โครงการวิจัยประจำปีงบประมาณเงินรายได้ 2547

เรื่อง
การศึกษาการทำเว็บไมน์นิ่งโดยใช้เทคนิคการค้นหากฎความสัมพันธ์
Association Rule Discovery for Web Mining

หัวหน้าโครงการวิจัย

รศ. ดร.วรพจน์ กรีสุระเดช

Assoc. Professor Dr. Worapoj Kreesuradej

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

โครงการวิจัยประจำปีงบประมาณเงินรายได้ 2547

เรื่อง

การศึกษาการทำเว็บไมน์นิ่งโดยใช้เทคนิคการค้นหากฎความสัมพันธ์

Association Rule Discovery for Web Mining

หัวหน้าโครงการวิจัย

รศ. ดร.วรพจน์ กรีสุระเดช

Assoc. Professor Dr. Worapoj Kreesuradej

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

สารบัญ

บทคัดย่อ

เว็บล็อกทรานเซคชั่นประกอบไปด้วยข้อมูลที่บันทึกการเรียกใช้งานของผู้เรียกใช้ งานวิจัยนี้จะนำเสนออัลกอริธึมซึ่งมีหน้าที่หลักในการค้นหากฎเพื่อใช้ในการทำนาย ข้อมูลที่เป็นลักษณะเว็บเพจนั้นแตกต่างจากข้อมูลทั่วไปตรงที่ลำดับการเรียกใช้เว็บเพจ (user's next requests) นั้นจะมีความสำคัญมาก ดังจะเห็นว่าเมื่อนำวิธีการ Sequential Pattern กับข้อมูลที่เป็นเว็บเพจนั้นจะมีการกระโดดของข้อมูลเกิดขึ้น เมื่อนำผลที่ได้นั้นเมื่อนำไปใช้งานในการทำนายจะทำให้ผลลัพธ์ที่มีประสิทธิภาพต่ำ

จากการศึกษาทำให้ทราบถึงปัญหาว่าการค้นพบกฎและการลดกฎความสัมพันธ์ดังกล่าว ขนาดของการสับกฎ (slide window) จะถูกกำหนดไว้ตายตัว ซึ่งจะส่งผลโดยตรงต่อการค้นหากฎ ดังเช่น เมื่อมีการกำหนดการสับกฎที่มีขนาดเล็กเกินไป จะทำให้กฎที่มีขนาดยาวไม่สามารถค้นพบได้ ในทางกลับกันหากกำหนดการสับกฎมากเกินไป ก็จะเกิดการสูญเสียกฎที่มีขนาดเล็กเพราะไม่สามารถค้นพบได้เช่นกัน โดยการแก้ปัญหาจะให้วิธีการแบบ Adaptive Window นั้นหมายถึงว่าจะมีการสับกฎได้ทั้งกฎที่มีขนาดเล็กและใหญ่ ซึ่งผลลัพธ์ที่ได้จะทำให้สามารถค้นพบทั้งกฎที่มีรูปแบบสั้นและยาวนั้นเอง ทำให้ความแม่นยำในการทำนายสูงขึ้นด้วย

หลักการในการพิจารณากฎจะให้ Path-based Model โดยการสร้างกฎที่มีความยาวของเว็บเพจ มากขึ้นจะใช้วิธีการ Join Operation เพื่อสร้าง A New Candidate Generations (Ck) โดยวิธีการนี้จะทำให้เวลาในการประมวลผลลดน้อยลงไปด้วย เพราะเหตุว่ากฎหรือรูปแบบการเรียกใช้ บางรูปแบบที่มีความน่าจะเป็นน้อยมากก็จะถูกลดทอนออกไปนั้นเอง

ในการทดลองได้มีการกำหนดกฎหลักขึ้นมา 5 กฎ จากนั้นได้มีการสร้างข้อมูลประกอบเพิ่มเติมดังตารางเว็บล็อก แล้วนำข้อมูลผ่านขั้นตอนของอัลกอริธึมที่ได้ออกแบบไว้ หลังจากนั้นจะได้ข้อมูลที่เป็นลักษณะต้นไม้ (Tree) โดยผลลัพธ์ที่ได้จะถูกลดทอนบางกฎด้วยค่า Confidence และ Page Prediction โดยขั้นตอนสุดท้ายจะมีการพรุนนิ่งบางโหนดทิ้งเพื่อให้เหลือเฉพาะกฎที่มีความแม่นยำสูงอีกทั้งผลลัพธ์ที่ได้ก็จะปรากฏกฎที่เราได้กำหนดไว้ด้วยเช่นกัน

Abstract

Web Log Transaction composed of data that records web page usage of users. This research will purpose algorithm that has major responsibility in searching rules for prediction. Data in form of webpage is different from any other data in sequence of webpage usage as user's next requests that is very important. Sequential Pattern applies to data of webpage will occur the skip in data error and this will create lower effective result.

Referring to the problem, we could learn that relative rule incurred and decreased affects from slide window that is fixed. This result in rules searching for example when slide window is too small it will result in long rule cannot be found. Vice versa, in case of too much slide window, the lost of small rule will be incurred because it could not be found neither. The solution is Adaptive Window that is switching both small and large rules that result in the foundation of both short and long rules. Also the correction of prediction increases.

The principle in rules consideration will create Path-based Model by long rule establishment more and apply the Join Operation method in order to create A New Candidate Generation (Ck). By this method, the time consumption in processing will be reduced because some rules or usage patterns that are lower possibility would be tare off.

In part of experiment of this research, 5-principle rules then the additional data used in web log table will be established. After that the data is processed by algorithm would be designed then the data as Tree result in decreased some rules by Confidence and Page Prediction. The last process is pruning some nodes in order to leave only rule that is very concise and result in what we specify.

1. ความเป็นมา ความสำคัญของปัญหาและจุดประสงค์ของงานวิจัย

แหล่งข้อมูลที่สำคัญในงานวิจัยนี้คือ เว็บล็อกไฟล์ ที่ได้จากจากเว็บเซอร์เวอร์ ซึ่งจะบันทึกพฤติกรรมการเรียกใช้งานของยูสเซอร์ โดยข้อมูลที่เก็บไว้ดังกล่าวนั้นสามารถนำมาใช้ในการทำนายความต้องการเรียกใช้เว็บเพจได้ ดังจะเห็นว่าข้อมูลที่บันทึกไว้จะประกอบด้วยส่วนหลักๆ เช่น เลขไอพี ,ขนาดของไฟล์ที่เรียก, วัน และเวลา ซึ่งสามารถที่จะค้นหารูปแบบการเข้าเรียกใช้งานโดยการวิเคราะห์จากข้อมูลดัง กล่าวข้างต้น โดยจุดประสงค์ในการวิจัยจะมีประโยชน์ใช้ในการทำนายความต้องการของยูสเซอร์ที่จะเรียกใช้งานเว็บเพจในลำดับต่อไป โดยการทำนายนี้จะมีความแม่นยำสูง

ในส่วน Mining sequential patterns [11] นั้นเมื่อนำมาประยุกต์กับข้อมูลที่เป็นเว็บเพจในการทำนายความต้องการของยูสเซอร์ที่จะเรียกใช้งานเว็บเพจในลำดับต่อไป (user's next requests) จะให้ผลที่คลาดเคลื่อนเพราะเหตุว่า การใช้ Sequential pattern จะเกิดการกระโดดของไอเท็ม ซึ่งเมื่อนำมาใช้กับเว็บเพจในการทำนายเว็บเพจในลำดับต่อไปจะให้ผลลัพธ์ที่มีประสิทธิภาพต่ำ

ในส่วนของงานวิจัยในปัจจุบันนี้ Pitknow et al [5] ได้ให้คำแนะนำเกี่ยวกับการทำนายโดยมีพื้นฐานมาจาก K^{th}-order Markov model โดยกล่าวว่ารูปแบบหรือว่าเส้นทางเดิน (path) ที่มีลักษณะยาวจะให้ความน่าเชื่อถือกว่าเส้นทางเดินที่มีลักษณะสั้น อย่างไรก็ตามเส้นทางเดินที่ยาวกว่าเมื่อนำมาวิเคราะห์ในงานวิจัยจะมีเส้นทางเดินที่เป็นลักษณะ noisy path มากตามไปด้วย ซึ่งนั้นหมายถึงว่าจะมีผลต่อความแม่นยำให้น้อยลงไปด้วยเช่นกัน Su et al [7] ได้นำเสนอการสร้างรูปแบบจาก N-gram โดยเลือกใช้ a smoothing algorithm ที่เรียกว่า Cascading model โดยเป็นการเพิ่มความน่าเชื่อถือให้กับเส้นทางเดินที่มีลักษณะยาวไว้

โดยในงานวิจัยเกี่ยวกับการทำนายเว็บเพจหลายๆงาน ซึ่งมีจุดประสงค์ในการเพิ่มประสิทธิภาพของระบบอินเตอร์เน็ต ลดปริมาณการส่งข้อมูล (Traffic) ในระบบ ทำให้การเรียกใช้เว็บเพจนั้นเร็วขึ้น ทำให้ระบบโดยร่วมฉลาดขึ้น ลดการทำงานของเซอร์เวอร์ให้น้อยลง [8]

ปัญหาที่เจอใน[9] นั้น การค้นพบกฎและการลดกฎความสัมพันธ์ดังกล่าว ขนาดของการสับกฎ (slide window) จะถูกกำหนดไว้ตายตัว ซึ่งจะส่งผลโดยตรงต่อการค้นหากฎ ดังตัวอย่างเช่น เมื่อมีการกำหนดการสับกฎที่มีขนาดเล็กเกินไป จะทำให้กฎที่มีขนาดยาวไม่สามารถค้นพบได้ สมมุติว่ากำหนดการสับกฎเท่ากับ 3 เช่น LogT = {A, B, C, D, E, F} เราจะได้กฎ {A, B, C} ➔ D แต่เราจะไม่สามารถค้นเจอกฎเหล่านี้เลย เช่น {A, B, C, D} ➔ E, {A, B, C, D, E} ➔ F ในกรณีที่กำหนดการสับกฎมากเกินไป เช่นกำหนดเท่ากับ 4 เราจะได้กฎ {A, B, C, D} ➔ E แต่จะสูญเสียกฎที่มีขนาดเล็กดังเช่น {A} ➔ B, {B} ➔ C, {A,B} ➔ C

1.1 จุดประสงค์ของงานวิจัย

จากปัญหาดังกล่าวที่ได้ศึกษานั้นในงานวิจัยนี้สามารถจำแนกจุดประสงค์ได้เป็นลำดับดังต่อไปนี้

- ศึกษาข้อมูลที่เป็นลักษณะเว็บเพจ นำมาวิเคราะห์และแก้ปัญหาในการทำนายการเรียกใช้เว็บเพจ
- การแก้ไขปัญหาในการกำหนดการสับกฎที่เป็นลักษณะตายตัว โดยเพิ่มประสิทธิภาพการสับกฎเป็นลักษณะเปลี่ยนแปลงได้ (Adaptive window) ซึ่งจะทำให้ปัญหาที่เกิดขึ้นจากการกำหนดการสับกฎแบบตายตัวหมดไป โดยทำให้การค้นหากฎเจอทั้งกฎที่มีขนาดสั้นและยาว ส่งผลให้ความแม่นยำเพิ่มขึ้นตามไปด้วย
- การลดทอนกฎบางกฎในขณะมีการทำกระบวนการ Pattern Discovery โดยมีการนำค่า Support Factor มาพิจารณาร่วมในการลดทอนกฎที่ไม่จำเป็นออกไป

2. ทฤษฎีและหลักการที่เกี่ยวข้องโดยย่อ

2.1 Mining Sequential Patterns

เราจะกล่าวถึงปัญหาของการไมนิ่ง Sequential pattern เมื่อใช้กับฐานข้อมูลที่เป็นลักษณะเว็บเพจ ก่อนอื่นจะกล่าวถึงตัวอย่างของฐานข้อมูลการเช่าภาพยนตร์ของร้านวีดีโอแห่งหนึ่ง เมื่อลูกค้ามาเช่าภาพยนตร์เรื่อง Harry Potter และต่อด้วย Lord of The Ring และสุดท้ายคือเรื่อง The Last Samurai ซึ่งการเช่าภาพยนตร์ดังกล่าวนั้นมิจำเป็นต้องต้องมีการเช่าตามลำดับ ซึ่งลูกค้าสามารถที่จะเช่าวีดีโออื่นๆ ระหว่างการเช่าวีดีโอทั้งสามดังกล่าวนี้ก็ได้ หากเป็นกรณีดังกล่าวข้างต้นนี้ กล่าวได้ว่า Sequential Pattern นี้ก็เป็นที่ยอมรับได้ (Support) แต่การใช้งานสำหรับเว็บเพจลำดับในการเรียกใช้งานจะมีผลต่อการทำนาย ซึ่งจะได้กล่าวในโอกาสต่อไป

เมื่อกำหนดให้ ฐานข้อมูลพฤติกรรมของลูกค้าโดยแต่ละกิจกรรมจะประกอบด้วย รหัสลูกค้า, เวลาการซื้อและรายชื่อสินค้าที่จัดซื้อ (ซึ่งในเวลาเดียวกันจะไม่มีลูกค้าซื้อสินค้ามากกว่าหนึ่งรายการ) โดยที่ Sequence คือลำดับของรายการ Items โดยที่เราจะแสดงให้ itemset i $(i_1,i_2,\ldots i_m)$ เมื่อ i_j คือ item ใดๆ โดยที่ลำดับของ sequence s $\langle s_1,s_2,s_n\rangle$ โดยที่ s_j คือ itemset ใดๆ

Sequence $\langle a_1\ a_2\ a_n\rangle$ จะประกอบไปด้วย sequence อื่นๆ $\langle b_1\ b_2\ b_m\rangle$ โดยที่ $i_1 < i_2 <\ldots<i_n$ ดังตัวอย่าง $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$ $a_n \subseteq b_{in}$ ดังตัวอย่าง Sequence $\langle$ (7) (3 8) (9) (4 5 6) (8)$\rangle$ ประกอบไปด้วย $\langle$(3) (4 5) (8)$\rangle$ ดังนั้น (3) $\subseteq$ (3 8) , (4 5) $\subseteq$ (4 5 6) และ (8) $\subseteq$ (8) อย่างไรก็ตาม $\langle$(3) (5) $\rangle$ ไม่ได้อยู่ใน $\langle$(3 5)$\rangle$ จะเห็นได้ว่าหาก Sequence ใดที่เป็น Maximal Sequence แล้วนั้น sequence ดังกล่าวจะไม่อยู่ sequence อื่นๆอีกต่อไป เราจะจัดเรียงกิจกรรม (Transaction) ตามเงื่อนไขของเวลาที่เกิด โดย T_1, T_2,.....T_n โดยที่กลุ่มของ item ใน T_i แสดงด้วย itemset (T_i) โดยที่ลำดับของลูกค้าที่ใช้บริการจะเป็นลักษณะดังนี้คือ $\langle$itemset(T_i), itemset(T_2),...itemset (T_n)$\rangle$

ปัญหาใน Mining sequential patterns คือการค้นหา Maximal sequence ซึ่งจะต้องมีค่า minimum support โดยแต่ละ Maximal sequence ใดๆ นั้นแสดง sequential pattern. โดยที่ Sequence ใดๆ ที่มีค่า minimum support ก็จะเป็น large sequence นั้นเอง

ดังตัวอย่างพิจารณาฐานข้อมูลดังรูปที่ 1 โดยที่ฐานข้อมูลดังกล่าวได้มีการจัดเรียงตามรหัส (customer ID)และเวลาการเกิดกิจกรรม (transaction-time) รูปที่ 2 แสดงฐานข้อมูลในรูปของ customer sequence. เมื่อกำหนดให้ minimum support มีค่าเท่ากับ 25% ซึ่งจะได้ Maximal sequence ดังนี้คือ $\langle$(30) (90)$\rangle$ และ $\langle$(30) (40 70)$\rangle$ โดยที่ $\langle$(30) (90)$\rangle$ เกิดขึ้นกับ ID 1,4 โดยที่ ID 4 ได้มีการซื้อ item(40, 70) ระหว่าง item 30 และ item 90. ส่วน Sequence pattern $\langle$30(40 70)$\rangle$ เกิดจาก ID2,4 โดยที่ ID2 ซื้อ 40 ตามด้วย 60 และ 70 ตามลำดับ

Customer Id	Transaction Time	Items Bought
1	June 25 '04	30
1	June 30 '04	90
2	June 20 '04	10, 20
2	June 15 '04	30
2	June 20 '04	40, 60, 70
3	June 25 '04	30, 50, 70
4	June 25 '04	30
4	June 25 '04	40, 70
4	June 25 '04	90
5	June 12 '04	90

รูปที่ 1 ฐานข้อมูลถูกจัดเรียงโดยรหัสและเวลาของกิจกรรมนั้น [11]

Customer Id	Customer Sequence
1	⟨(30) (90)⟩
2	⟨(10 20) (30) (40 60 70)⟩
3	⟨(30 50 70)⟩
4	⟨(30) (40 70) (90)⟩
5	⟨(90)⟩

รูปที่ 2 การจัดเรียง sequence ของฐานข้อมูล [11]

Sequence Patterns with support > 25%
⟨(30) (90)⟩
⟨(30 40 70)⟩

รูปที่ 3 แสดงคำตอบ [11]

L_1

1-Sequences	Support
⟨1⟩	4
⟨2⟩	2
⟨3⟩	4
⟨4⟩	4
⟨5⟩	4

L_2

2-Sequences	Support
⟨1 2⟩	2
⟨1 3⟩	4
⟨1 4⟩	3
⟨1 5⟩	3
⟨2 3⟩	2
⟨2 4⟩	2
⟨3 4⟩	3
⟨3 5⟩	2
⟨4 5⟩	2

L_3

3-Sequences	Support
⟨1 2 3⟩	2
⟨1 2 4⟩	2
⟨1 3 4⟩	3
⟨1 3 5⟩	2
⟨2 3 4⟩	2

L_4

4-Sequences	Support
⟨1 2 3 4⟩	2

รูปที่ 4 แสดง Large Sequences [11]

ตัวอย่างของ Sequence ที่ไม่มีค่า minimum support คือ sequence ⟨(10 20) (30)⟩ ที่เกิดขึ้นเฉพาะใน ID2. ส่วน Sequence ⟨(30)⟩, ⟨(40)⟩, ⟨(90)⟩, ⟨(30) (40)⟩, ⟨(30) (70)⟩ และ ⟨(40) (70)⟩ ถึงแม้ว่ามีค่า minimum support แต่ว่าไม่ได้เป็น Maximal sequence.

2.2 The Sequence Algorithm

โครงสร้างโดยทั่วไปของอัลกอริธึมสำหรับ Sequence Phase โดยแต่ละส่วนจะมีการค้นหา Large sequence โดยที่เราจะนำมันไปสร้าง Candidate sequence และนำ Candidate sequence ไปทำการสร้าง Large sequence ในขั้นตอนต่อๆไป โดยในขั้นตอนแรกนั้นจะเป็นการหา 1-sequence ที่มีค่า minimum support นั้นเอง

2.2.1 Algorithm Apriori.

อัลกอริธึมดังกล่าวในรูปที่ 5 นั้นแต่ละส่วนจะใช้ large sequence จาก ส่วนก่อนหน้าในการสร้าง candidate sequence และหลังจากนั้นจะวัดค่า support อีกทั้งยังมีการค้นหา Maximal sequence

L_1 = {large 1-sequences};//Result of litemset phse

For (k=2; $L_{k-1} \neq \varnothing$; k++} do

Begin

C_k = New candidate generated from L_{k-1}

(see section Appriori candidate generation)

for each customer-sequence c in the database do

Increment the count of all candidates in C_k

That are contained in c

L_k = Candicates in C_k with minimum support.

End

Answer = Maximul Sequences in $U_k L_k$;

รูปที่ 5 Algorithm Apriori

2.2.2 Apriori Candidate Generation

หน้าที่ของ The apriori-generate จะใช้ตัวแปร L_{k-1} โดยขั้นตอนแรกคือการ Join L_{k-1} กับ L_{k-1}

Insert into C_k

Select $p.litemset_1$,...$p.litemset_{k-1}$, $q.litemset_{k-1}$

From L_{k-1} p, L_{k-1} q

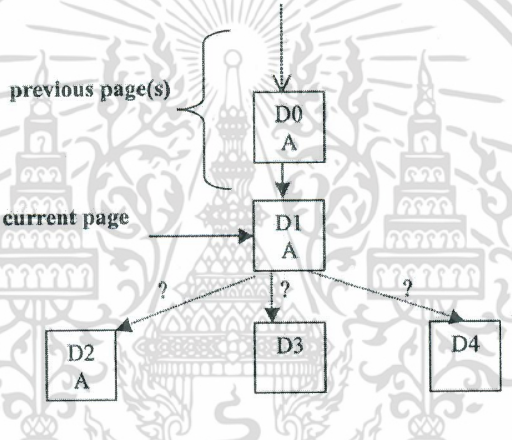
Where $p.litemset_1 = q.litemset_1$,...

$p.litemset_{k-2} = q.litemset_{k-2}$;

2.3 N-gram Model

โดยทั่วไป N-gram model มีด้วยกัน 2 แบบ คือ Point-based model และ path-base model. Point-based prediction model นั้นจะใช้ 1-gram ในการสร้าง prediction model โดยที่การทำนาย ความต้องการเรียกหน้าเพจต่อไปจะคำนึงจาก หน้าเพจปัจจุบันที่เรียกใช้งานอยู่ดังตัวอย่างในรูปที่ 7 เพจ D1, เพจ D2, เพจ D3 และ เพจ D4 ซึ่งจะมีโครงสร้างดังรูป ตัวอย่างเช่นเว็บเพจที่ผู้เรียกเข้าใช้งานอยู่ปัจจุบัน คือ เพจ A ดังนั้นการทำนายว่าต่อไปจะมีการเรียกใช้เพจไหนนั้น จะนำเพจ A ซึ่งเป็นเพจปัจจุบันมาเป็นข้อมูลประกอบเพื่อการพิจารณาตัดสินใจเท่านั้น

ส่วน Path-based prediction นั้นจะให้การพิจารณาทั้งหน้าเพจที่เปิดใช้อยู่ปัจจุบันและเส้นทางเดินของเว็บเพจที่ผ่านมาแล้วด้วย ซึ่งโดยส่วนมากแล้วจะประกอบด้วยเว็บเพจมากกว่าหนึ่งเพจ ดังตัวอย่าง 7 เพจ D0, เพจ D1, เพจ D2, เพจ D3 และ เพจ D4 สมมุติว่า เพจปัจจุบันที่เรียกใช้อยู่คือ D1 และก่อนหน้านั้นได้มีการเรียกใช้เพจอื่นๆบ้างแล้ว ดังนั้น Path-base Model จะใช้เส้นทางเดินจากในอดีตจนหนึ่งปัจจุบันคือ D1 เพื่อใช้ในการทำนายความต้องการของหน้าเพจต่อไป



รูปที่ 7 Path-based Prediction Model

ดังจะเห็นได้ว่า Point-based prediction model นั้นให้ผลการทำนายที่มีความแม่นยำน้อย เพราะเหตุว่าไม่ได้นำเพจที่เคยเยี่ยมชมมาประกอบการพิจารณาด้วย กล่าวได้ว่าข้อมูลเหล่านี้จะมีความสำคัญพอสมควรในการใช้ประกอบการวิเคราะห์การทำนาย ดังนั้นจะเห็นได้ว่า Path-based model จะได้รับความนิยมใช้งานมากกว่า สิ่งสำคัญในการสร้าง N-gram Model นั้นควรจะพิจารณาถึงปริมาณ ขนาดของ N-gram ที่จะให้ความแม่นยำสูงนั้นควรจะเป็นเท่าไหร่

Pitkow et al [5] ได้ให้คำแนะนำเกี่ยวกับการทำนายโดยพื้นฐานของ Kth-order Markov models. หากว่าใช้ เส้นทางเดิน (Path) ที่มีขนาดยาวจะให้ความแม่นยำกว่าเส้นทางเดินแบบสั้น แต่ผลจากการใช้เส้นทางเดินแบบยาวนั้น เมื่อเกิดการประมวลผลของอัลกอริธึมแล้วอาจจะได้เส้นทางส่วนเกินเพิ่มขึ้นมากตามไปด้วย ซึ่งนั้นหมายถึงในบางครั้งความแม่นยำอาจจะลดตามไปด้วย

Su et al [7] ได้ทดสอบและศึกษาความแม่นยำของ N-gram Model ที่มีความแตกต่างกันและผลสรุป กล่าวว่า N-gram Model ที่มีขนาดยาวจะให้ความแม่นยำกว่า N-gram model ที่มีขนาดสั้น

2.4 Prediction Model Construction

กล่าวถึงการสร้าง Prediction Model ในหลายๆแบบด้วยกัน ซึ่ง Rule-Representation Methods มันเป็นเครื่องมือสำคัญในงานวิจัย โดยจะกล่าวได้ดังต่อไปนี้ การแยกเว็บเพจจากล็อกไฟล์ โดยใช้ Association rule ในการแยกแยะข้อมูลต่างๆ จากล็อกไฟล์

Web Logs และ User Sessions ในงานวิจัยนี้จะใช้ข้อมูลจาก Web server logs ซึ่งสิ่งสำคัญในลำดับต้นๆ คือการทำความเข้าใจเกี่ยวกับข้อมูลต่างๆ ที่ได้จากล็อกไฟล์ ซึ่งข้อมูลเหล่านี้จะนำมาใช้ประกอบในการสร้าง Prediction Model ต่างๆ ดังรูปที่ 8 นั้นจะแสดงล็อกไฟล์จากเว็บเซฟเวอร์ โดยปกติแล้วล็อกไฟล์นั้นประกอบด้วยเรคคอร์ดหลายๆ เรคคอร์ด ซึ่งจะมากจะน้อยขึ้นอยู่กับปริมาณการเยี่ยมชมของผู้เรียกใช้งานหรือยูสเซอร์ ซึ่งการเยี่ยมชมเว็บไซต์ต่างๆ ของของยูสเซอร์นั้น จะถูกบันทึกเก็บไว้ในไฟล์ดังกล่าวนี้ โดยแต่ละยูสเซอร์ ที่ถูกบันทึกจะแยกเป็นแต่ละเรคคอร์ด โดยจะประกอบไปด้วยส่วนสำคัญๆ ดังนี้ คือ

- User's host name หรือ IP address
- Time stamp ซึ่งจะบันทึกเวลาในการเรียกใช้งานของยูสเซอร์
- HTTP method (GET, Post,etc) วิธีการเรียกใช้งานเว็บเพจของยูสเซอร์
- URL ของ เว็บเพจที่ถูกเรียกใช้
- Status code ของการตอบสนองจาก HTTP
- Number of byte จำนวนหรือขนาดของไฟล์ที่ถูกเรียกใช้งาน ขนาดของไฟล์ดังกล่าวนี้จะเป็นตัวบอกว่าไฟล์ที่ยูสเซอร์เรียกใช้งานนั้นมีขนาดเท่าไหร่

```
in24.inetnebr.com - - [01/Aug/1995:00:00:01 -0400] "GET /shuttle/missions/sts-68/news/sts-68-mcc-
05.txt HTTP/1.0" 200 1839
uplherc.upl.com - - [01/Aug/1995:00:00:07 -0400] "GET / HTTP/1.0" 304 0
uplherc.upl.com - - [01/Aug/1995:00:00:08 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 304
```

รูปที่ 8 ข้อมูลที่ได้จาก Log File

เมื่อเราได้ล็อกไฟล์มาแล้วประการต่อมาคือการทำการตัด (Clean Processing) บางส่วนที่ไม่จำเป็นออกไป ซึ่งสิ่งต่างๆเหล่านี้ดังเช่น ภาพต่างๆ, ไฟล์วีดีโอคลิป อื่นๆ ซึ่งส่วนประกอบที่ไม่จำเป็นเหล่านี้ ควรจะตัดออกไปเพื่อที่จะให้ข้อมูลที่เหลือนั้นเป็นข้อมูลที่จำเป็นสำหรับการสร้าง Prediction Model เท่านั้น

ขั้นตอนต่อไปคือการแยกเหตุการณ์ของแต่ละยูสเซอร์ (User Session) จากเว็บล็อกไฟล์ ที่ผ่านการกลั่นกรองเรียบร้อยแล้ว ดังตัวอย่าง สมมุติว่าเว็บล็อกไฟล์ประกอบด้วย การเรียกใช้งานเว็บเพจต่างๆดังต่อไปนี้

Time	User ID	Requested Document
00:00:01	U1	A
00:00:02	U2	B
00:00:03	U2	C
00:00:04	U3	D
00:00:05	U1	E
.....	.....	.....

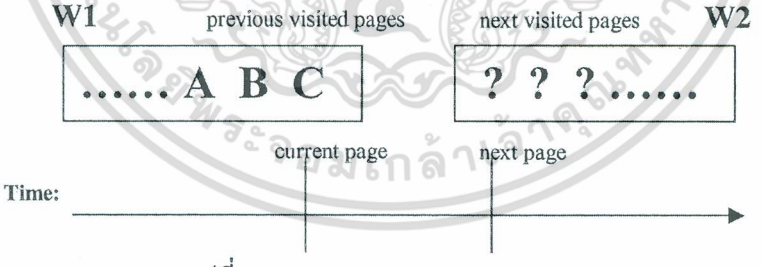
User ID	Session Sequence
U1	A, E, ...
U2	B, C, ...
U3	D, ...
.....	.....

รูปที่ 9 แสดงการเรียกใช้งานเว็บเพจของยูสเซอร์แต่ละท่าน

2.4.1 Moving Window Pairs และ Log Table

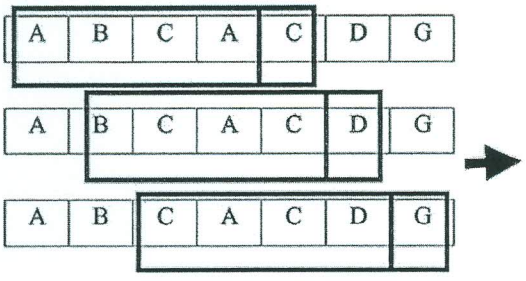
ในการค้นหารูปแบบการเรียกใช้งานของยูสเซอร์ (Access Pattern) ที่เป็น Association rules นั้น เราจะได้มาจากการทำประมวลผลจากข้อมูลเว็บล็อกไฟล์ โดยพิจารณาลำดับและเวลา เราได้ให้คำนิยามคำว่า "Moving window pair" โดยที่จะประกอบด้วยสอง Adjacent window โดยที่หน้าต่างแรกนั้นเราเรียกว่า Antecedent window ซึ่งจะครอบคลุมเว็บเพจที่เคยเยี่ยมชมมาแล้ว โดยเรียงจากอดีตจนถึงปัจจุบัน โดยการเปรียบเทียบเวลาเป็นเงื่อนไขหลัก และในหน้าต่างที่สองเราเรียกว่า Consequent window ซึ่งจะครอบคลุมเว็บเพจที่เยี่ยมชมในอนาคต

ในการเรียกใช้งานหน้าต่างทั้งสองแบบนั้นในโอกาสต่อไปจะเรียก Antecedent window = W1 และ Consequent window = W2 ดังจะเห็นได้ว่าเว็บเพจในส่วน W1 นั้นจะประกอบไปด้วยเว็บเพจของ W2 ด้วย ดังนั้นจากกรณีดังกล่าวข้างต้นจะเห็นได้ว่า เว็บเพจใน W1 จะมีผลในการเรียกใช้งานเว็บเพจในการทำนายด้วยเช่นกัน



รูปที่ 10 แสดง Moving window pair

จากการประยุกต์ใช้งาน Moving pair window กับข้อมูลที่ได้ผ่านการกรั่นกรองแล้ว สิ่งที่เราจะได้รับคือตารางความสัมพันธ์ที่เป็นกฎความสัมพันธ์ ซึ่งจะถูกนำมาทำการไมนิ่ง โดยจำนวนของคอลัมพ์นั้นจะเป็นตัวบอกขนาดของ Moving pair window เราจะเรียกตารางดังกล่าวนี้ว่า Log Table โดยที่มันจะแสดงเหตุการณ์ทั้งหมดของเว็บล็อกนั้นเอง ดังรูป 10 แสดงตัวอย่างของเหตุการณ์ (A, B, C, A, C, D, G) เมื่อเรากำหนดขนาดของ W1 = 4 และ W2 = 1 ดังรูป 11 เมื่อมีการเลื่อน Moving Pair window แต่ละครั้งเราจะได้รูปแบบการเรียกใช้งานดังรูปด้านล่าง



W1				W2
D1	D2	D3	D4	P1
A	B	C	A	C
B	C	A	C	D
C	A	C	D	G

รูปที่ 11 แสดงการเลื่อนของ Moving Pair Window

ขนาดของ W2 นั้นขึ้นอยู่กับว่าความสามารถของการทำนายว่าจะเป็นเท่าไร ดังตัวอย่าง หากให้ W2 = 2 นั้นหมายถึงว่าสามารถที่จะทำนายได้สองเพจ ในงานวิจัยนี้จะใช้ W2 = 1 เพื่อให้ง่ายในการปรับเปลี่ยนค่าพารามิเตอร์ต่างๆนั้นเอง

2.5 Rule Representation methods

จะกล่าวถึงการแตกกฎตามรูปแบบของ LHS➔RHS จากตารางล็อกตามที่ได้กล่าวไว้ข้างต้น กล่าวได้ว่า RHS คือเพจที่จะถูกเรียกใช้งานต่อไปตามกฎความสัมพันธ์ ในส่วนวิธีการแตก LHS เราจะนำเสนอ 5 วิธีการด้วยกัน

- วิธีการแรกเรียกว่า the subset rules

ซึ่งกฎเหล่านี้นั้นจะเหมือนกับกฎความสัมพันธ์แบบเดิมซึ่งจะเป็นวิธีที่ไม่ได้นำลำดับและตัวที่อยู่ติดกันมาใช้ประกอบการวิเคราะห์ ดังนั้นเมื่อเรานำวิธีการกฎความสัมพันธ์ดังเช่น Aprioi method [10] มาใช้ในการประมวลผลกับตารางล็อกไฟล์ เราจะได้ subset rules ดังตารางที่ 1 ต่อไปนี้

W1	W2	Extracted Rules
A, B, C	D	{A, B, C} → D, {A, B} → D, {B, C} → D, {A, C} → D, {A} → D, {B} → D, {C} → D

ตารางที่ 1 แสดงตัวอย่างการใช้ วิธี The Subset Rule representation

- วิธีทีสอง เรียกว่า The sub sequence rules.

ซึ่งจะเป็นการนำลำดับของข้อมูล (Order) เข้ามาพิจารณาด้วย โดยจะเรียงตามลำดับการเกิดของเว็บเพจ ดังตารางที่ 2 ตามตัวอย่างเมื่อใช้วิธีการ sub sequence rules จะได้ A, B เมื่อ A เป็นเว็บเพจที่เกิดก่อน B กล่าวได้ว่าวิธีการนี้จะคล้ายคลึงกับอัลกอริธึมใน Sequential mining [4]

W1	W2	Extracted Rules
A, B, C	D	(A, B, C) → D, (A, B) → D, (B, C) → D, (A, C) → D, (A) → D, (B) → D, (C) → D

ตารางที่ 2 แสดงตัวอย่างการใช้วิธี The sub sequence Rule representation

- วิธีที่สาม เรียกว่า The latest-subsequence rules.

จะมีการนำลำดับของข้อมูล(Order) และข้อมูล ณ ปัจจุบัน (decency) เข้ามาพิจารณาด้วย ดังตัวอย่างเมื่อตารางล็อกได้ถูกแยกแยะด้วยวิธีการ the latest-subsequence rule จะได้ข้อมูลต่างๆ ดังตัวอย่าง จะสังเกตเห็นว่าส่วนของ W1 นั้นเพจสุดท้ายจะเป็นข้อมูลชุดปัจจุบัน นั้นคือ C เสมอ

W1	W2	Extracted Rules
A, B, C	D	(A, B, C) → D, (B, C) → D, (A, C) → D, (C) → D

ตารางที่ 3 แสดงตัวอย่างการใช้วิธี The latest subsequence rule representation

- วิธีการที่ สี่ เรียกว่า The substring rules.

จะมีการนำลำดับของข้อมูล (Order) และข้อมูลที่มีลักษณะอยู่ติดกัน (Adjacency) เข้ามาพิจารณาด้วย ดังตัวอย่างจะเห็นข้อมูลหรือกฎที่ผ่านการแยกแยะแล้วนั้นจะต้องมีการเรียงติดกันเสมอ ดังจะเห็นว่ากฎ A, C ➔ D นั้นจะไม่สามารถค้นพบด้วยวิธีดังกล่าวนี้ เพราะว่าเป็นข้อมูลที่ไม่ติดกัน

W1	W2	Extracted Rules
A, B, C	D	<A, B, C> → D, <A, B> → D, <B, C> → D, <A> → D, <B> → D, <C> → D

ตารางที่ 4 แสดงตัวอย่างการใช้งานของ the Substring rule representation

- วิธีการที่ ห้า เรียกว่า The latest-substring rules.

จะมีการนำลำดับของข้อมูล (Order), ข้อมูลที่มีลักษณะอยู่ติดกัน (Adjacency) และข้อมูล ณ ปัจจุบัน (decency) เข้ามาพิจารณาด้วย

W1	W2	Extracted Rules
A, B, C	D	<A, B, C> → D, <B, C> → D, <C> → D

ตารางที่ 5 แสดงตัวอย่างการใช้งานของ the latest substring rule representation

โดยให้แต่ละกฎนั้นจะอยู่ในรูปแบบ LHS➔RHS โดยจะมีการนิยามค่า Support และ confidence factor ขึ้นมาดังต่อไปนี้

$$\text{sup} = \frac{count(LHS, RHS)}{count(Table)} \qquad \text{sup}(LHS) = \frac{count(LHS)}{count(Table)} \qquad conf = \frac{\text{sup}(LHS, RHS)}{\text{sup}(LHS)}$$

จากสมการข้างต้น ซึ่งจากการนับค่าจากจำนวนการเกิดของตารางจะได้ Count (table) ซึ่งนับจากตารางล็อก มาใช้ในการคำนวณ

สิ่งสำคัญจากการนำวิธีการไมนิ่งกฎความสัมพันธ์ดังกล่าวข้างต้นไปใช้งานนั้นคือการกรองกฎบางกฎทิ้ง โดยค่า Minimum support และ minimum confidence หากกฎใดมีค่าเหล่านี้ต่ำกว่าค่าที่กำหนดไว้ ซึ่งการกระทำเหล่านี้เรียกว่าการพรุนนิ่งเป็นวิธีการที่ได้กล่าวไว้ในอัลกอริธึมไมนิ่งความสัมพันธ์ [4, 10, 11]

2.6 Tree construction

วิธีการสำคัญอีกวิธีหนึ่งเมื่อเราได้กฎมาแล้วคือการนำกฎเหล่านี้มาสร้าง Tree ก็คือ Latest substring index tree (LSIT) ซึ่งเป็นวิธีการสร้าง prediction model ที่มีประสิทธิภาพและมีการใช้หน่วยความจำน้อยที่สุด โดยเรามีหลักการสร้าง tree ดังต่อไปนี้

- แต่ละ กฎ เปรียบได้ดัง โหนดหนึ่งโหนด
- โหนดตัวบนแสดงเป็นโหนดแม่ (parent node) ซึ่งโหนดตัวล่างจะเป็นโหนดลูก (children node)
- รูทของ tree จะเป็น default rule

ดังตัวอย่างต่อไปนี้จะประกอบด้วย 6 โหนด ซึ่งสร้างมาจากการทำนายที่มีผลการทำนาย 5 กฎด้วยกัน

เมื่อสร้าง LSIT เรียบร้อยแล้วจะเห็นได้ว่าจะมีการเลื่อนหรือเปลี่ยนแปลงโหนดบางโหนด ด้วยวิธีการดังกล่าวต่อไปนี้ ขั้นตอนการพรุนนิ่ง (pruning process) จะใช้หลักการ post-order traverse โดย

- หากโหนดลูกมีค่า Confidence น้อยกว่า confidence แม่ โหนดดังกล่าวก็จะถูกตัดทิ้ง
- หากโหนดลูกมีการทำนาย เหมือนกับโหนดแม่ โหนดดังกล่าวก็จะถูกตัดทิ้งเช่นกัน

จะมีการโปรโมต บางโหนดภายใต้โหนดแม่ดังกล่าวนั้น ดังรูปที่ 12

โหนด <C> ➔ N จะถูกพรุน เพราะมีค่า confidence น้อยกว่า โหนดแม่

โหนด <A> ➔ M จะถูกพรุน เพราะว่ามีการทำนายเว็บเพจต่อไป เหมือนกับโหนดแม่

หลังจาก 2 โหนดดังกล่าวนี้ถูกพรุน

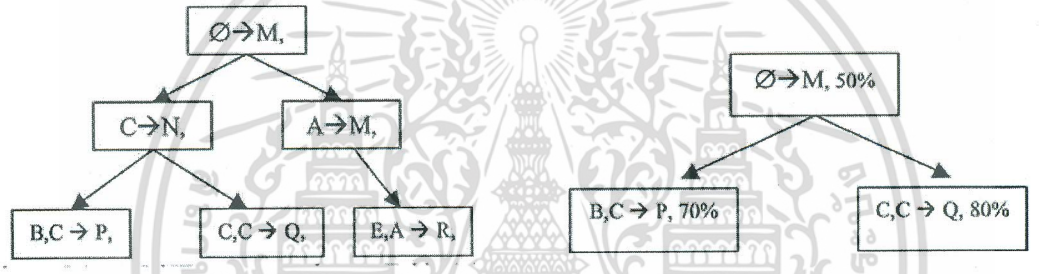
โหนด <E,A> ➔ R ก็จะถูกโปรโมต และเมื่อเปรียบเทียบจะเห็นว่ามีค่า Confidence น้อยกว่าโหนดแม่ ดังนั้นจึงถูกพรุนในลำดับต่อไป

หลังจากเสร็จสิ้นการพรุนจะได้ LSIT ดังรูปที่ 12

กรณีทดสอบหาก <C> ➔ ? ซึ่งจะให้การค้นหาโดยเพจ C เมื่อทำการค้นหาแล้วไม่เจอเพจ C ดังนั้นจะกฎสุดท้ายที่ได้คือ รูท M ดังนั้นการทำนายจึงให้เพจ M

กรณีทดสอบหาก <A, B, C> ➔ ? จะทำการค้นหาโดยให้เพจ C เมื่อทำการค้นหาไม่เจอต่อไปจะให้เส้นทาง B, C ซึ่งจะพบโหนด P ดังนั้นจึงไม่มีความจำเป็นในการค้นหาความสัมพันธ์ต่อไปแล้ว การทำนายจึงเป็น เพจ P

Rules	Pessimistic confidence
Ø → M	50%
<C> → N	40%
<A> → M	30%
<B, C> → P	70%
<C, C> → Q	80%
<E, A> → R	40%



รูปที่ 12 แสดงข้อมูลและวิธีการสร้างต้นไม้พร้อมทั้งการพรุนโหนดทิ้ง

โดยทั่วไป LSIT จะช่วยลดขนาดในการแบ่งชั้น (Classifier) ซึ่งจากการเปรียบเทียบจะเห็นได้ว่าขนาดของ Latest-substring rules set เมื่อผ่านการ LSIT pruning ขนาดของกฎที่เหลือจะมีขนาดน้อยลง จากงานวิจัยของ Tianyi Li [6] ระบุว่าด้วยวิธีการดังกล่าวนี้กฎต่างๆที่ไม่จำเป็นจะถูกตัดทิ้งเป็นปริมาณสัดส่วนถึง 4/5 จะถูกตัดทิ้งออกไป

3. อัลกอริธึมใหม่สำหรับการค้นหารูปแบบลำดับของเว็บไซต์

3.1 อัลกอริธึมที่ใช้ในงานวิจัย

Begin main:

L1 = {large 1-sequences};

For { k=2; $L_{k-1} \neq \varnothing$; k++} do

Begin

C_k New candidate generated from L_{k-1}

(See a new candidate generation)

for each web-page sequence c in the web transaction table do

Increment the count of all candidates in C_k that are contained in c.

L_k = Candidates in C_k with minimum support.

End

รูปที่ 13 A New Algorithm for Web Sequential Pattern Discovery

The Candidate generate function take as argument L_{k-1}, the set of all large (k-1)-sequences. The function work as follows. First. Join L_{k-1} with L_{k-1}

$$a_1 a_2 a_3 \ldots a_{k-1} \otimes b_1 b_2 b_3 \ldots b_{k-1} = \begin{cases} \phi & \text{Otherwise} \\ a_1\, a_2\, a_3 \ldots a_n & \\ a_2 = b_1, a_2 = b_3, \ldots, a_{k-1} = b_{k-2} & \end{cases}$$

รูปที่ 14 A new Candidate Generations

ในการสร้าง C_k นั้นเราได้ให้คำนิยามการ Join Operation ใหม่ซึ่งดังจะได้ทราบมาแล้วว่าในลักษณะของ Sequence Pattern นั้นจะเกิดการกระโดดกันของ Item ที่เกิดจากการ Join ดังนั้น Sequence ที่ได้จะขาดลำดับ (Order information) เมื่อเรานำมาประยุกต์ใช้กับเว็บเพจจะทำให้เว็บเพจเกิดการกระโดดได้เช่นกัน ดังนั้นจะส่งผลให้ความแม่นยำลดลงไป หรือการทำนายผิดเพี้ยนไปด้วย

ดังตัวอย่างนี้เป็น: A new Candidate Generation

$$DE \otimes ED \Rightarrow DE \rightarrow D$$
$$DE \otimes EY \Rightarrow DE \rightarrow Y$$
$$ED \otimes DE \Rightarrow ED \rightarrow E$$
$$DE \otimes EZ \Rightarrow DE \rightarrow Z$$

ได้มีการกำหนดกฎเพื่อใช้ในการทดลอง โดยกฎเหล่านี้ได้กำหนดขึ้นมาดังนี้ คือ D,E → Y, X1 → C, D → E, C → D, D, E → Y ซึ่งกฎดังกล่าวข้างต้นเหล่านี้ได้นำไปเพิ่มเติมในตารางล็อก แล้วนำข้อมูลผ่านขั้นตอนของอัลกอริธึมที่ได้ออกแบบไว้ โดยตารางที่ 6 โดยข้อมูลทั้งหมดที่ 7 IP ด้วยกันโดยมีการเรียกใช้เว็บเพจทั้งหมด 5 เพจ เมื่อข้อมูลได้ผ่านขั้นตอน Cleaning Model โดยขั้นจะผ่านอัลกอริธึมในรูปที่ 13 โดยจะได้ข้อมูลเป็นลักษณะ Moving Pair Windows โดยที่หากว่าผู้เรียกใช้เว็บเพจมีการเรียกใช้เว็บเพจที่เหมือนกัน อัลกอริธึมจะนับการเกิดของ Pattern นั้นเพียงครั้งเดียว

ตัวอย่าง IP1 มีรูปแบบการเข้า D → E, E → D, D → E, E → Y ดังจะเห็นว่า D → E นั้นเกิดขึ้นด้วยกัน 2 ครั้ง หลังจากผ่านอัลกอริธึมแล้วจะได้ Candidate Generation C_1 ดังตารางที่ 7

IP1	D	E	D	E	Y
IP2	X4	X5	D	E	Y
IP3	X1	C	D	E	Z
IP4	X1	C	D	E	Z
IP5	X1	C	D	E	Z
IP6	E	D	E	Y	X
IP7	D	E	Y	X1	X2

ตารางที่ 6 แสดงข้อมูลที่ใช้ในการทดลอง

RULE	LSH	Support
D	7	1
E	7	1
Y	4	0.571429
X4	1	0.142857
X5	1	0.142857
X1	4	0.571429
C	3	0.428571
Z	3	0.428571
X	1	0.142857

ตารางที่ 7 Candidate Generation C_1

ดังจะเห็นว่าข้อมูลบางส่วนจะถูกกรองออกไปหากข้อมูลนั้นมีค่า (support $\leq$ 0.142) ดังเช่น X4 ,X5, X, X2 และกฎที่เหลือจะถูกบันทึกไว้ดังนี้คือ D, E, Y, X1, C, Z. ดังรูปในตารางที่ 8 กฎบางกฎจะถูกส่งผ่านไปยัง Candidate Generation ซึ่งเป็นการ Join Operation ดังตัวอย่างต่อไปนี้ เป็นการ Operation Join ของ C3 = L2 $\otimes$ L2

$DE \otimes ED \Rightarrow DE \rightarrow D : ED \otimes DE \Rightarrow ED \rightarrow E$

$DE \otimes EY \Rightarrow DE \rightarrow Y : X1C \otimes CD \Rightarrow X1C \rightarrow Y$

$DE \otimes EZ \Rightarrow DE \rightarrow Z : CD \otimes DE \Rightarrow CD \rightarrow E$

ค่าสุดท้ายที่บันทึกในตารางที่ 9 คือ Candidate Generation C3 โดยตารางสุดท้ายคือ ตารางที่ 10 จะเหลือ Larger Sequence L5

RULE	LSH	Support
D	7	1
E	7	1
Y	4	0.571429
X1	4	0.571429
C	3	0.428571
Z	3	0.428571

ตารางที่ 8 Larger Sequence L1

RULE	LSH	LSH-RSH	Supt LSH-RSH	Con
DE -> D	8	1	0.142857143	0.125
DE -> Y	8	4	0.571428571	0.5
DE -> Z	8	3	0.428571429	0.375
ED -> E	2	2	0.285714286	1
X1 C -> D	3	3	0.428571429	1
CD -> E	3	3	0.428571429	1

ตารางที่ 9 Candidate Generation C3

RULE	LSH	LSH-RSH	Sup LSH-RSH	Con
X1 C D E -> Z	3	3	0.428571429	1

ตารางที่ 10 Larger Sequences L5

การสิ้นสุดในส่วนของอัลกอริธึมนั้นเกิดจาก เมื่อไม่มีการสร้าง Candidate นั้นเอง เราไม่สามารถสร้าง Candidate ขึ้นมาได้อีกแล้ว ซึ่งนั้นเป็นการสิ้นสุดการทำงานส่วนการสร้างกฎนั้นเอง

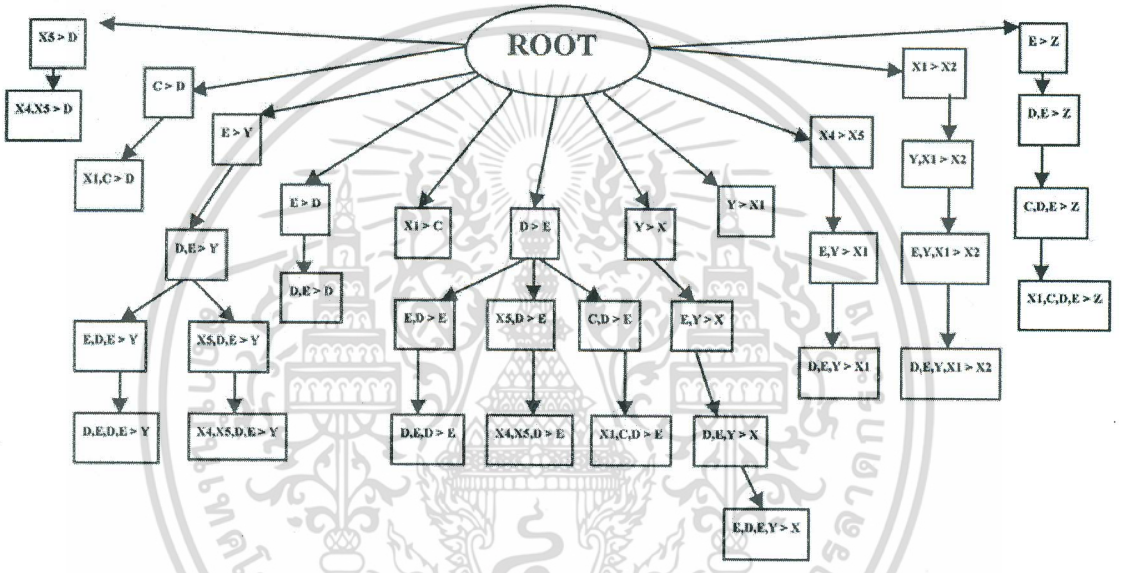
3.2 การสร้างต้นไม้ (Tree Pruning)

ในขั้นตอนต่อไปเราจะนำกฎมาสร้างเป็นลักษณะต้นไม้ (Tree) โดยมีหลักการสร้างดังต่อไปนี้ แต่ละ กฎ เปรียบได้ดัง โหนดหนึ่งโหนด

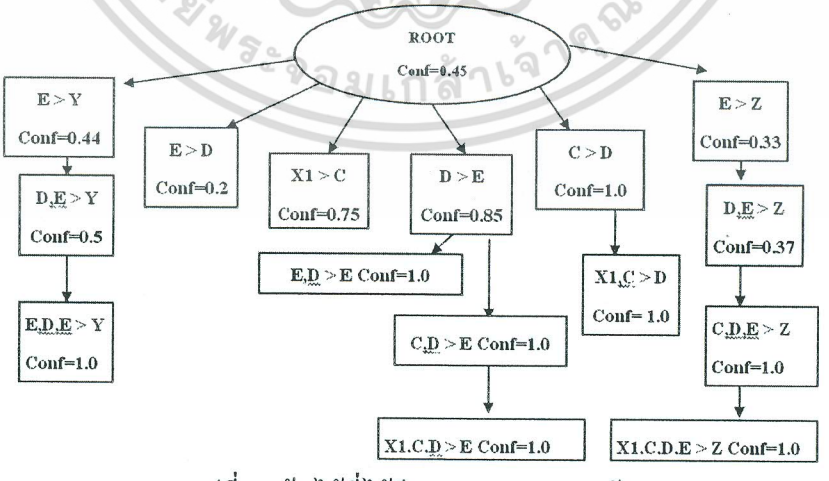
- โหนดตัวบนแสดงเป็นโหนดแม่ (parent node) ซึ่งโหนดตัวล่างจะเป็นโหนดลูก (children node)
- รูทของ tree จะเป็น default rule

ในการพรุนนิ่งนั้นจะใช้วิธีการ post-order ดังที่ได้กล่าวไว้ข้างต้น ดังจะเห็นได้ว่าในรูปที่ 16

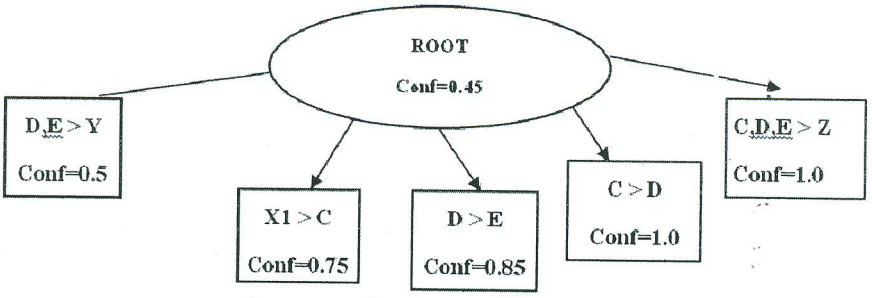
- E ➔ $Y_{Conf} < Root_{Conf}$ จะถูกพรุนเพราะเมื่อเปรียบเทียบค่า Confidence แล้วค่าของโหนดนั้นมีค่าน้อยกว่าตัวแม่ และทำการโปรโมต D,E ➔ Y เช่นเดียวกัน E,D,E ➔ Y จะถูกพรุนเพราะเหตุว่ามีค่า Confidence น้อยกว่าตัวโหนดแม่เช่นกัน ซึ่งตัวโหนดแม่ดังกล่าวตัวปัจจุบันนี้คือ D, E ➔ Y.
- E ➔ $D_{Conf} < Root_{Conf}$ จะถูกพรุนเพราะมีค่า Confidence น้อยกว่าตัวแม่
- E,D ➔ E and C,D ➔ E จะถูกพรุนเพราะว่าในส่วนการทำนายให้ผลเหมือนกับโหนดแม่
- E ➔ $Z_{Conf} < Root_{Conf}$ จะถูกพรุนและทำการโปรโมต D,E ➔ Z เป็นโหนดแม่ของ C,D,E ➔ Z.
- D,E ➔ $Z_{Conf} < Root_{Conf}$ จะถูกพรุนเพราะว่าค่า Confidence มีค่าน้อยกว่าตัวแม่ และทำการโปรโมต C,D,E ➔ Z.



รูปที่ 15 ต้นไม้แบบดั้งเดิม สร้างจากตารางล็อก



รูปที่ 16 ต้นไม้ที่ได้มีการลดกฎบางกฎแล้ว



รูปที่ 17 ต้นไม้ที่ผ่านการพรุนเรียบร้อยแล้ว

4. เอกสารอ้างอิง

[1] J. Srivastava, R. Cooley, M. Deshpande and P. Tan, "Web Usage mining: Discovery and Applications of Usage Patterns from Web Data", ACM SIGKDD , Jan 2000.

[2] U. Fayyad,G. Piatetsky-Shapiro,and P. Smyth. From data mining to knowledge discovery: An overview. InProc. ACM KDD, 1994.

[3] M.S. Chen, J. Han, and P.S. Yu. Data mining: An overview from a database perspective. IEEE Transactions on Knowledge and Data Engineering, 8(6):866-883, 1996.

[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. of the 20th VLDB Conference, Santiago, Chile, 1994.

[5] J.Pitkow and P.Pirolli. Mining Longest Repeating Subsequences to Predict World Wide Web Surfing. In Second USENIX Symposium on Internet Technologies and Systems, C0, 1999.

[6] lan Tian Yi Li. Web-document prediction and presending using association rule sequential classifiers, Simon Fraser University, S2001

[7] Z. Su, Q. Yang, Y. Lu, and H. Zhang. Whatnext: A Prediction System for Web Requests Using N-gram Sequence Models. In Proc. of the First Int'l Conf. on Web Information Systems and Engineering Conference, Hong Kong June 2000.

[8] Q. Yang, H. Zhang, and T. Li.Mining Web Logs for Prediction Models in WWW Caching and Prefetching. In Proc. ACM SIGKDD, 2001.

[9] I.Zukerman, D.W.Albrecht, and A.E.Nicholson. Predicting User's Request on the WWW. In Proce., Monash University, June 1999.

[10] R. Agrawal and R. Srikant. Fast algorithms for mining associatin rules. In Proc. Of the VLDB Conference, Santiago, Chile, September 1994. Expanded version available as IBM Research Report RJ9839, June 1994.

[11] R. Agrawal and R. Srikant. Mining Sequential patterns. Research Report RJ 9910, IBM Almaden Research Center, Sanjose, California, October 1994.