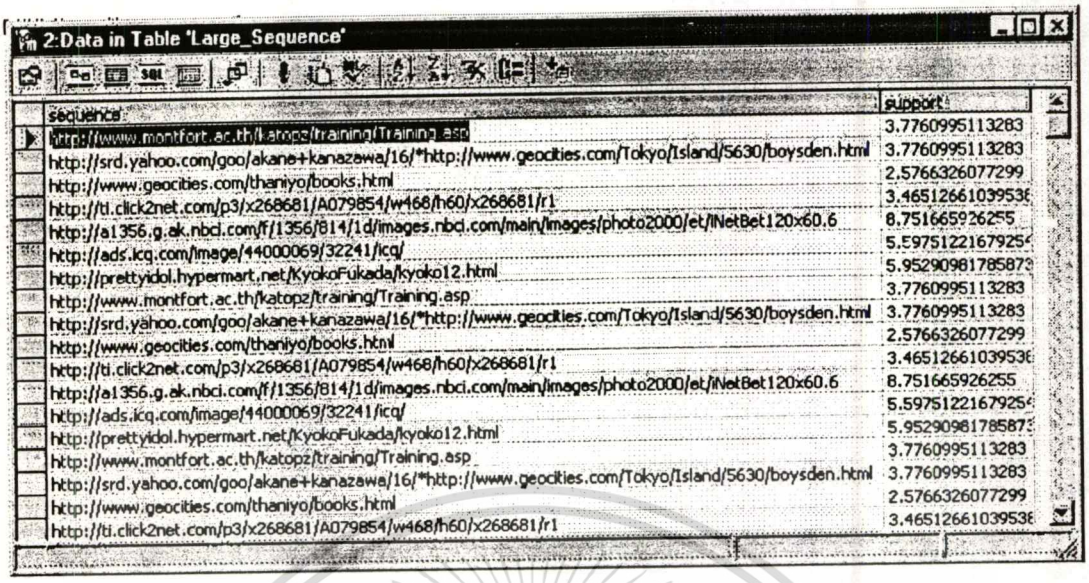
เพื่อใช้เป็นอินพุตของการทำไมนิ่งต่อไป ผลลัพธ์จากตัวอย่างเซสชันไฟล์ 2 บันทัดแรกในรูปที่ 5.5 สามารถหา Maximum forward reference ดังแสดงในรูปที่ 5.6

```
5178,14993,4352,4351,16086,3602
5178,14993,4352,4351,16086,3602,17132
5178,3783,4195
5178,3783,4195,18792,4256,6103
```

รูปที่ 5.6 ตัวอย่าง Maximum forward reference

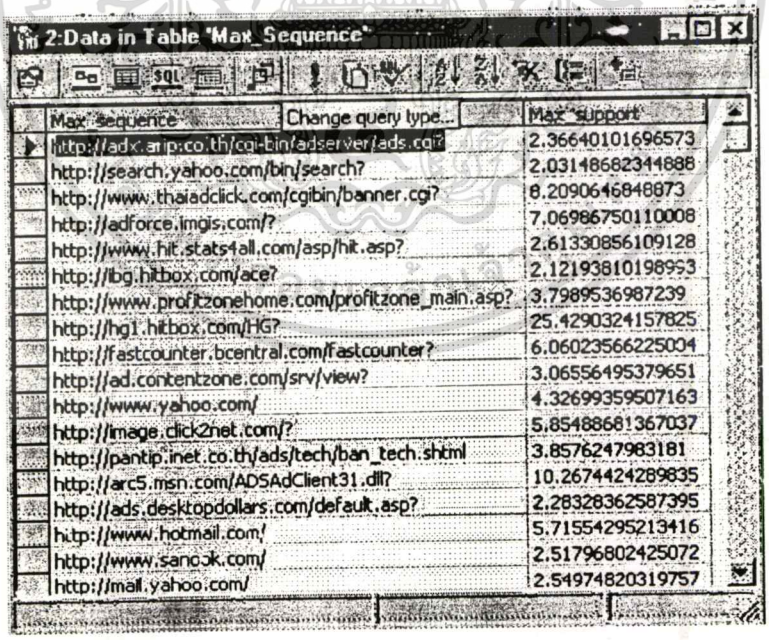
5.1.2 ส่วนการไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บ

หลังจากได้ทรานส์แอ็กชันในส่วนแรกแล้วขั้นตอนต่อไปเป็นการนำเอาทรานส์แอ็กชันนี้มาใช้ในการหาเส้นทางการเดินทางภายในเว็บโดยใช้อัลกอริทึม FS (Full Scan) ดังที่ได้อธิบายไว้ในบทที่ 3 ค่าอินพุตที่รับจากผู้ใช้คือเปอร์เซนต์ค่าสนับสนุน (Minimum support) ซึ่งคือเปอร์เซนต์ของทรานส์แอ็กชันที่บรรจุรูปแบบเส้นทางการเดินทางที่ให้ไว้ ซึ่งถูกใช้เป็นเทรสโฮลด์เพื่อจำกัดจำนวนของรูปแบบเส้นทางการเดินทางที่ค้นพบและรายงานออกมา โดยผลลัพธ์ที่ได้ออกมาคือ Large reference sequence หรือเส้นทางการเดินทางที่มีจำนวนครั้งของปรากฏมากกว่าค่าสนับสนุนที่ตั้งไว้ ผลลัพธ์นี้ถูกนำไปเก็บไว้ที่ตาราง “Large_Sequence” ตัวอย่างของ Large reference sequence ที่ตั้งค่าสนับสนุนเท่ากับ 0.5 % แสดงไว้ดังรูปที่ 5.7



รูปที่ 5.7 ตัวอย่าง Large reference sequence

เมื่อได้ Large reference sequence แล้วขั้นตอนต่อไปคือการนำ Large reference sequence นี้มาใช้หา Maximal reference sequence (“Hot” access pattern) ซึ่งเป็นผลลัพธ์สุดท้ายของการค้นหาเส้นทางการเดินทาง



รูปที่ 5.8 ตัวอย่าง Maximum reference sequence

5.2 การทดสอบโปรแกรม

ก่อนที่จะนำไปใช้งานจริงจะต้องมีการทดสอบการทำงานของโปรแกรมว่าสามารถทำงานได้จริงตามที่ออกแบบไว้หรือไม่ โดยจะเริ่มทำการทดสอบแบบ Bottom-up ซึ่งเป็นการทดสอบจากส่วนย่อยของโปรแกรมก่อน เมื่อทดสอบในส่วนย่อย ๆ แต่ละส่วนเรียบร้อยแล้ว จึงนำทดสอบร่วมกันเพื่อรวมเป็นโปรแกรมทั้งหมด ระหว่างที่ได้ทำการทดสอบได้มีการแก้ไขปรับปรุงข้อผิดพลาดต่าง ๆ ควบคู่กันไปด้วย ซึ่งสามารถอธิบายวิธีการดำเนินการอย่างคร่าว ๆ ได้ดังนี้

1. การทดสอบโปรแกรมในส่วนของการเตรียมข้อมูล

ทดสอบว่าโปรแกรมสามารถเตรียมข้อมูลได้อย่างถูกต้องหรือไม่ ในส่วนแรกทำการทดสอบการทำความสะอาดว่ามีการกรองเฉพาะเว็บเพจที่ผู้ใช้ร้องขอโดยตรงมาได้ถูกต้องหรือไม่ ผลจากการกรองในการทดสอบครั้งแรก ๆ ยังไม่ถูกต้องนักเนื่องจากมีบาง entry ของไฟล์ access log ที่ผู้ใช้ไม่ได้ร้องขอโดยตรง แต่เกิดจากการดาวน์โหลดอย่างอัตโนมัติโดยเทรกที่แทรกใน HTML หลุดเข้ามาเนื่องจากมีการตรวจสอบเงื่อนไขของส่วนต่อท้ายไม่ครบถ้วน จึงได้เพิ่มการตรวจสอบเงื่อนไขของส่วนต่อท้ายเพิ่มขึ้นเพื่อให้ครอบคลุม entry เหล่านี้ให้ได้มากที่สุด โดยผลลัพธ์ของการทำความสะอาดข้อมูลจะมีขนาดเล็กลงอย่างมาก

การระบุผู้ใช้และการระบุเซสชันในการทดสอบได้ใช้เวลาเซสชัน 30 นาทีซึ่งเป็นค่าเวลาดีฟอลท์ที่ผลิตภัณฑ์ที่เกี่ยวข้องกับการระบุเซสชันของข้อมูลจากเว็บที่มีอยู่ในปัจจุบันเลือกใช้อยู่มาใช้ในการทดสอบของเรา ซึ่งผลลัพธ์ที่ได้ของแต่ละเซสชันจะได้ลำดับของการเข้าถึงที่ค่อนข้างยาว โดยมีค่าเฉลี่ยของการเข้าถึงของแต่ละเซสชันประมาณ 25 การเข้าถึง

เมื่อได้เซสชันของแต่ละผู้ใช้แล้วจากนั้นจึงนำผลลัพธ์นี้มาทำการจัดรูปแบบให้อยู่ในรูปแบบ Maximum reference sequence ซึ่งผลลัพธ์ถูกต้อง

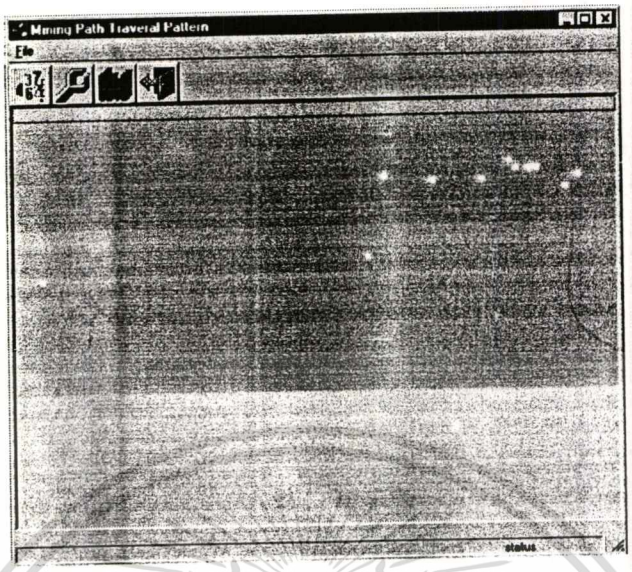
2. การทดสอบโปรแกรมในส่วนการไมนิ่งค้นหาเส้นทางเดินทางภายในเว็บ

ในส่วนนี้เป็นส่วนที่ใช้อัลกอริทึม FS ในการค้นหาเส้นทางการเดินทางภายในเว็บโดยผลที่ได้คือรูปแบบเส้นทางการเดินทางภายในเว็บ (Large reference sequence) เนื่องจากไฟล์ access log ที่นำมาทดสอบเป็นไฟล์ล็อกของพร็อกซีเซิร์ฟเวอร์ดังนั้นจึงเก็บทุก ๆ การร้องขอเว็บไซต์ทุกไซต์ของผู้ใช้ ทำให้ค่าสนับสนุนที่ทำให้ค้นพบเส้นทางการเดินทางมีค่าน้อยมาก

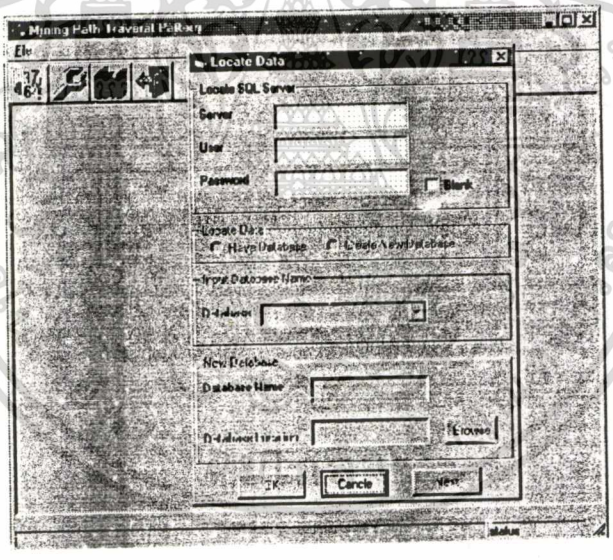
เมื่อได้เส้นทางการเดินทางภายในเว็บแล้วจากนั้นจะนำเอาเส้นทางเหล่านี้มาหา Maximum reference sequence ซึ่งผลลัพธ์ที่ได้ถูกต้อง

3. การทดสอบโปรแกรมโดยรวม

เมื่อเปิดโปรแกรมขึ้นมาจะปรากฏหน้าจอดังรูปที่ 5.9



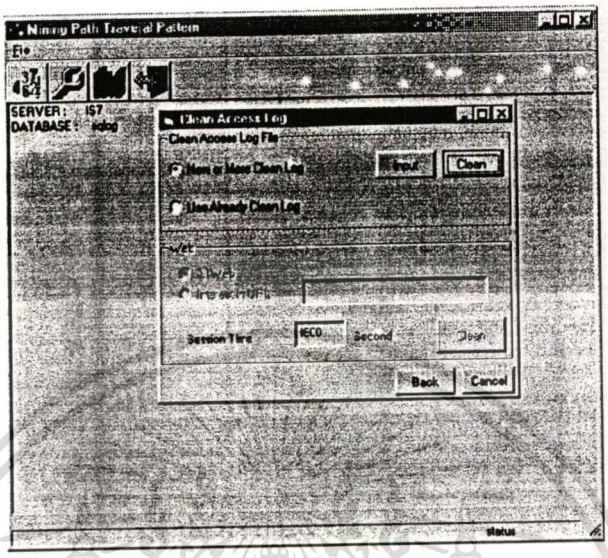
รูปที่ 5.9 หน้าจอหลัก



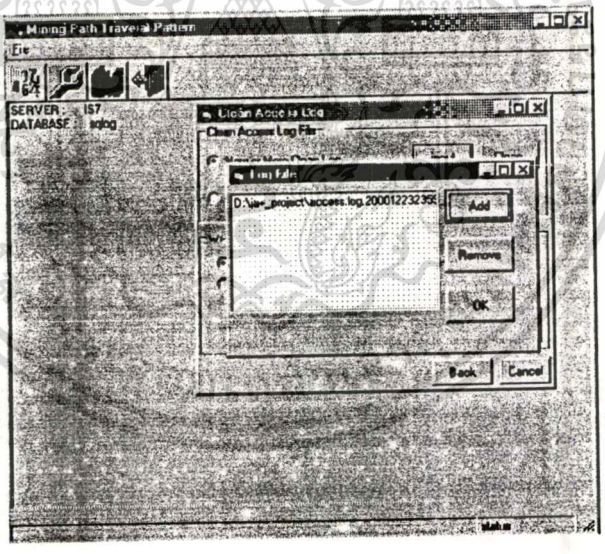
รูปที่ 5.10 หน้าจอรับข้อมูลเข้า

จากนั้นจะทำการนำเข้าข้อมูลเพื่อทำการค้นหาเส้นทาง ซึ่งจะมีหน้าจอดังรูป 5.10 ในส่วนแรกจะทำการติดต่อกับ SQL server ที่ต้องการ จากนั้นจะมีส่วนให้เลือกว่าจะใช้ฐานข้อมูลที่มีอยู่แล้วหรือสร้างขึ้นใหม่ เมื่อมีการกำหนดเรียบร้อยแล้วส่วนต่อไปเป็นหน้าจอทำความสะอาดข้อมูล ดังรูป 5.11 ซึ่งมีส่วนให้เลือกว่าจะทำการทำความสะอาดข้อมูลใหม่/เพิ่ม หรือนำข้อมูลที่ผ่านการทำ

ความสะอาดเรียบร้อยแล้วที่ได้ถูกเก็บไว้ในฐานข้อมูลมาใช้ เมื่อทำการเลือกแล้วส่วนต่อไปเป็นส่วนให้ผู้ใช้กำหนดเว็บไซต์ที่สนใจและเวลาเซสชันที่จะใช้ในการระบุเซสชัน

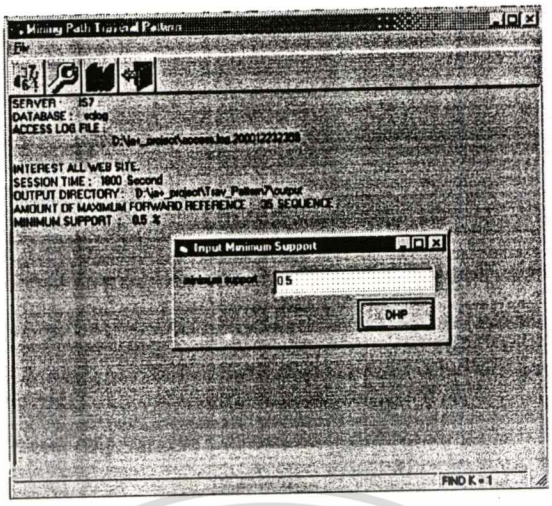


รูปที่ 5.11 หน้าจอทำความสะอาดข้อมูล



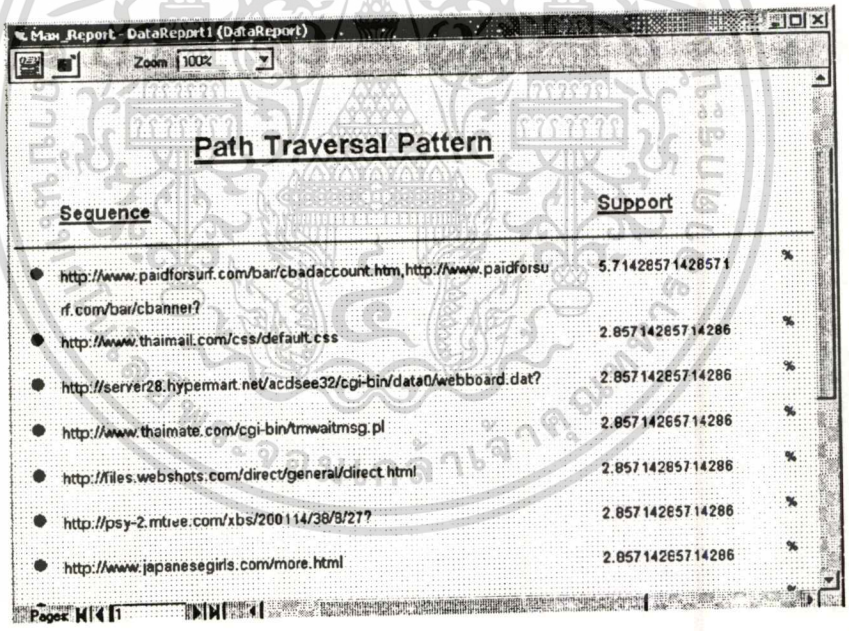
รูปที่ 5.12 หน้าจอรับไฟล์ล็อกใหม่/เพิ่มเติม

เมื่อทำความสะอาดข้อมูลเรียบร้อยแล้วต่อไปเป็นการทำไมนิ่งค้นหาเส้นทางการเดินทางโดยจะมีหน้าจอรับค่าสนับสนุนที่น้อยที่สุดที่สนใจ ดังรูปที่ 5.13



รูปที่ 5.13 หน้าจอรับค่าสนับสนุน

เมื่อทำการหาเส้นทางการเดินทางเสร็จแล้วจะมีหน้าจอแสดงผลลัพธ์ที่หาได้ดังรูปที่ 5.14



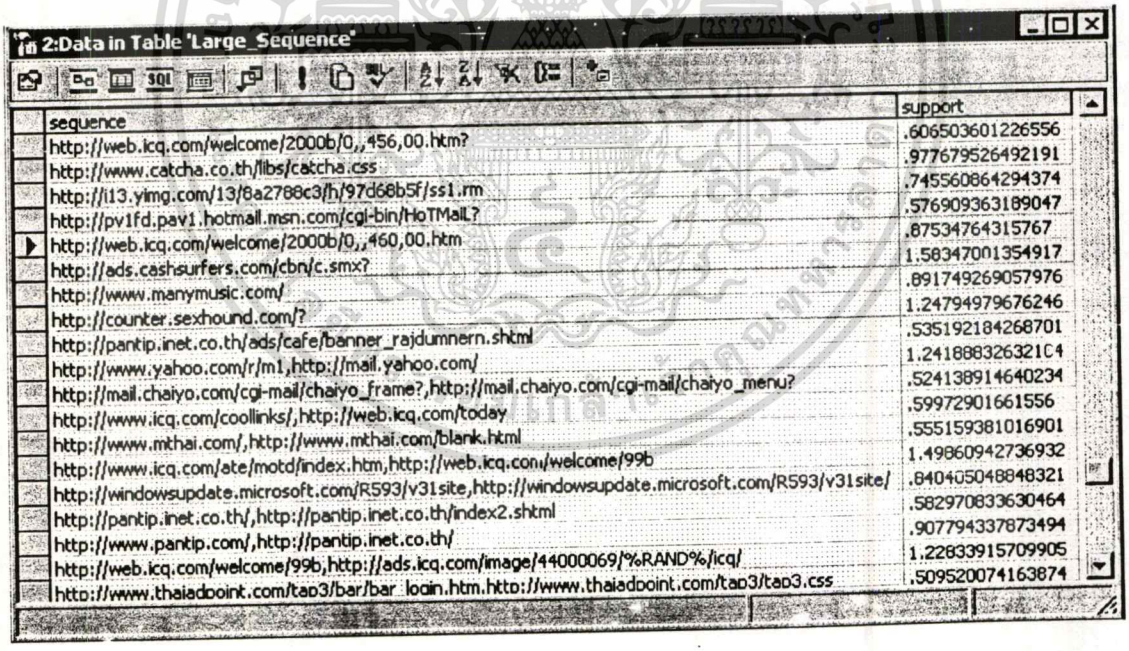
รูปที่ 5.14 หน้าจอแสดงผลลัพธ์

จากการทดลองทำการทดสอบโดยนำไฟล์ access.log ซึ่งเก็บการเข้าถึงในแต่ละวันจำนวน 3 ไฟล์ ซึ่งขนาดของทั้งสองไฟล์รวมกันมีขนาด 1.12 GB หลังจากที่ผ่านการทำความสะอาดแล้ว ได้จำนวนการเข้าถึงของผู้ใช้ (log entry) ทั้งหมด 1,687,173 entry, URL ที่ต่างกัน 318,922 URL

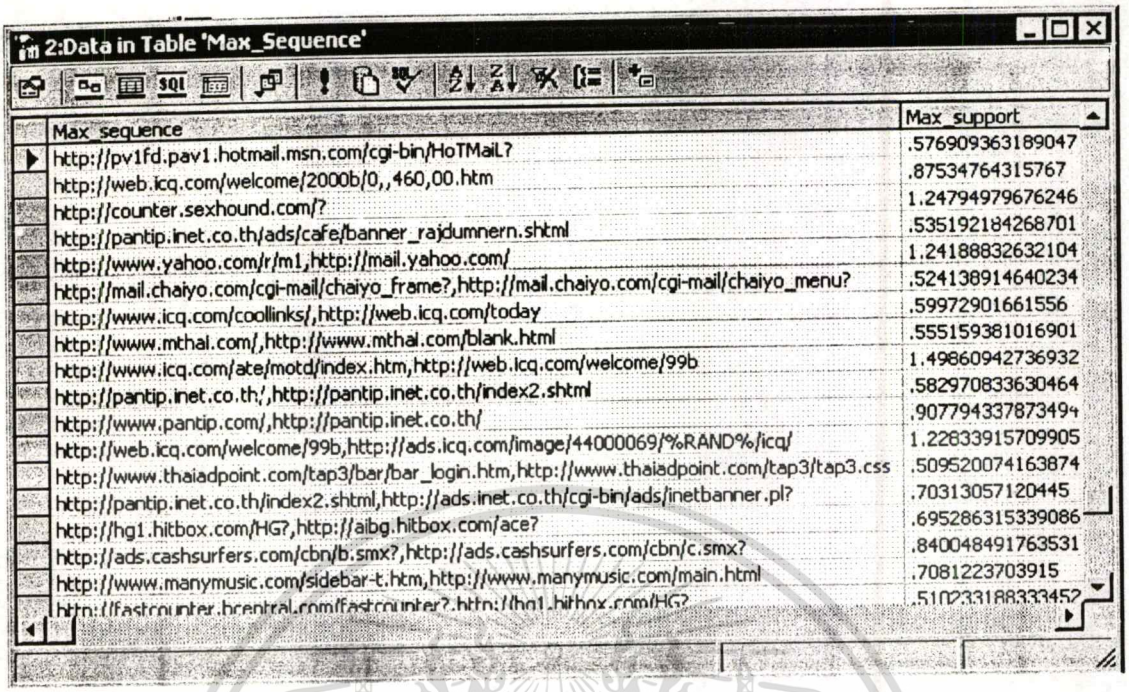
โดยการตั้งเวลาเซสชันเท่ากับ 1800 วินาที และสุดท้ายของขั้นตอนการเตรียมข้อมูลคือจัดให้อยู่ในรูปแบบ maximum forward reference ได้ผลลัพธ์จำนวน 280,460 ทรานส์แอ็กชัน

เมื่อได้ maximum forward reference แล้วขั้นตอนต่อไปทำการไมนิ่งหารูปแบบเส้นทางการเดินทางภายในเว็บโดยการรับค่าสนับสนุนจากผู้ใช้ จากการให้ผู้ใช้กำหนดเงื่อนไขค่าสนับสนุนเท่ากับ 0.5% ซึ่งก็คือต้องมีจำนวนทรานส์แอ็กชันที่บรรจุเส้นทางการเดินทางนี้อย่างน้อย 280,460 × 0.5 % = 1,402 ทรานส์แอ็กชัน ผลลัพธ์ของเส้นทางการเดินทาง (Large reference sequence) ได้เส้นทางการเดินทางจำนวน 226 เส้นทาง และผลลัพธ์ของ Maximum reference sequence จำนวน 203 เส้นทาง โดยจำนวนการเข้าถึงในเส้นทางการเดินทางมากที่สุดเท่ากับ 2 การเข้าถึง ตัวอย่างผลลัพธ์ดังแสดงในรูปที่ 5.15 และ 5.16 ตามลำดับ

ใน Large reference sequence มีเส้นทางที่มีการเข้าถึง 1 การเข้าถึงจำนวน 171 เส้นทาง และการเดินทางที่มีการเข้าถึง 2 การเข้าถึงจำนวน 25 เส้นทาง และใน Maximam reference sequence มีเส้นทางที่มีการเข้าถึง 1 การเข้าถึงจำนวน 169 เส้นทาง และการเดินทางที่มีการเข้าถึง 2 การเข้าถึงจำนวน 25 เส้นทาง



รูปที่ 5.15 ตัวอย่างจากการทดสอบหา Large reference sequence



2:Data in Table 'Max_Sequence'

Max_sequence	Max_support
http://pv1fd.pav1.hotmail.msn.com/cgi-bin/HoTMaiL?	.576909363189047
http://web.icq.com/welcome/2000b/0,,460,00.htm	.87534764315767
http://counter.sexhound.com/?	1.24794979676246
http://pantip.inet.co.th/ads/cafe/banner_rajdumnern.shtml	.535192184268701
http://www.yahoo.com/r/m1,http://mail.yahoo.com/	1.24188832632104
http://mail.chaiyo.com/cgi-mail/chaiyo_frame?,http://mail.chaiyo.com/cgi-mail/chaiyo_menu?	.524138914640234
http://www.icq.com/coollinks/,http://web.icq.com/today	.59972901661556
http://www.mthai.com/,http://www.mthai.com/blank.html	.555159381016901
http://www.icq.com/ate/motd/index.htm,http://web.icq.com/welcome/99b	1.49860942736932
http://pantip.inet.co.th/,http://pantip.inet.co.th/index2.shtml	.582970833630464
http://www.pantip.com/,http://pantip.inet.co.th/	.907794337873494
http://web.icq.com/welcome/99b,http://ads.icq.com/image/44000069/%RAND%/icq/	1.22833915709905
http://www.thaiadpoint.com/tap3/bar/bar_login.htm,http://www.thaiadpoint.com/tap3/tap3.css	.509520074163874
http://pantip.inet.co.th/index2.shtml,http://ads.inet.co.th/cgi-bin/ads/inetbanner.pl?	.70313057120445
http://hg1.hitbox.com/HG?,http://aibg.hitbox.com/ace?	.695286315339086
http://ads.cashsurfers.com/cbn/b.smx?,http://ads.cashsurfers.com/cbn/c.smx?	.840048491763531
http://www.manymusic.com/sidebar-t.htm,http://www.manymusic.com/main.html	.7081223703915
http://fastcounter.hcentral.com/fastcounter?,http://hg1.hitbox.com/HG?	.510233188333452

รูปที่ 5.16 ตัวอย่างจากการทดสอบหา Maximum reference sequence

จากการทดสอบป้อนค่าสนับสนุนที่ต่างกันสามารถสังเกตได้ว่าค่าสนับสนุนของผลลัพธ์ maximum reference sequence มีค่าน้อยมาก เนื่องจากการทดสอบนี้ใช้ข้อมูลจากพร็อกซีเซริฟเวอร์ ซึ่งจัดเก็บการร้องขอเว็บไซต์ของผู้ใช้ทุกการร้องขอที่ออกไปยังเว็บเซิร์ฟเวอร์ของเว็บไซต์หนึ่ง ๆ ทำให้มีการกระจายของข้อมูลจำนวนมาก ดังนั้นรูปแบบที่เกิดขึ้นหนึ่ง ๆ จึงมีความถี่ของการปรากฏน้อยมากเมื่อเทียบกับจำนวนการร้องขอทั้งหมด ทำให้ค่าสนับสนุนของแต่ละรูปแบบเส้นทางการเดินทางมีค่าน้อยลงตามไปด้วย

5.3 การตีความหมายของเส้นทางการเดินทางภายในเว็บ

การวิเคราะห์เส้นทางการเดินทางภายในเว็บสามารถนำมาใช้ในการหาเส้นทางที่มีผู้เข้าเยี่ยมชมที่เกิดขึ้นบ่อยครั้ง ผลลัพธ์ที่ได้สามารถนำมาช่วยในการตัดสินใจทางการตลาด หรือช่วยให้ผู้พัฒนาเว็บไซต์เข้าใจพฤติกรรมของผู้ใช้หรือข้อบกพร่องในการออกแบบเว็บไซต์ได้ ตัวอย่างความรู้ที่ได้จากการวิเคราะห์เส้นทางการเดินทางภายในเว็บเช่น ในขณะนี้เราทำการวิเคราะห์เว็บไซต์ใดเว็บไซต์หนึ่งซึ่งมีผลของเส้นทางการเดินทางดังนี้

- 70 % ของผู้ใช้ที่เข้าถึงเพจ /company/product2 ได้มีการเริ่มต้นเดินทางที่ /company ผ่าน /company/new , /company/products และ /company/product1 ตามลำดับ

- 80 % ของผู้ที่เข้าถึงเว็บไซค์นี้ได้มีการเริ่มต้นที่เพจ /company/product
- 65 % ของผู้ใช้ที่เข้าสู่เว็บไซค์จะออกจากเว็บหลังจากที่มีการเรียกดูเพจน้อยกว่าหรือเท่ากับ 4 ครั้ง

จากตัวอย่างทั้ง 3 เส้นทางการเดินทาง จากกฎข้อแรกสามารถแนะนำได้ว่าข้อมูลที่มีอยู่ในเพจ /company/product2 มีประโยชน์และมีความน่าสนใจต่อผู้ใช้ แต่จะเห็นว่าผู้ใช้ต้องเดินทางผ่านหลายเพจกว่าจะถึงเพจที่ต้องการซึ่งไม่ถูกต้อง ในกฎข้อที่สองสามารถบอกได้ว่าผู้ใช้ได้เข้าถึงเว็บไซค์นี้ผ่านเพจที่ไม่ใช่เพจหลัก (ตัวอย่างเช่นเพจ /company) ซึ่งจากกฎข้อนี้ผู้พัฒนาเว็บอาจจะนำมาช่วยในการออกแบบเว็บไซค์โดยทำการรวมเอาข้อมูลที่ต้องการสื่อให้ผู้ใช้ทราบเข้ามาใส่ไว้ในเพจนี้ และกฎข้อสุดท้ายบ่งชี้การออกจากเว็บไซค์ของผู้ใช้ เนื่องจากผู้ใช้ส่วนมากจะไม่มีการบราวส์เกิน 4 ครั้งในเว็บของเราเพราะฉะนั้นผู้ออกแบบเว็บควรจะต้องแน่ใจว่าข้อมูลสำคัญที่ต้องการจะสื่อให้ผู้ใช้ทราบได้ถูกบรรจุอยู่ใน 4 เพจที่เป็นเพจที่ผู้ใช้เข้าถึงบ่อยครั้ง

จากผลลัพธ์เส้นทางการเดินทางที่โปรแกรมได้สร้างออกมานั้นเป็นเพียงแค่เส้นทางการเดินทางที่ได้ขึ้นบ่อยเท่านั้น ส่วนการตีความหมายและการนำไปใช้งานนั้นผู้ใช้จะต้องเป็นผู้ตีความและตัดสินใจเองเนื่องจากผู้ใช้เป็นผู้ที่รู้จักลักษณะและสภาวะแวดล้อมขององค์กรที่จะสามารถนำความรู้ที่ได้จากการไมนิ่งนี้ไปใช้งานได้

บทที่ 6
บทสรุป

6.1 สรุปผลการศึกษา

โปรแกรมสำหรับไมนิ่งรูปแบบเส้นทางการเดินทางภายในเว็บที่พัฒนาขึ้นมานั้น เป็นโปรแกรมที่พยายามนำเอาข้อมูลล็อกที่ทางองค์กรได้จัดเก็บไว้นำมาใช้ประโยชน์โดยการค้นหาความรู้ที่ได้จากล็อกของการเข้าถึงเหล่านี้

ในการศึกษานี้ได้ทำการสร้างโปรแกรมสำหรับไมนิ่งสำหรับการหารูปแบบเส้นทางการเดินทางภายในเว็บ ซึ่งตัวโปรแกรมสามารถแบ่งขั้นตอนออกเป็น 2 ขั้นตอนหลัก ๆ คือการเตรียมข้อมูล, การหา maximal forward reference, การหา large reference sequence และการหา maximal reference sequence โดยในขั้นตอนแรกทำการเตรียมข้อมูลโดยการแปลงข้อมูลและเก็บเฉพาะข้อมูลที่ต้องใช้จากไฟล์ล็อกเก็บไว้ในฐานข้อมูล จากนั้นจะทำการแปลงข้อมูลล็อคที่ได้มาจากการเตรียมข้อมูลให้อยู่ในรูปแบบของ maximal forward reference โดยมีการกรองผลกระทบของการเข้าถึงย้อนกลับออกไป และสนใจแต่ลำดับการเข้าถึงแบบไปข้างหน้าของผู้ใช้ ในขั้นนี้มีการใช้อัลกอริทึม MF ในการหา maximal forward reference ซึ่งผลลัพธ์ที่ได้จะถูกนำไปเก็บไว้ในฐานข้อมูล จากนั้นในขั้นที่สองจะทำการหา large reference sequence โดยการแสกนผ่านฐานข้อมูล ซึ่งบรรจุ maximal forward reference ไว้ โดยอัลกอริทึมสำหรับการหา large reference sequenceในการศึกษานี้ได้ใช้อัลกอริทึม FS ซึ่งใช้หลักการของอัลกอริทึม DHP โดยใช้เทคนิค hashing และ pruning เข้ามาช่วยทำมีประสิทธิภาพมากขึ้นในการทำงาน

ผลจากการพัฒนาโปรแกรมทำให้ได้เครื่องมือสำหรับการค้นหารูปแบบเส้นทางการเดินทางภายในเว็บของผู้ใช้ ซึ่งผลลัพธ์ที่ได้ออกมานั้นผู้ใช้จะต้องนำมาตีความหมายด้วยตนเองว่าควรจะนำไปประยุกต์ใช้ให้เกิดประโยชน์แก่องค์กรได้อย่างไรบ้าง

6.2 ข้อเสนอแนะ

โปรแกรมนี้ยังมีข้อจำกัดอยู่หลายประการที่ควรจะต้องปรับปรุงแก้ไข เพื่อให้โปรแกรมมีความยืดหยุ่นเหมาะสมกับองค์กรที่จะนำไปใช้ประโยชน์ สิ่งที่ควรจะแก้ไขสำหรับผู้ที่จะพัฒนาต่อไปมีรายละเอียดดังนี้

- เพจที่ผู้ใช้ร้องขอในความเป็นจริงแล้วในไฟล์ล็อกของเว็บเซิร์ฟเวอร์จะไม่ถูกเก็บไว้ทุกการร้องขอ เนื่องจากการมีอยู่ของแคชและพร็อกซีเซิร์ฟเวอร์ทำให้ผู้ใช้ไม่ต้องมีการร้องขอเพจโดยตรงจากเว็บเซิร์ฟเวอร์ ทำให้การค้นหาพฤติกรรมของผู้ใช้ผิดพลาดได้ เพราะฉะนั้นควรต้องหาทางแก้ไขข้อเท็จจริงนี้ให้ได้เพื่อที่จะได้พฤติกรรมที่ถูกต้องจริง ๆ โดยแนวทางแก้มีได้หลายแนวทาง เช่น เขียนสคิปต์บังคับให้ผู้ใช้ต้องคาวน์โหลดเพจใหม่ทุกครั้งที่มีการเรียกดูเพจ การทำ cache busting เป็นต้น

- การระบุผู้ใช้ในโปรแกรมนี้มีการแยกแยะผู้ใช้โดยพิจารณาจาก IP Address เท่านั้น ซึ่งเป็นการพิจารณาที่ไม่เพียงพอเนื่องจากว่าอาจจะมีผลกระทบจากการที่ผู้ใช้เรียกร้องขอเพจผ่านพร็อกซีเซิร์ฟเวอร์ทำให้ข้อมูลที่ถูกบันทึกที่ไฟล์ล็อกปรากฏเป็น IP Address เดียวกันถึงแม้ว่าจะเป็นผู้ใช้คนละคนกันก็ตามซึ่งไม่ถูกต้อง ดังนั้นจึงต้องหาวิธีระบุผู้ใช้ที่แน่นอนให้ได้

- ขั้นตอนการไมนิ่งค้นหาเส้นทางการเดินทางภายในเว็บใช้เวลาค่อนข้างมากสำหรับข้อมูลปริมาณมาก

- การนำเสนอเส้นทางการเดินทางยังไม่มีความหลากหลาย ควรจะนำเสนอให้มีหลายรูปแบบเพื่อให้ผู้ใช้ทำความเข้าใจได้ง่ายขึ้น

บรรณานุกรม

Agrawal,R. and Srikant,R. 1994. "Fast Algorithms for Mining Association Rules." 487-499. In *Proc. of the 20th VLDB Conference*, Santiago, Chile.

Chen, M.S. et.al. 1996. "Data Mining for Path Traversal Patterns in a Web Environment." 385-392. In *Proceedings of the 16th International Conference on Distributed Computing Systems*.

Cooley,R. et. al. 1997. "Grouping Web Page References into Transactions for Mining WWW Browsing Patterns." Technical Report TR 97-021, University of Minnesota, Dept. of Computer Science, Minneapolis.

Cooley,R. et.al. 1997. " Web Mining: Information and Pattern Discovery on the World Wide Web." 558-567. In *International Conference on Tools with Artificial Intelligence*, Newport Beach, CA.

Cooley,R. et.al 1999. "Data Preparation for Mining World Wide Web Browsing Patterns."

Park, J.S. et.al. 1997. "Using a Hash-Based Method with Transaction Trimming for Mining Association Rules" 813-825. *IEEE Transactions on Knowledge and Data Engineering* VOL. 9 NO. 5.

ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาวอุบลพรรณ อุบลนุช
สถานที่เกิด	เชียงใหม่
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยเชียงใหม่
ปีการศึกษาที่สำเร็จการศึกษา	ปีการศึกษา 2541