บทที่ 2
ทบทวนวรรณกรรมที่เกี่ยวข้อง.

2.1 บทนำ

วรรณกรรมที่เกี่ยวข้องกับงานวิจัยนี้ถูกเรียบเรียงไว้ตามลำดับ เริ่มต้นจากแนวคิดของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ (Association Rules Discovery) รวมถึงค่าที่ใช้ในกจรประเมินความน่าสนใจของกฎความสัมพันธ์ (Interestingness Measure of Association Rules) ที่มีงานวิจัยในอดีตเคยเสนอไว้ทั้งหมด 8 ค่า ต่อมาจะอธิบายถึงตัวแบบการประเมินความน่าสนใจของกฎความสัมพันธ์ใหม่ โดยเริ่มจากการอธิบายถึงข้อบกพร่องของตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นดั้งเดิม ตามด้วยการอธิบายตัวแบบการประเมินความน่าสนใจของกฎความสัมพันธ์แบบใหม่ และต่อด้วยการอธิบายคุณสมบัติที่ค่าประเมินความน่าสนใจควรจะมีทั้งหมด 16 คุณสมบัติ รวมถึงสรุปคุณสมบัติที่ค่าประเมินความน่าสนใจทั้ง 8 ค่าและค่าความเชื่อมั่นใหม่มี หัวข้อถัดมาเป็นการอธิบายแนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไข (Concept of Revision Control, Version Control) ซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control Software, Version Control Software) รวมถึงระบบคอนเคอเรนท์เวอร์ชั่น (Concurrent Versions System, CVS) ซึ่งเป็นระบบที่ผู้วิจัยสนใจนำข้อมูลมาทำการทดสอบ หัวข้อถัดมาเป็นการรวบรวมวรรณกรรมที่เกี่ยวข้องกับการประยุกต์ใช้การทำเหมืองข้อมูลด้วยเทคนิคค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Applying Association Rule Discovery in Software Archive) ตั้งแต่อดีตจนถึงปัจจุบัน รวมถึงอธิบายถึงขั้นตอนวิธีการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Data Mining in Software Archives) ของงานวิจัยต่างๆในอดีตอย่างละเอียด ตั้งแต่การจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Preparing Data for Mining in Software Archives) จนถึงขั้นตอนการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Data Mining in Software Archives) และขั้นตอนการสร้างคำแนะนำจากกฎความสัมพันธ์ (Generating Suggestions for Situation) สุดท้ายจะกล่าวถึงการวัดประสิทธิภาพของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์

2.2 การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ (Association Rules Discovery)

การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ (Association Rules discovery) คือการค้นหาความสัมพันธ์ที่มีความเกี่ยวข้องและเชื่อมโยงกันระหว่างรายการข้อมูล (Data Items) ภายในฐานข้อมูลขนาดใหญ่ที่มีอยู่เพื่อนำไปใช้ในการวิเคราะห์หรือทำนายปรากฏการณ์ต่างๆ (Agrawal et al., 1993) ตัวอย่างในการนำกฎความสัมพันธ์ไปประยุกต์ใช้ในทางปฏิบัติที่ได้รับความนิยมมากที่สุดคือ การวิเคราะห์พฤติกรรมการซื้อของลูกค้า (Market Basket Analysis)

นิยามทางคณิตศาสตร์ของการค้นหากฎความสัมพันธ์นั้นสามารถอธิบายได้ดังนี้ กำหนดให้ เซต $I = \{i_1, i_2, \ldots, i_m\}$ เป็นเซตของรายการข้อมูล (items) ที่มีอยู่ทั้งหมดและให้ เซต $T = \{t_1, t_2, \ldots, t_n\}$ เป็นเซตของทรานแซคชั่น โดยที่แต่ละทรานแซคชั่น t_n ประกอบด้วยเซตย่อย I_j (j = 1, 2, .. m) ที่เป็นเซตย่อยของเซตของรายการข้อมูล I เซตของรายการข้อมูล I_j นั้นถูกเรียกว่า เซตรายการ (Itemset) ดังนั้นกฎความสัมพันธ์ r ก็คือคู่ของเซตรายการ I_1 และเซตรายการ I_2 โดยที่เซตรายการ I_1 และเซตรายการ I_2 เป็นเซตย่อยของเซต I ที่ไม่มีสมาชิกที่ซ้อนทับกันและเซตรายการ I_2 ไม่เท่ากับเซตว่าง เซตรายการ I_1 ถูกเรียกว่า เซตรายการที่มาก่อน (Antecedent Itemset) และเซตรายการ I_2 ถูกเรียกว่า เซตรายการที่ตามมา (Consequent Itemset) และกำหนดสัญลักษณ์ $I_1 \rightarrow I_2$ แทนกฎความสัมพันธ์ที่มีเซตรายการ I_1 เป็นเซตรายการที่มาก่อน และเซตรายการ I_2 เป็นเซตรายการที่ตามมา โดยที่ I_2 ไม่ใช่เซตว่าง (Olivier et al., 2008)

วิธีการที่ดีวิธีหนึ่งที่จะระบุว่ากฎความสัมพันธ์นั้นๆมีความตรงประเด็นมากน้อยเพียงใด คือการกำหนดค่าที่ใช้ในการประเมินคุณภาพของกฎความสัมพันธ์หรือที่เรียกว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ ตัวแบบของการประเมินความน่าสนใจของกฎความสัมพันธ์ที่เป็นตัวแบบแรกและดั้งเดิมคือตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น (Support-Confidence Model) ซึ่งใช้ค่า 2 ค่าในการประเมิน ค่าแรกถูกเรียกว่าค่าสนับสนุน (Support) ของกฎความสัมพันธ์ ใช้สัญลักษณ์ Support($I_1 \rightarrow I_2$) ค่าสนับสนุนนี้สามารถคำนวณได้จากจำนวนของทรานแซคชั่นที่มีทั้งส่วนเซตรายการที่มาก่อน (I_1) และเซตรายการที่ตามมา (I_2) ของกฎความสัมพันธ์หารด้วยจำนวนของทรานแซคชั่นทั้งหมด ซึ่งค่าสนับสนุนนี้ก็คือค่าความน่าจะเป็นที่จะมีทรานแซคชั่นที่มีทั้งรายการในเซตรายการที่มาก่อนและเซตรายการที่ตามมาในทรานแซคชั่นทั้งหมดนั่นเอง ในบางครั้งค่าสนับสนุนของกฎความสัมพันธ์นั้นอาจถูกนำไปใช้ในรูปร้อยละของจำนวนทรานแซคชั่น

ทั้งหมด ค่าสนับสนุนถูกใช้ในการประเมินความถี่ (Frequent) ในการปรากฏของรายการ (Item) หรือเซตรายการ หรือกฎความสัมพันธ์ นอกจากการพิจารณาค่าสนับสนุนของกฎความสัมพันธ์ก็คือการพิจารณาค่าความเชื่อมั่น (Confidence) ของกฎความสัมพันธ์ ใช้สัญลักษณ์ Conf($I_1 \rightarrow I_2$)) ค่าความเชื่อมั่นนี้ก็คืออัตราส่วนของค่าสนับสนุนของกฎความสัมพันธ์กับค่าสนับสนุนของเซตรายการที่มาก่อน หรืออาจเรียกได้ว่าค่าความเชื่อมั่นก็คือค่าความน่าจะเป็นที่จะมีรายการทุกรายการในเซตรายการที่มาก่อนแล้วจะมีรายการทุกรายการในเซตรายการที่ตามมาด้วย ค่าความเชื่อมั่นนั้นถูกนำไปใช้ในการประเมินระดับความน่าสนใจของกฎความสัมพันธ์

นอกจากตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น (Support-Confidence Model) แล้วการประเมินระดับความน่าสนใจของกฎความสัมพันธ์ยังสามารถใช้วิธีการอื่นๆได้ ซึ่งถูกรวบรวมเอาไว้และอธิบายอย่างละเอียดในหัวข้อถัดไป

2.3 การประเมินความน่าสนใจของกฎความสัมพันธ์ (Interestingness Measure of Association Rules)

หัวข้อนี้ได้รวบรวมวรรณกรรมต่างๆที่เกี่ยวข้องกับการเสนอตัวแบบที่นำมาใช้ในการประเมินความน่าสนใจของกฎความสัมพันธ์ตั้งแต่ในอดีตจนถึงในปัจจุบันอันได้แก่ ค่าสนับสนุน (Support), ค่าความเชื่อมั่น (Confidence), ค่าคอนวิคชั่น (Conviction), ค่าลิฟท์ (Lift), ค่าเลฟเวอเรจ (Leverage), ค่าคัฟเวอเรจ (Coverage), ค่าสหสัมพันธ์ (Correlation) และ ค่าอัตราส่วนออดส์ (Odds Ratio) ตามลำดับ รวมถึงวรรณกรรมที่มีการสนับสนุนตัวแบบข้างต้นและวรรณกรรมที่พิสูจน์ข้อบกพร่องของตัวแบบบางตัวด้วย เพื่อให้ง่ายต่อการเปรียบเทียบกันของทุกตัวแบบที่นิยามขึ้นมาจากค่าทางสถิติจึงกำหนดนิยามค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูลดังนี้

กำหนดให้ P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล
n(X) คือ จำนวนของทรานแซคชั่นที่มีรายการ X ปรากฏอยู่
N คือ จำนวนของทรานแซคชั่นทั้งหมดในฐานข้อมูล

$$P(X) = \frac{n(X)}{N}$$

2.3.1 ค่าสนับสนุน (Support)

ค่าสนับสนุนถูกนำเสนอขึ้นมาครั้งแรกโดย Agrawal และคณะในปีค.ศ. 1993 (Agrawal et al., 1993) ค่าสนับสนุนคือค่าความน่าจะเป็นของรายการนั้นนั่นเอง โดยปกติค่าสนับสนุนนั้นเป็นค่าที่ถูกใช้ในการแสดงความถี่ในการปรากฏของแต่ละรายการ แต่ค่าสนับสนุนนี้ก็สามารถนำมาใช้ในการประเมินความน่าสนใจของกฎความสัมพันธ์ได้ สมการในการคำนวณค่าสนับสนุนของรายการ X แสดงได้ดังนี้

กำหนดให้ Support(X) คือ ค่าสนับสนุนของรายการ X

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$$Support(X) = P(X)$$

การหาค่าสนับสนุนของรายการหรือบางที่ก็ถูกเรียกว่าค่าความถี่ของรายการ เซตรายการที่มีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำที่กำหนดไว้จะเรียกเซตรายการนั้นว่าเซตรายการความถี่สูง (Frequent Itemset) หรือเซตรายการใหญ่ (large Itemset) หรือกล่าวคือค่าสนับสนุนนี้จะถูกนำมาใช้เพื่อเป็นการคัดกรองรายการทั้งหมดในฐานข้อมูลให้เหลือแต่เพียงรายการที่มีความถี่สูงตามที่กำหนดไว้เพื่อนำเซตรายการเหล่านั้นไปค้นหากฎความสัมพันธ์ในภายหลัง

ในกรณีที่จะใช้ค่าสนับสนุนมาประเมินระดับความน่าสนใจ ก็คือค่าความถี่ของกฎความสัมพันธ์นั้นนั่นเอง สมการในการคำนวณค่าสนับสนุนของกฎความสัมพันธ์ แสดงได้ดังนี้

กำหนดให้ Support(R) คือ ค่าสนับสนุนของกฎความสัมพันธ์ R

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

P(Y) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ Y ในฐานข้อมูล

P(X and Y) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และ Y ในฐานข้อมูล

$$\text{Support}(X \rightarrow Y) = P(X \text{ and } Y)$$

ข้อเสียของการใช้ค่าสนับสนุนก็คือการก่อให้เกิดปัญหาขาดแคลนรายการ (rare item problem) เนื่องจากบางรายการนั้นเป็นรายการที่ปรากฏอยู่ในฐานข้อมูลจำนวนน้อยจึงถูกคัดออกไป แต่ในความจริงนั้นรายการที่ปรากฏอยู่ในฐานข้อมูลจำนวนน้อยอาจสามารถนำไปสร้างเป็นกฎความสัมพันธ์ที่น่าสนใจหรือมีคุณค่าได้ ค่าที่เป็นไปได้ของค่าสนับสนุนนั้นอยู่ในพิสัย [0, 1] ถ้าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์เป็นอิสระต่อกันแล้วค่าสนับสนุนจะเท่ากับ 0 ถ้าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์มีการปรากฏขึ้นพร้อมกันเสมอค่าสนับสนุนจะเท่ากับ 1 (Agrawal et al., 1993; Sheikh et al., 2004)

2.3.2 ค่าความเชื่อมั่น (Confidence)

ค่าความเชื่อมั่นถูกนำเสนอขึ้นมาครั้งแรกโดย Agrawal และคณะในปีค.ศ. 1993 (Agrawal et al., 1993) เช่นเดียวกับค่าสนับสนุนนิยามของค่าความเชื่อมั่นคือความน่าจะเป็นที่จะพบกฎความสัมพันธ์ $X \rightarrow Y$ ในทรานแซคชั่นที่มีรายการ X ปรากฏอยู่ ดังนั้นการคำนวณค่าความเชื่อมั่นของกฎความสัมพันธ์ $X \rightarrow Y$ และกฎความสัมพันธ์ $Y \rightarrow X$ นั้นจะให้ค่าที่แตกต่างกัน สมการในการคำนวณค่าความเชื่อมั่นแสดงได้ดังนี้

กำหนดให้ Conf(R) คือ ค่าความเชื่อมั่นของกฎความสัมพันธ์ R

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

P(X and Y) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และ Y ในฐานข้อมูล

P(Y|X) คือ ค่าความน่าจะเป็นในการพบรายการ Y ในทรานแซคชั่นที่มีรายการ X อยู่แล้ว

$$\text{Conf}(X \rightarrow Y) = \frac{\mathbf{P(X \text{ and } Y)}}{\mathbf{P(X)}}$$

หรือ $$\text{Conf}(X \rightarrow Y) = P(Y|X)$$

เนื่องจากค่าความเชื่อมั่นและค่าสนับสนุนนั้นถูกพัฒนาขึ้นมาพร้อมกันโดย Agrawal และคณะ ค่าสนับสนุนจะถูกนำมาใช้เพื่อเป็นการคัดกรองรายการทั้งหมดในฐานข้อมูลให้เหลือแต่เพียงรายการที่มีความถี่สูงตามที่กำหนดไว้ ต่อจากนั้นค่าความเชื่อมั่นจะถูกนำมาใช้ในการสร้างกฎ

ความสัมพันธ์ขึ้นมาจากผลลัพธ์เซตรายการที่มีความถี่สูงที่ได้มาจากการหาค่าสนับสนุนโดยการคำนวณหาค่าความเชื่อมั่นของทุกกฎความสัมพันธ์ที่เป็นไปได้จากเซตรายการที่มีความถี่สูงทั้งหมด ค่าความเชื่อมั่นของกฎความสัมพันธ์ใดที่มากกว่าค่าความเชื่อมั่นขั้นต่ำ (Minimum Confidence) ที่กำหนดไว้จะถือว่ากฎความสัมพันธ์นั้นมีอยู่จริงและมีความน่าสนใจ

ปัญหาที่เกิดขึ้นจากการใช้ค่าความเชื่อมั่นคือค่าความเชื่อมั่นนั้นค่อนข้างจะอ่อนไหวง่ายต่อความถี่ของเซตรายการที่ตามมา กล่าวคือถ้ากฎความสัมพันธ์นั้นมีเซตรายการที่ตามมาที่มีความถี่สูงมาก (ในขณะที่เซตรายการที่มาก่อนมีความถี่ที่น้อยกว่ามาก) จะทำให้ค่าความเชื่อมั่นของกฎศวามสัมพันธ์มีค่าสูงส่งผลให้กฎความสัมพันธ์นั้นถูกพิจารณาว่ามีความน่าสนใจ ซึ่งในความจริงเซตรายการที่มาก่อนและเซตรายการที่ตามมาอาจมีความสัมพันธ์กันน้อยมากก็ได้ (Sheikh et al., 2004) ค่าที่เป็นไปได้ของค่าความเชื่อมั่นนั้นอยู่ในพิสัย [0, 1] ถ้าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์เป็นอิสระต่อกันแล้วค่าความเชื่อมั่นจะเท่ากับ 0 ถ้าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์มีการปรากฏขึ้นพร้อมกันเสมอค่าความเชื่อมั่นจะเท่ากับ 1 (Agrawal et al., 1993; Sheikh et al., 2004)

2.3.3 ค่าคอนวิคชั่น (Conviction)

ค่าคอนวิคชั่นถูกเสนอขึ้นมาครั้งแรกโดย Brin และคณะในปีค.ศ. 1997 (Brin et al., 1997) ค่าคอนวิคชั่นถูกพัฒนาขึ้นมาเพื่อแก้ไขข้อด้อยของค่าความเชื่อมั่นที่ไม่สามารถจะบ่งบอกทิศของกฎความสัมพันธ์ได้ภายในการคำนวณครั้งเดียว กล่าวคือการใช้ค่าความเชื่อมั่นจะต้องมีการคำนวณทั้งค่าความเชื่อมั่นของกฎความสัมพันธ์ทั้งไปและกลับเพื่อเลือกกฎความสัมพันธ์ที่มีค่าค่าความเชื่อมั่นสูงกว่ากัน ค่าคอนวิคชั่นของกฎความสัมพันธ์ $X \rightarrow Y$ นั้นจะเปรียบเทียบความน่าจะเป็นในการมีรายการ X ปรากฏโดยที่ไม่มีรายการ Y ปรากฏ สมการในการคำนวณค่าคอนวิคชั่นแสดงได้ดังนี้

กำหนดให้ Conviction(R) คือ ค่าคอนวิคชั่นของกฎความสัมพันธ์ R

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$P(\overline{X})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่ไม่มีรายการ X ในฐานข้อมูล

$P(X \text{ and } \overline{Y})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X แต่ไม่มี

รายการ Y ในฐานข้อมูล

$$\text{Conviction}(X \rightarrow Y) = \frac{P(X)P(\overline{Y})}{P(X \text{ and } \overline{Y})}$$

โดยที่ $P(\overline{Y})$ คือความน่าจะเป็นที่ไม่มีรายการ Y ปรากฏในทรานแซคชั่น ค่าที่เป็นไปได้ของค่าคอนวิคชั่นนั้นอยู่ในพิสัย [0, $+\infty$] ถ้าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์เป็นอิสระต่อกันแล้วค่าคอนวิคชั่นจะเท่ากับ 1 แต่ถ้าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์มีการปรากฏขึ้นพร้อมกันเสมอค่าคอนวิคชั่นจะเท่ากับ $+\infty$ (Brin et al., 1997; Sheikh et al., 2004)

2.3.4 ค่าลิฟท์ (Lift)

ค่าลิฟท์ถูกเสนอขึ้นมาครั้งแรกโดย Brin และคณะในปีค.ศ. 1997 (Brin et al., 1997) ค่าลิฟท์นั้นเป็นค่าที่ใช้ในการประเมินความถี่ในการปรากฏร่วมกันของเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์โดยที่เซตรายการที่มาก่อนและเซตรายการที่ตามมานั้นเป็นอิสระต่อกันทางสถิติ ข้อดีประการหนึ่งของการคำนวณค่าลิฟท์คือการใช้ค่าลิฟท์จะไม่พบกับปัญหาขาดแคลนรายการ สมการในการคำนวณค่าลิฟท์แสดงได้ดังนี้

กำหนดให้ Lift(R) คือ ค่าลิฟท์ของกฎความสัมพันธ์ R

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

P(X and Y) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และรายการ Y ในฐานข้อมูล

P(Y|X) คือ ค่าความน่าจะเป็นในการพบรายการ Y ในทรานแซคชั่นที่มีรายการ X อยู่แล้ว

Conf(R) คือ ค่าความเชื่อมั่นของกฎความสัมพันธ์ R

$$\text{Lift}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(X)P(Y)}$$

หรือ

$$\text{Lift}(X \rightarrow Y) = \frac{P(Y|X)}{P(Y)}$$

หรือ $$\text{Lift}(X \rightarrow Y) = \frac{\text{Conf}(X \rightarrow Y)}{\text{P(Y)}}$$

ค่าที่เป็นไปได้ของค่าลิฟท์นั้นอยู่ในพิสัย [0, +∞] ถ้าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์เป็นอิสระต่อกันแล้วค่าลิฟท์จะเท่ากับ 1 ถ้าค่าลิฟท์มีค่ามากกว่า 1 หมายถึงเซตรายการที่มาก่อนและเซตรายการที่ตามมามีความสัมพันธ์เชิงบวกต่อกัน ถ้าค่าลิฟท์มีค่าน้อยกว่า 1 หมายถึงเซตรายการที่มาก่อนและเซตรายการที่ตามมามีความสัมพันธ์เชิงลบต่อกัน (Brin et al., 1997; Sheikh et al., 2004)

2.3.5 ค่าเลฟเวอเรจ (Leverage)

ค่าเลฟเวอเรจถูกเสนอขึ้นมาครั้งแรกโดย Piatetsky-Shapiro และคณะในปี 1991 (Piatetsky-Shapiro et al., 1991) ค่าเลฟเวอเรจถูกพัฒนาขึ้นมาเพื่อประเมินความต่างของค่าความน่าจะเป็นของการปรากฏเซตรายการที่มาก่อนและเซตรายการที่ตามมาขึ้นพร้อมกันโดยที่เซตรายการทั้งสองนั้นมีการขึ้นต่อกันทางสถิติกับค่าความน่าจะเป็นของการปรากฏเซตรายการที่มาก่อนและเซตรายการที่ตามมาที่ปรากฏอย่างเป็นอิสระต่อกันในฐานข้อมูล สมการในการคำนวณค่าเลฟเวอเรจแสดงได้ดังนี้

กำหนดให้ Leverage(R) คือ ค่าเลฟเวอเรจของกฎความสัมพันธ์ R

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

P(X and Y) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และรายการ Y ในฐานข้อมูล

$$\text{Leverage}(X \rightarrow Y) = \text{P(X and Y)} - \text{P(X)P(Y)}$$

เหตุผลสำคัญที่มีการพัฒนาค่าเลฟเวอเรจขึ้นมานั้นมาจากความต้องการตั้งวิธีการขายสินค้า โดยทำการค้นหาว่าการขายสินค้าทั้งสองเซตรายการร่วมกันกับการขายที่เป็นอิสระต่อกันแบบไหนมีค่ามากกว่ากัน ข้อด้อยสำคัญของค่าเลฟเวอเรจก็คือการใช้ค่านี้อาจทำให้ประสบกับปัญหาขาดแคลนรายการได้ ค่าที่เป็นไปได้ของค่าเลฟเวอเรจนั้นอยู่ในพิสัย [-∞, +∞] (Piatetsky-Shapiro et al., 1991; Sheikh et al., 2004)

2.3.6 ค่าคัฟเวอเรจ (Coverage)

ค่าคัฟเวอเรจคือค่าสนับสนุนของเซตรายการที่มาก่อน หรือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการที่มาก่อนในฐานข้อมูลนั่นเอง แนวคิดของค่าคัฟเวอเรจคือการใช้ความถี่ของเซตรายการที่มาก่อนของกฎความสัมพันธ์มาใช้เป็นตัวระบุถึงความน่าสนใจของกฎความสัมพันธ์นั้น สมการในการคำนวณค่าคัฟเวอเรจแสดงได้ดังนี้

กำหนดให้ Coverage (R) คือ ค่าคัฟเวอเรจของกฎความสัมพันธ์ R

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$$\text{Coverage}(X \rightarrow Y) = P(X)$$

ค่าที่เป็นไปได้ของค่าคัฟเวอเรจนั้นอยู่ในพิสัย [0, 1] (Michael., 2009)

2.3.7 ค่าสหสัมพันธ์ (Correlation)

ค่าสหสัมพันธ์หรือค่าสหสัมพันธ์นี้เป็นเทคนิคทางสถิติที่สามารถแสดงได้ว่าเซตรายการใดมีความสัมพันธ์กันอย่างแข็งแรง สมการในการคำนวณค่าสหสัมพันธ์แสดงได้ดังนี้

กำหนดให้ Corr(R) คือ ค่าสหสัมพันธ์ของกฎความสัมพันธ์ R

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

P(X and Y) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และรายการ Y ในฐานข้อมูล

$$\text{Corr}(X \rightarrow Y) = \frac{(P(X \text{ and } Y) - P(X)P(Y))}{\sqrt{P(X)P(Y)P(1-P(X))(1-P(Y))}}$$

ค่าสหสัมพันธ์นี้สามารถนำมาใช้ประเมินลักษณะของความสัมพันธ์ระหว่างเซตรายการที่มาก่อนและเซตรายการที่ตามมาได้ ค่าที่เป็นไปได้ของค่าสหสัมพันธ์นั้นอยู่ในพิสัย [-1, 1] โดยที่ค่าสหสัมพันธ์ที่เป็นค่าลบ หมายถึง เซตรายการทั้งสองแปรผกผันกัน ถ้าค่าสหสัมพันธ์เท่า -1 จะหมายถึงเซตรายการทั้งสองแปรผกผันกันอย่างสมบูรณ์ ค่าสหสัมพันธ์ที่เป็นค่าบวก หมายถึง เซต

รายการทั้งสองแปรตามกัน ถ้าค่าสหสัมพันธ์เท่า 1 จะหมายถึงเซตรายการทั้งสองแปรผกตามกันอย่างสมบูรณ์ และถ้าค่าสหสัมพันธ์เป็น 0 หมายถึงเซตรายการทั้งสองไม่มีความสัมพันธ์ต่อกันเลย (Sheikh et al., 2004)

2.3.8 ค่าอัตราส่วนออดส์ (Odds Ratio)

ค่าอัตราส่วนออดส์ คือค่าประเมินทางสถิติที่คำนวณหาความสัมพันธ์ระหว่างตัวแปร 2 ตัวที่แต่ละตัวเป็นตัวแปรจัดกลุ่มที่แบ่งออกเป็น 2 กลุ่ม แนวคิดของออดส์มาจากการเสี่ยงโชค (gambling) ตัวอย่างเช่น ออดส์ของม้าที่จะแข่งชนะเป็น 3 : 1 หมายถึงความน่าจะเป็นที่ม้าจะชนะ 3 ครั้งต่อการไม่ชนะ 1 ครั้ง เมื่อนำค่าทางสถิตินี้มาประยุกต์ใช้กับการประเมินกฎความสัมพันธ์จะได้ว่า ค่าอัตราส่วนออดส์ คืออัตราส่วนระหว่างออดส์ของเหตุการณ์ที่มีเซตรายการที่มาก่อนและเซตรายการที่ตามมาปรากฏหรือไม่ปรากฏทั้งคู่กับออดส์ของเหตุการณ์ที่มีเซตรายการที่มาก่อนปรากฏแต่เซตรายการที่ตามมาไม่ปรากฏหรือเหตุการณ์ที่ไม่มีเซตรายการที่มาก่อนปรากฏแต่มีเซตรายการที่ตามมาปรากฏ โดยค่าอัตราส่วนออดส์ มีสูตรคำนวณดังนี้ สมการในการคำนวณค่าคอนวิคชั่นแสดงได้ดังนี้

กำหนดให้ Odds(R) คือ ค่าอัตราส่วนออดส์ของกฎความสัมพันธ์ R

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$P(\overline{X})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่ไม่มีรายการ X ในฐานข้อมูล

P(X and Y) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และรายการ Y ในฐานข้อมูล

$P(\overline{X}$ and $\overline{Y})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่ไม่มีรายการ X และรายการ Y ในฐานข้อมูล

$P(X$ and $\overline{Y})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และไม่รายการ Y ในฐานข้อมูล

$$\text{Odds}(X \rightarrow Y) = \frac{(P(X \text{ and } Y)\ P(\overline{X} \text{ and } \overline{Y}))}{(P(X \text{ and } \overline{Y})\ P(\overline{X} \text{ and } Y))}$$

ค่าที่เป็นไปได้ของค่าอัตราส่วนออดส์นั้นอยู่ในพิสัย [0, +∞] ถ้าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์เป็นอิสระต่อกันแล้วค่าอัตราส่วนออดส์จะเท่ากับ 0 กฎความสัมพันธ์ที่มีความน่าสนใจมากจะมีค่าอัตราส่วนออดส์เท่ากับ +∞ (Sheikh et al., 2004)

2.4 ตัวแบบการประเมินความน่าสนใจของกฎความสัมพันธ์ใหม่

การค้นหากฎความสัมพันธ์นั้นจัดเป็นหนึ่งในงานที่สำคัญและได้รับความนิยมมากในการทำเหมืองข้อมูล ตัวแบบดั้งเดิมที่ใช้ในประเมินในการค้นหากฎความสัมพันธ์นั้นก็คือตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น ที่ได้กล่าวไปแล้วในข้างต้น ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นนี้จะใช้การประเมินค่าความเชื่อมั่นเป็นตัวชี้วัดระดับความน่าสนใจของกฎความสัมพันธ์ การประเมินความน่าสนใจของกฎความสัมพันธ์ด้วยวิธีแบบดั้งเดิมนี้มีข้อบกพร่องอยู่หลายประการซึ่งจะอธิบายข้อบกพร่องเหล่านี้อย่างละเอียดในหัวข้อ 2.4.1 ต่อจากนั้นจะกล่าวถึงตัวแบบการประเมินความน่าสนใจของกฎความสัมพันธ์ที่เสนอโดย Liu และคณะในปี 2008 (Liu et al., 2008) ในงานวิจัยนี้ผู้วิจัยจะเรียกตัวแบบใหม่ดังกล่าวนี้ว่า ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นใหม่ (Support-New Confidence Model) ของ Liu และคณะ (Liu et al., 2008)

2.4.1 ข้อบกพร่องของตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น

การทำเหมืองข้อมูลของกฎความสัมพันธ์โดยใช้ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น (Support-Confidence Model) นั้นค่าที่ใช้ในการประเมินความน่าสนใจของกฎความสัมพันธ์ $X \rightarrow Y$ ก็คือค่าความเชื่อมั่นของกฎความสัมพันธ์ (Conf($X \rightarrow Y$)) แต่ในบางครั้งการทำเหมืองข้อมูลของกฎความสัมพันธ์โดยตัวแบบนี้อาจจะทำให้ได้รับกฎความสัมพันธ์ที่ไม่มีความเกี่ยวข้องกันจริงและอาจนำไปสู่การกำหนดกฎความสัมพันธ์ที่ผิดหรือเป็นผลบวกลวง (False Positive) ได้ การนำตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่นไปใช้งานจึงมีข้อจำกัดดังแสดงในตัวอย่างต่อไปนี้

<u>ตัวอย่างที่ 1</u> กำหนดให้ตารางด้านล่างนี้เป็นตารางแสดงทรานแซคชั่นอย่างง่าย แต่ละแถวแทนรายการต่างๆในทรานแซคชั่น แต่ละหลักแทนทรานแซคชั่น 1 ทรานแซคชั่น จำนวนทรานแซคชั่นทั้งหมดที่แสดงในตารางคือ 8 ทรานแซคชั่นคือทรานแซคชั่น T1-T8 มีรายการทั้งหมด 3 รายการคือ รายการ X รายการ Y และรายการ Z หมายเลข 1 หมายถึงมีรายการของแถวนั้นปรากฏอยู่บนทรานแซคชั่นของหลักนั้น และหมายเลข 0 หมายถึงไม่มีรายการของแถวนั้นปรากฏอยู่บนทรานแซคชั่นของหลักนั้น (Tan et al., 2002; Liu et al., 2008)

ตารางที่ 2-1 แสดงตัวอย่างทรานแซคชั่นที่ประกอบด้วยรายการ X Y และ Z

	T1	T2	T3	T4	T5	T6	T7	T8
X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

พิจารณาตารางข้างต้นจะเห็นได้ว่ารายการ X และรายการ Y นั้นมีความสัมพันธ์ที่แปรผันตรงกันหรือมีสหสัมพันธ์เชิงบวกต่อกัน (Positively Correlated) เป็นส่วนใหญ่ กล่าวคือถ้ามีรายการ X ปรากฏอยู่บนทรานแซคชั่นก็จะมีรายการ Y ปรากฏบนทรานแซคชั่นเป็นส่วนใหญ่และในทางกลับกันถ้าไม่มีรายการ X ปรากฏอยู่บนทรานแซคชั่นก็จะไม่มีรายการ Y ปรากฏบนทรานแซคชั่นเป็นส่วนใหญ่เช่นกัน นอกจากนั้นยังสามารถสังเกตได้ว่ารายการ X และรายการ Z นั้นมีความสัมพันธ์ที่แปรผกผันกันหรือมีสหสัมพันธ์เชิงลบต่อกัน (Negatively Correlated) เป็นส่วนใหญ่ (ร้อยละ 62.5) ด้วย กล่าวคือถ้ามีรายการ X ปรากฏอยู่บนทรานแซคชั่นก็จะไม่มีรายการ Z ปรากฏบนทรานแซคชั่นเป็นส่วนใหญ่และในทางกลับกันถ้าไม่มีรายการ X ปรากฏอยู่บนทรานแซคชั่นก็จะมีรายการ Z ปรากฏบนทรานแซคชั่นเป็นส่วนใหญ่เช่นกัน แต่อย่างไรก็ตามเมื่อได้คำนวณหาค่าสนับสนุนและค่าความเชื่อมั่นของกฎความสัมพันธ์ $X \rightarrow Y$ แล้วได้ค่า 25% และ 50% ตามลำดับ และเมื่อคำนวณหาค่าสนับสนุนและค่าความเชื่อมั่นของกฎความสัมพันธ์ $X \rightarrow Z$ แล้วได้ค่า 37.5% และ 75% ตามลำดับ จากค่าสนับสนุนและค่าความเชื่อมั่นของกฎความสัมพันธ์ทั้งสองบ่งชี้ว่ากฎความสัมพันธ์ $X \rightarrow Z$ มีความน่าสนใจมากกว่ากฎความสัมพันธ์ $X \rightarrow Y$ ตัวอย่างนี้ได้แสดงให้เห็นจุดบกพร่องของการใช้ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น ในการค้นหากฎความสัมพันธ์คือ กฎความสัมพันธ์ที่มีค่าความเชื่อมั่นที่มีค่าสูงในตัวอย่างกลับเป็นกฎความสัมพันธ์ที่มีความสัมพันธ์แปรผกผันกันหรือมีความสัมพันธ์เชิงลบต่อกันสูง

<u>ตัวอย่างที่ 2</u> กำหนดให้ตารางด้านล่างนี้เป็นตารางแสดงการปรากฏของรายการชาและรายการกาแฟในทรานแซคชั่นการซื้อสินค้าในร้านขายของชำแห่งหนึ่ง แถว t และ t' คือร้อยละของทรานแซคชั่นที่มีรายการชาปรากฏอยู่และไม่มีรายการชาปรากฏอยู่ตามลำดับ หลัก c และ c' คือร้อยละของทรานแซคชั่นที่มีรายการกาแฟปรากฏอยู่และไม่มีรายการกาแฟปรากฏอยู่ตามลำดับ (Brin et al., 1997; Liu et al., 2008)

ตารางที่ 2-2 แสดงตัวอย่างร้อยละของการปรากฏของรายการบนทรานแซคชั่นของร้านขายของชำแห่งหนึ่ง

	c	c'	รวม
t	20	5	25
t'	70	5	75
รวม	90	10	100

จากข้อมูลในตารางข้างต้น เมื่อนำไปใช้ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น เพื่อค้นหากฎความสัมพันธ์จะได้ผลลัพธ์คือได้กฎความสัมพันธ์ $t \rightarrow c$ มีค่าสนับสนุนเท่ากับ 20% ซึ่งในทางปฏิบัติถือว่าเป็นค่าที่ค่อนข้างสูงมาก ค่าความเชื่อมั่นของกฎความสัมพันธ์ $t \rightarrow c$ นี้จะเป็นตัวบ่งชี้ถึงความน่าจะเป็นที่ลูกค้าจะซื้อกาแฟเมื่อลูกค้านั้นซื้อชา ซึ่งมีค่าเท่ากับความน่าจะเป็นที่จะซื้อชาและกาแฟหารด้วยความน่าจะเป็นที่จะซื้อชา นั่นคือ 20/25 = 0.8 หรือ 80% ซึ่งจัดได้ว่าเป็นค่าความเชื่อมั่นที่สูงมาก จากค่าสนับสนุนและค่าความเชื่อมั่นที่คำนวณไว้สามารถสรุปได้ว่ากฎความสัมพันธ์ $t \rightarrow c$ นี้เป็นกฎความสัมพันธ์ที่มีอยู่จริงอย่างสมเหตุสมผล

แต่ในความจริงแล้วข้อมูลที่แสดงในตารางไม่ใช่ข้อมูลทั้งหมดในฐานข้อมูลของร้านขายของชำแห่งนี้ ซึ่งอาจเป็นไปได้ว่าความสัมพันธ์ระหว่างการซื้อชาและการซื้อกาแฟในทรานแซคชั่นใดๆเป็นความสัมพันธ์แปรผกผันกัน จากตารางจะสามารถเห็นได้เพียงค่าร้อยละของการซื้อชาและซื้อกาแฟด้วยคือร้อยละ 20 ค่าที่กำหนดให้ในตารางไม่เพียงพอจะนำมาคำนวณค่าสหสัมพันธ์ได้ โดยตรงแต่ว่าสามารถทราบทิศของความสัมพันธ์ระหว่างการซื้อชาและการซื้อกาแฟได้โดยการใช้การคำนวณค่าลิฟท์ดังสูตร P(tUc)/(P(t)xP(c)) = 0.2/(0.25x0.9) = 0.89 ค่าลิฟท์ที่คำนวณมาได้นั้นมีค่าน้อยกว่า 1 ซึ่งบ่งชี้ว่าระหว่างการซื้อชาและการซื้อกาแฟมีความสัมพันธ์เชิงลบต่อกันหรือการซื้อชาและการซื้อกาแฟแปรผกผันกันนั่นเอง จากตัวอย่างนี้แสดงเห็นถึงข้อบกพร่องของการใช้ตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น ที่ทำให้ได้กฎความสัมพันธ์ที่ขัดแย้งต่อทิศของสหสัมพันธ์

2.4.2 ตัวแบบการประเมินความน่าสนใจของกฎความสัมพันธ์ใหม่

การประเมินความน่าสนใจของกฎความสัมพันธ์ด้วยวิธีการต่างๆ ที่กล่าวไปในข้างต้นนั้นแสดงให้เห็นอย่างชัดเจนว่าการใช้ตัวแบบประเมินที่แตกต่างกันส่งผลให้ได้กฎความสัมพันธ์ที่น่าสนใจต่างกันออกไป (Sheikh et al., 2004) เหตุผลสำคัญที่ทำให้เป็นเช่นนั้นก็เพราะความไม่สอดคล้องกันของความน่าจะเป็นของการมีกฎความสัมพันธ์ปรากฏกับค่าสหสัมพันธ์ของกฎ

ความสัมพันธ์ ในปี 2008 Liu และคณะ (Liu et al., 2008) ได้นำเสนอตัวแบบการประเมินความน่าสนใจของกฎความสัมพันธ์ขึ้นมาใหม่ โดยการพิสูจน์ทฤษฎีบทที่เกี่ยวข้องกับตัวแบบค่าสนับสนุน-ค่าความเชื่อมั่น กับทฤษฎีบทค่าสหสัมพันธ์ และตั้งชื่อการประเมินความน่าสนใจของกฎความสัมพันธ์แบบใหม่นี้ว่า ค่าความเชื่อมั่นใหม่ (New Confidence) โดยใช้สัญลักษณ์ NConf($X \rightarrow Y$) แทนค่าความเชื่อมั่นใหม่นี้ ซึ่งสามารถคำนวณได้จากสูตรดังต่อไปนี้

กำหนดให้ NConf(R) คือ ค่าความเชื่อมั่นใหม่ของกฎความสัมพันธ์ R

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$P(\overline{X})$ คือ ค่าความน่าจะเป็นในการไม่พบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

P(X and Y) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และรายการ Y ในฐานข้อมูล

$P(X \text{ and } \overline{Y})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และไม่รายการ Y ในฐานข้อมูล

P(X|Y) คือ ค่าความน่าจะเป็นในการพบรายการ X ในทรานแซคชั่นที่มีรายการ Y อยู่แล้ว

$$\text{NConf}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \overline{Y})}{P(\overline{Y})}$$

หรือ $$\text{NConf}(X \rightarrow Y) = P(X|Y) - P(X|\overline{Y})$$

ค่าที่เป็นไปได้ของค่าความเชื่อมั่นใหม่นั้นอยู่ในพิสัย [-1, 1] ถ้าค่าความเชื่อมั่นใหม่มีค่าเท่ากับ 0 หมายความว่าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์เป็นอิสระต่อกัน ถ้าค่าความเชื่อมั่นใหม่น้อยกว่า 0 แสดงว่าทั้งเซตรายการที่มาก่อนและเซตรายการที่ตามมาแปรผกผันกันหรือสหสัมพันธ์เชิงลบ และถ้าค่าความเชื่อมั่นใหม่มากกว่า 0 แสดงว่าทั้งเซตรายการที่มาก่อนและเซตรายการที่ตามมาแปรผันตามกันหรือสหสัมพันธ์เชิงบวก (Liu et al., 2008)

ในงานวิจัยของ Liu และคณะ (Liu et al., 2008) ที่ได้เสนอค่าความเชื่อมั่นใหม่ข้างต้นนั้น Liu และคณะได้ทำการเปรียบเทียบความสามารถของค่าความเชื่อมั่นใหม่กับค่าประเมินความน่าสนใจของกฎความสัมพันธ์อื่นๆ ทั้งหมด 8 ค่าคือ ค่าสนับสนุน (Support), ค่าความเชื่อมั่น (Confidence), ค่าคอนวิคชั่น (Conviction), ค่าลิฟท์ (Lift), ค่าเลฟเวอเรจ (Leverage), ค่าคัฟเวอเรจ (Coverage), ค่าสหสัมพันธ์ (Correlation) และ ค่าอัตราส่วนออดส์ (Odds Ratio) โดยใช้ฐานข้อมูลทรานแซคชั่นสมมุติขนาด 10 ทรานแซคชั่น ผลของการเปรียบเทียบคือ ค่าความเชื่อมั่นใหม่สามารถบ่งบอกทิศของความสัมพันธ์ได้อย่างถูกต้องและสอดคล้องกับค่าเลฟเวอเรจและค่าสหสัมพันธ์ แต่ค่าความเชื่อมั่นใหม่นั้นสามารถระบุความแตกต่างของความน่าสนใจของกฎความสัมพันธ์ 2 กฎความสัมพันธ์ใดๆที่ค่าเลฟเวอเรจและค่าสหสัมพันธ์ไม่สามารถระบุได้ (กล่าวคือกฎความสัมพันธ์ 2 กฎที่คำนวณค่าค่าเลฟเวอเรจหรือค่าสหสัมพันธ์ได้เท่ากันทั้ง 2 กฎ แต่ค่าความเชื่อมั่นใหม่ให้ค่าที่แตกต่างกันระหว่าง 2 กฎ) ส่วนค่าประเมินความน่าสนใจของกฎความสัมพันธ์อื่นๆให้ค่าที่ขัดแย้งกับค่าสหพันธ์ (กล่าวคือเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์นั้นมีความสัมพันธ์เชิงลบต่อกันแต่กลับให้ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ที่สูงออกมา)

เนื่องจากการเปรียบเทียบค่าความเชื่อมั่นใหม่กับค่าอื่นๆในงานวิจัยของ Liu และคณะ (Liu et al., 2008) เป็นการทดสอบกับฐานข้อมูลทรานแซคชั่นสมมุติขนาด 10 ทรานแซคชั่นเท่านั้น จึงค่อนข้างขาดความน่าเชื่อถือว่าค่าความเชื่อมั่นใหม่จะให้ประสิทธิภาพในการค้นหากฎความสัมพันธ์ที่ดีกว่าค่าอื่นๆจริง ผู้วิจัยจึงรวบรวมวรรณกรรมที่เกี่ยวข้องกับคุณสมบัติที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ควรจะมีในหัวข้อ 2.5 ทั้งหมด 16 คุณสมบัติ และทำการเปรียบเทียบคุณสมบัติที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ทั้ง 8 ค่ารวมถึงค่าความเชื่อมั่นใหม่มี

2.5 คุณสมบัติที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ควรจะมี

ในปีค.ศ. 1991 Piatetsky-Shapiro และคณะ (Piatetsky-Shapiro, 1991) ได้เสนอคุณสมบัติ 3 ประการสำคัญที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะมี ดังนี้

กำหนดให้ M คือ ค่าประเมินความน่าสนใจของกฎความสัมพันธ์

$X \rightarrow Y$ คือ กฎความสัมพันธ์ที่มีเซตรายการที่มาก่อนคือเซต X และเซตรายการที่ตามมาคือเซต Y

P(X) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

P(X and Y) คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และ Y ในฐานข้อมูล

1. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ต้องเท่ากับ 0 เมื่อเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์เป็นอิสระต่อกันทางสถิติ (M = 0 ถ้า P(X and Y) = P(X)P(Y))
2. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรต้องเพิ่มขึ้นเมื่อ P(X and Y) เพิ่มขึ้นในขณะที่ P(X) และ P(Y) คงที่
3. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรต้องเพิ่มขึ้นเมื่อ P(X) (หรือ P(Y)) ลดลงในขณะที่ P(X and Y) และ P(Y) (หรือ P(X)) คงที่

Piatetsky-Shapiro และคณะ (Piatetsky-Shapiro, 1991) เสนอคุณสมบัติของค่าประเมินความน่าสนใจของกฎความสัมพันธ์ทั้ง 3 คุณสมบัติขึ้นมาให้มีความเป็นทั่วไป (General) มากที่สุด โดยในทั้ง 3 คุณสมบัติจะอ้างอิงถึงพารามิเตอร์เพียง 3 ตัวคือ P(X) P(Y) และ P(X and Y) ซึ่งเป็นค่าทางสถิติที่เกี่ยวข้องกับกฎความสัมพันธ์เท่านั้น (Freitas, 1999) ในงานวิจัยนี้ผู้วิจัยจะใช้สัญลักษณ์ P1 P2 และ P3 แทนการอ้างถึงคุณสมบัติที่ 1 2 และ 3 ของ Piatetsky-Shapiro และคณะ (Piatetsky-Shapiro, 1991)

คุณสมบัติ P1 นั้นเป็นคุณสมบัติที่อธิบายว่าถ้าเซตรายการที่มาก่อนและเซตรายการที่ตามมานั้นเป็นอิสระต่อกันแล้วกฎความสัมพันธ์นั้นก็ควรจะต้องไม่มีความน่าสนใจเลยหรือมีค่าประเมินความน่าสนใจของกฎความสัมพันธ์นั้นเท่ากับ 0 คณะวิจัยหลายคณะวิจารณ์ว่าคุณสมบัติข้อนี้นั้นไม่ค่อยยืดหยุ่นและจำกัดมากเกินไป (Tan et al, 2002) ในปี ค.ศ. 2002 Tan และคณะ (Tan et al, 2002) จึงเสนอให้ผ่อนคลายคุณสมบัติข้อนี้ใหม่เป็น ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ต้องเท่ากับ k เมื่อเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์เป็นอิสระต่อกันทางสถิติ (M = k ถ้า P(X and Y) = P(X)P(Y)) โดยที่ k คือค่าคงที่ แต่อย่างไรก็ตามงานวิจัยต่างๆ (Freitas , 1999; Major and Mangano, 1995; Mcgarry, 2005; Geng and Hamilton, 2006; Liu et al., 2008; Heravi, 2009) ที่นำคุณสมบัติทั้ง 3 ข้อของ Piatetsky-Shapiro และคณะไปใช้หรืออ้างอิงถึงก็ยังคงอ้างอิงตามต้นฉบับคือค่าประเมินความน่าสนใจของกฎความสัมพันธ์ควรจะเท่ากับ 0 เมื่อกฎนั้นไม่มีความน่าสนใจเลย คุณสมบัติ P2 นั้น

เป็นคุณสมบัติที่อธิบายว่าในขณะที่ค่าสนับสนุนของเซตรายการที่มาก่อนและค่าสนับสนุนของเซตรายการที่ตามมาคงที่ ถ้าค่าสนับสนุนของทั้งกฎความสัมพันธ์เพิ่มขึ้น ความน่าสนใจของกฎความสัมพันธ์นั้นก็ควรจะเพิ่มขึ้นตามด้วย กล่าวคือถ้ากฎความสัมพันธ์มีความสัมพันธ์เชิงบวกกันมากขึ้น กฎความสัมพันธ์นั้นก็ควรจะน่าสนใจเพิ่มขึ้น และคุณสมบัติ P3 นั้นเป็นคุณสมบัติที่อธิบายว่าในขณะที่ค่าสนับสนุนของกฎความสัมพันธ์และค่าสนับสนุนของเซตรายการที่มาก่อน (หรือ ค่าสนับสนุนของเซตรายการที่ตามมา) คงที่ ถ้าค่าสนับสนุนของเซตรายการที่ตามมา (หรือค่าสนับสนุนของเซตรายการที่มาก่อน) ลดลงแล้ว ความน่าสนใจของกฎความสัมพันธ์นั้นควรจะเพิ่มขึ้น (Piatetsky-Shapiro, 1991; Geng and Hamilton, 2006)

ในปีค.ศ. 1995 Major และ Mangano (Major and Mangano, 1995) เสนอให้นำคุณสมบัติทั้ง 3 ข้อของ Piatetsky-Shapiro และคณะ (Piatetsky-Shapiro, 1991) พร้อมกับแนะนำให้เพิ่มอีก 1 คุณสมบัติเข้าไปดังนี้

กำหนดให้ M คือ ค่าประเมินความน่าสนใจของกฎความสัมพันธ์

$X \rightarrow Y$ คือ กฎความสัมพันธ์ที่มีเซตรายการที่มาก่อนคือเซต X และเซตรายการที่ตามมาคือเซต Y

$P(X)$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$P(\overline{X})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่ไม่มีรายการ X ในฐานข้อมูล

$Conf(X \rightarrow Y)$ คือ ค่าความเชื่อมั่นของกฎความสัมพันธ์ $X \rightarrow Y$

1. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรต้องเพิ่มขึ้นเมื่อ P(X) เพิ่มขึ้น ในขณะที่ P(Y) $P(\overline{Y})$ และ $Conf(X \rightarrow Y)$ คงที่

คุณสมบัติของนี้มักจะถูกอ้างถึงและนำไปใช้พร้อมๆกับคุณสมบัติทั้ง 3 ข้อของ Piatetsky-Shapiro และคณะ (Piatetsky-Shapiro, 1991) ดังนั้นคุณสมบัตินี้มักจะถูกเรียกว่าคุณสมบัติของที่ 4 ที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ควรจะมี ในงานวิจัยนี้ผู้วิจัยจะใช้สัญลักษณ์ P4 แทนการอ้างถึงคุณสมบัติข้อนี้ของ Major และ Mangano (Major and Mangano, 1995)

คุณสมบัติ P4 นั้นเป็นคุณสมบัติที่อธิบายว่าในขณะที่ค่าความเชื่อมั่นของกฎความสัมพันธ์มีค่าคงที่ เมื่อค่าสนับสนุนของของเซตรายการที่มาก่อนเพิ่มขึ้นแล้ว ค่าประเมินความ

น่าสนใจของกฎความสัมพันธ์ M ก็ควรจะเพิ่มขึ้นตามด้วย ในขณะที่ P(Y) P($\overline{Y}$) และ P(Y|X) (หรือ Conf(X→Y) นั่นเอง) คงที่

ในปีค.ศ. 2002 Tan และคณะ (Tan et al, 2002) เสนอคุณสมบัติอีก 5 ประการที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะมี เพิ่มเติมจาก 4 ข้อของคณะวิจัย Piatetsky-Shapiro (Piatetsky-Shapiro, 1991) และ Major กับ Mangano (Major and Mangano, 1995) โดยที่คุณสมบัติ 5 ประการของ Tan และคณะนั้นเป็นคุณสมบัติที่อยู่บนฐานของการดำเนินการบนตารางหลายตัวแปร (Contingency Table) ขนาด 2 x 2 ดังนี้

ตารางที่ 2-3 แสดงตารางหลายตัวแปร (Contingency Table) ขนาด 2 x 2 ของกฎความสัมพันธ์ X→Y

	Y	$\overline{Y}$	
X	n(X and Y)	n(X and $\overline{Y}$)	n(X)
$\overline{X}$	n($\overline{X}$ and Y)	n($\overline{X}$ and $\overline{Y}$)	n($\overline{X}$)
	n(Y)	n($\overline{Y}$)	N

กำหนดให้ M คือ ค่าประเมินความน่าสนใจของกฎความสัมพันธ์

X→Y คือ กฎความสัมพันธ์ที่มีเซตรายการที่มาก่อนคือเซต X และเซตรายการที่ตามมาคือเซต Y

N คือ จำนวนของทรานแซคชั่นทั้งหมดในฐานข้อมูล

n(X) คือ จำนวนของทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

n($\overline{X}$) คือ จำนวนของทรานแซคชั่นที่ไม่มีรายการ X ในฐานข้อมูล

n(X and Y) คือ จำนวนของทรานแซคชั่นที่มีรายการ X และ Y ในฐานข้อมูล

1. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรมีคุณสมบัติสมมาตรภายในการเปลี่ยนแปลงลำดับ
2. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรคงที่เมื่อมีการขยายตามแถวหรือขยายตามหลัก

3. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรสามารถบ่งชี้ทิศทางของความสัมพันธ์ได้
4. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรคงที่เมื่อมีการดำเนินการทั้งตามแถวและตามหลักพร้อมกัน
5. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M จะต้องไม่มีความสัมพันธ์กับจำนวนของทรานแซคชั่นที่มีเซตรายการ X เซตรายการ Y หรือทั้งคู่

ในงานวิจัยนี้ผู้วิจัยจะใช้สัญลักษณ์ O1 O2 O3 O4 และ O5 แทนการอ้างถึงคุณสมบัติที่ 1 2 3 4 และ 5 ของ Tan และคณะ (Tan et al, 2002)

คุณสมบัติ O1 นั้นเป็นคุณสมบัติที่อธิบายว่ากฎความสัมพันธ์ $X \rightarrow Y$ และกฎความสัมพันธ์ $Y \rightarrow X$ ควรมีค่าประเมินความน่าสนใจของกฎความสัมพันธ์ที่เท่ากัน Tan และคณะ (Tan et al, 2002) ผู้เสนอคุณสมบัติข้อนี้กล่าวว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ที่ไม่แสดงคุณสมบัติสมมาตรสามารถแก้ไขให้มีคุณสมบัติสมมาตรได้โดยการกำหนดให้ค่า M ของกฎความสัมพันธ์ $X \rightarrow Y$ และค่า M ของกฎความสัมพันธ์ $Y \rightarrow X$ มีค่าเท่ากับค่าใดค่าหนึ่งที่มากกว่าหรือเท่ากับ $\max(M(X \rightarrow Y), M(Y \rightarrow X))$ นั่นเอง คุณสมบัติข้อนี้ถูกปฏิเสธว่าไม่เป็นจริงในหลายงานวิจัย (Geng and Hamilton, 2006) คุณสมบัติ O2 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าการประเมินความน่าสนใจของกฎความสัมพันธ์ $X \rightarrow Y$ เมื่อตอนที่ n(X and Y) มีค่าเท่ากับ Z ควรจะเท่ากับค่าการประเมินความน่าสนใจของกฎความสัมพันธ์ $X \rightarrow Y$ เมื่อตอนที่ n(X and Y) มีค่าเท่ากับ k_1k_2Z โดยที่ k_1, k_2 เป็นค่าคงที่บวกที่มาขยายตามแถวและตามหลักตามลำดับ เป็นต้น คุณสมบัติ O3 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าการประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะต้องบ่งบอกได้ว่าเซตรายการที่มาก่อนและเซตรายการที่ตามมามีความสัมพันธ์เชิงบวกหรือความสัมพันธ์เชิงลบต่อกันได้ คุณสมบัติ O4 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าการประเมินความน่าสนใจของกฎความสัมพันธ์ $X \rightarrow Y$ ควรเท่ากับค่าการประเมินความน่าสนใจของกฎความสัมพันธ์ $\bar{X} \rightarrow \bar{Y}$ ด้วย และคุณสมบัติ O5 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M จะต้องไม่เปลี่ยนแปลงเมื่อ n($\bar{X}$ and $\bar{Y}$) เปลี่ยนแปลง กล่าวคือค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M จะต้องไม่มีความสัมพันธ์กับจำนวนของทรานแซคชั่นที่ไม่มีทั้งเซตรายการที่มาก่อนและเซตรายการที่ตาม (Tan et al, 2002)

ในปีค.ศ. 2004 Lenca และคณะ (Lenca et al, 2004) ก็เสนอคุณสมบัติ 5 ข้อที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะมีเช่นกัน หลังจากนั้นในปี ค.ศ. 2007 Lenca และคณะได้ปรับปรุงคุณสมบัติทั้ง 5 ข้อนั้นเล็กน้อยเพื่อให้มีความยืดหยุ่นมากขึ้น (Lenca et al, 2007) ดังนี้

กำหนดให้ M คือ ค่าประเมินความน่าสนใจของกฎความสัมพันธ์

$X \rightarrow Y$ คือ กฎความสัมพันธ์ที่มีเซตรายการที่มาก่อนคือเซต X และเซตรายการที่ตามมาคือเซต Y

$P(X)$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$P(\overline{X})$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่ไม่มีรายการ X ในฐานข้อมูล

$P(X \text{ and } Y)$ คือ ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และ Y ในฐานข้อมูล

N คือ จำนวนของทรานแซคชั่นทั้งหมดในฐานข้อมูล

1. ถ้า $P(X \text{ and } \overline{Y}) = 0$ แล้ว ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะเป็นค่าคงที่หรือเป็นอนันต์ (infinity)
2. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะลดลงแบบเส้นตรง แบบพาราโบล่าหงาย หรือแบบพาราโบล่าคว่ำ เมื่อ $P(X \text{ and } \overline{Y})$ มีค่าเพิ่มขึ้น
3. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะเพิ่มขึ้นเมื่อ N เพิ่มขึ้นในขณะที่ P(X and Y) P(X) และ P(Y) คงที่
4. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ที่ดีควรจะต้องสามารถหาค่าเธรสโชลด์ (threshold) ที่แบ่งแยกระหว่างกฎความสัมพันธ์ที่น่าสนใจออกจากกฎความสัมพันธ์ที่ไม่น่าสนใจได้ง่าย
5. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ที่ดีควรจะต้องมีอรรถศาสตร์ (semantics) ที่ผู้ใช้สามารถเข้าใจได้ง่าย

ในงานวิจัยนี้ผู้วิจัยจะใช้สัญลักษณ์ Q1 Q2 Q3 Q4 และ Q5 แทนการอ้างถึงคุณสมบัติที่ 1 2 3 4 และ 5 ของ Lenca และคณะ (Lenca et al, 2004; Lenca et al, 2007)

คุณสมบัติ Q1 นั้นเป็นคุณสมบัติที่อธิบายว่าเมื่อกฎความสัมพันธ์นั้นมีความน่าสนใจสูงที่สุด หรือค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y ในฐานข้อมูลเท่ากับ 0 (หรือ ค่าความเชื่อมั่นเท่ากับ 1 นั่นเอง) แล้วค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะเป็นค่าคงที่ค่าใดค่าหนึ่งหรือเป็นค่าอนันต์เพื่อสื่อความหมายอย่างชัดเจนว่ากฎความสัมพันธ์นั้นน่าสนใจสูงที่สุด คุณสมบัติ Q2 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะต้องมีการลงลดเมื่อมีทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y ถูกเพิ่มเข้ามาในฐานข้อมูล ลักษณะของการลดลงจะเป็นอย่างไรนั้นขึ้นอยู่กับวัตถุประสงค์ของงานที่นำไปประยุกต์ใช้ ตัวอย่างเช่น ถ้าต้องการให้ค่าประเมินความน่าสนใจของกฎความสัม์พันธ์ M มีความคงทน (Tolerated) ต่อการเพิ่มขึ้นของทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y (กล่าวคือเมื่อมีทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y เพิ่มขึ้น ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรลดลงทีละเพียงเล็กน้อยเท่านั้น) ก็ควรกำหนดให้ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ลดลงแบบพาราโบล่าหงาย (Concave up) เมื่อมีทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y เพิ่มขึ้น แต่ถ้าต้องการให้ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M มีความอ่อนไหวมาก (กล่าวคือถ้ามีทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y เพิ่มขึ้นมาเพียงเล็กน้อยก็จะทำให้ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ลดลงอย่างรวดเร็ว) ก็ควรกำหนดให้ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ลดลงแบบพาราโบล่าคว่ำ (Concave down, Convex) เมื่อมีทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y เพิ่มขึ้น เป็นต้น คุณสมบัติ Q3 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะเพิ่มขึ้นเมื่อจำนวนของทรานแซคชั่นทั้งหมดเพิ่มขึ้นในขณะที่ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และ Y ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ Y คงที่ คุณสมบัติ Q4 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะต้องสามารถหาค่าเธรสโชลด์ (Threshold) ที่เป็นจุดแบ่งแยกระหว่างกฎความสัมพันธ์ที่น่าสนใจกับกฎความสัมพันธ์ที่ไม่น่าสนใจได้ง่าย และค่าเธรสโชลด์นั้นจะต้องสื่อความหมายที่ผู้ใช้สามารถเข้าใจได้ง่ายด้วย กล่าวคือค่าเธรสโชลด์นั้นควรเป็นค่ากลางระหว่างค่าสูงสุดกับค่าต่ำสุดนั่นเอง คุณสมบัติ Q5 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ที่ดีควรจะต้องสื่อความหมายที่ผู้ใช้สามารถเข้าใจได้ง่าย กล่าวคือเมื่อผู้ใช้อ่านค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M แล้วจะต้องสามารถเข้าใจได้โดยง่ายว่าค่านี้หมายถึงอะไร โดยไม่ต้องอธิบายใดๆเพิ่มเติม (Lenca et al, 2004; Lenca et al, 2007)

ในปีค.ศ. 2006 Geng และ Hamilton (Geng and Hamilton, 2006) ก็เสนอคุณสมบัติ 2 ข้อที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะมีเช่นกัน แต่คุณสมบัติทั้ง 2 ข้อของ Geng และ Hamilton (Geng and Hamilton, 2006) นี้จะเน้นประเมินที่ความสัมพันธ์ระหว่างค่าประเมินความน่าสนใจของกฎความสัมพันธ์ ค่าสนับสนุน และค่าความเชื่อมั่น ดังนี้

กำหนดให้ M คือ ค่าประเมินความน่าสนใจของกฎความสัมพันธ์

$X \rightarrow Y$ คือ กฎความสัมพันธ์ที่มีเซตรายการที่มาก่อนคือเซต X และเซตรายการที่ตามมาคือเซต Y

$Sup(X \rightarrow Y)$ คือ ค่าสนับสนุนของกฎความสัมพันธ์ $X \rightarrow Y$

$Conf(X \rightarrow Y)$ คือ ค่าความเชื่อมั่นของกฎความสัมพันธ์ $X \rightarrow Y$

N คือ จำนวนของทรานแซคชั่นทั้งหมดในฐานข้อมูล

$n(X)$ คือ จำนวนของทรานแซคชั่นที่มีรายการ X ในฐานข้อมูล

$n(\overline{X})$ คือ จำนวนของทรานแซคชั่นที่ไม่มีรายการ X ในฐานข้อมูล

$n(X \text{ and } Y)$ คือ จำนวนของทรานแซคชั่นที่มีรายการ X และ Y ในฐานข้อมูล

1. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรอยู่ในรูปของฟังก์ชั่นที่ขึ้นกับค่าสนับสนุน ($f(Sup(X \rightarrow Y))$) โดยที่ค่าของฟังก์ชั่นนี้ควรเพิ่มขึ้นเมื่อค่าสนับสนุนเพิ่มขึ้น ในขณะที่ขอบ (Margins) ของตารางหลายตัวแปร (ตารางที่ 2-3) คงที่
2. ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรอยู่ในรูปของฟังก์ชั่นที่ขึ้นกับค่าความเชื่อมั่น ($f(Conf(X \rightarrow Y))$) โดยที่ค่าของฟังก์ชั่นนี้ควรเพิ่มขึ้นเมื่อค่าความเชื่อมั่นเพิ่มขึ้น ในขณะที่ขอบ (Margins) ของตารางหลายตัวแปร (ตารางที่ 2-3) คงที่

ในงานวิจัยนี้ผู้วิจัยจะใช้สัญลักษณ์ S1 และ S2 แทนการอ้างถึงคุณสมบัติที่ 1 และ 2 ของ Geng และ Hamilton (Geng and Hamilton, 2006)

สมมุติให้ $n(X) = a$, $n(\overline{X}) = N - a$, $n(Y) = b$, $n(\overline{Y}) = N - b$, $Sup(X \rightarrow Y) = x$ และ $Conf(X \rightarrow Y) = y$ คุณสมบัติ S1 นั้นเป็นคุณสมบัติที่อธิบายว่า เมื่อค่า $n(X)$ $n(\overline{X})$ $n(Y)$ และ $n(\overline{Y})$ คงที่จะได้ว่า $P(X \text{ and } Y) = x$, $P(\overline{X} \text{ and } Y) = (\frac{b}{N}) - x$, $P(X \text{ and } \overline{Y}) = (\frac{a}{N})) - x$ และ $P(\overline{X} \text{ and } \overline{Y}) = 1 - \frac{a+b}{N}) + x$ แล้วค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ที่เขียนให้อยู่ในรูปของฟังก์ชั่นที่ขึ้นกับ x นั้นควรจะเพิ่มขึ้นเมื่อ x เพิ่มขึ้น และคุณสมบัติ S2 ก็เช่นเดียวกับคุณสมบัติ S1

คือเป็นคุณสมบัติที่อธิบายว่า เมื่อค่า n(X) n($\overline{X}$) n(Y) และ n($\overline{Y}$) คงที่จะได้ว่า P(X and Y) = $\frac{ay}{N}$, P($\overline{X}$ and Y) = $\frac{(b-ay)}{N}$, P(X and $\overline{Y}$) = $\frac{a(1-y)}{N}$ และ P($\overline{X}$ and $\overline{Y}$) = $1 - \frac{(a+b)}{N} + \frac{ay}{N}$ แล้วค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ที่เขียนให้อยู่ในรูปของฟังก์ชั่นที่ขึ้นกับ y นั้นควรจะเพิ่มขึ้นเมื่อ y เพิ่มขึ้น (Geng and Hamilton, 2006)

คุณสมบัติ S1 สอดคล้องโดยตรงกับคุณสมบัติ P2 (Heravi, 2009) และคุณสมบัติ S2 สอดคล้องโดยตรงกับคุณสมบัติ Q2 (Geng and Hamilton, 2006)

คุณสมบัติที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ควรจะมีทั้งหมด 16 ข้อข้างต้น คุณสมบัติที่ได้รับการยอมรับและถูกอ้างอิงถึงโดยงานวิจัยต่างๆ (Freitas , 1999; Major and Mangano, 1995; Mcgarry, 2005; Geng and Hamilton, 2006; Liu et al., 2008; Heravi, 2009) มากที่สุดคือคุณสมบัติ P1 P2 และ P3 ของ Piatetsky-Shapiro และคณะ (Piatetsky-Shapiro, 1991) และคุณสมบัติ P4 ของ Mango และ Mangano (Major and Mangano, 1995)

จากหัวข้อที่ 2.3 ผู้วิจัยได้ทบทวนวรรณกรรมของค่าประเมินความน่าสนใจของกฎความสัมพันธ์ทั้งหมด 8 ค่าคือ ค่าสนับสนุน (Support), ค่าความเชื่อมั่น (Confidence), ค่าคอนวิคชั่น (Conviction), ค่าลิฟท์ (Lift), ค่าเลฟเวอเรจ (Leverage), ค่าคัฟเวอเรจ (Coverage), ค่าสหสัมพันธ์ (Correlation) และ ค่าอัตราส่วนออดส์ (Odds Ratio) และหัวข้อ 2.4 ที่ผู้วิจัยทบทวนวรรณกรรมของค่าประเมินความน่าสนใจของกฎความสัมพันธ์ใหม่ที่ชื่อว่า ค่าความเชื่อมั่นใหม่ (New Confidence) ที่ถูกเสนอขึ้นโดย Liu และคณะ (Liu et al., 2008) ผู้วิจัยจึงรวบรวมงานวิจัยที่ทำการทดสอบคุณสมบัติทั้ง 14 คุณสมบัติกับค่าประเมินความน่าสนใจของกฎความสัมพันธ์ทั้งหมด 8 ค่าและค่าความเชื่อมั่นใหม่ การเปรียบเทียบคุณสมบัติของค่าประเมินความน่าสนใจของกฎความสัมพันธ์ 8 ค่าที่รวบรวมโดยคณะวิจัยของ Geng กับ Hamilton ในปี 2006 และคณะวิจัยของ Heravi ในปี 2009 (Geng and Hamilton, 2006; Heravi, 2009) และค่าความเชื่อมั่นใหม่ที่รวบรวมโดยผู้วิจัยแสดงดังตารางที่ 2-4 ในบทที่ 1

เนื่องจากงานวิจัยของ Liu และคณะในปี 2008 (Liu et al., 2008) ได้ทำการพิสูจน์คุณสมบัติค่าความเชื่อมั่นใหม่ไว้ทั้งหมดเพียง 5 คุณสมบัติคือ คุณสมบัติ P1 P2 P3 O1 และ O2 เท่านั้น ดังนั้นผู้วิจัยจึงพิสูจน์คุณสมบัติ P4 O3 O4 O5 Q1 Q2 Q3 S1 และ S2 ของค่าความเชื่อมั่นใหม่ดังนี้

<u>พิสูจน์คุณสมบัติ P4</u>

คุณสมบัติ P4 นั้นเป็นคุณสมบัติที่อธิบายว่าในขณะที่ค่าความเชื่อมั่นของกฎความสัมพันธ์มีค่าคงที่ เมื่อค่าสนับสนุนของของเซตรายการที่มาก่อนเพิ่มขึ้นแล้ว ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ก็ควรจะเพิ่มขึ้นตามด้วย ในขณะที่ P(Y) P($\overline{Y}$) และ P(Y|X) (หรือ Conf(X→Y) นั่นเอง) คงที่

จาก $NConf(X \rightarrow Y) = P(X|Y) - P(X|\overline{Y})$

$$NConf(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \overline{Y})}{P(\overline{Y})}$$

$$NConf(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(\overline{Y}|X)P(X)}{P(\overline{Y})}$$

(เนื่องจาก $P(X \text{ and } \overline{Y}) = P(\overline{Y}|X)P(X)$)

$$NConf(X \rightarrow Y) = \frac{P(Y|X)P(X)}{P(Y)} - \frac{P(\overline{Y}|X)P(X)}{P(\overline{Y})}$$

(เนื่องจาก $P(X \text{ and } Y) = P(Y|X)P(X)$)

$$NConf(X \rightarrow Y) = P(X)\left(\frac{P(Y|X)}{P(Y)} - \frac{P(\overline{Y}|X)}{P(\overline{Y})}\right)$$

จากข้อกำหนด P(Y) P($\overline{Y}$) และ P(Y|X) คงที่

กรณีที่พจน์ $\left(\frac{P(Y|X)}{P(Y)} - \frac{P(\overline{Y}|X)}{P(\overline{Y})}\right)$ มีค่าเป็นบวก จะทำให้ $NConf(X \rightarrow Y)$ มีค่าเพิ่มขึ้นเมื่อ P(X) เพิ่มขึ้น

กรณีที่พจน์ $\left(\frac{P(Y|X)}{P(Y)} - \frac{P(\overline{Y}|X)}{P(\overline{Y})}\right)$ มีค่าเป็นลบ จะทำให้ $NConf(X \rightarrow Y)$ มีค่าลดลงเมื่อ P(X) ลดลง

ดังนั้น ค่าความเชื่อมั่นใหม่ไม่มีคุณสมบัติ P4

<u>พิสูจน์คุณสมบัติ O3</u>

คุณสมบัติ O3 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าการประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะต้องบ่งบอกได้ว่าเซตรายการที่มาก่อนและเซตรายการที่ตามมามีความสัมพันธ์เชิงบวกหรือความสัมพันธ์เชิงลบต่อกันได้

เนื่องจาก ค่าที่เป็นไปได้ของค่าความเชื่อมั่นใหม่นั้นอยู่ในพิสัย [-1, 1] ถ้าค่าความเชื่อมั่นใหม่มีค่าเท่ากับ 0 หมายความว่าเซตรายการที่มาก่อนและเซตรายการที่ตามมาของกฎความสัมพันธ์เป็นอิสระต่อกัน ถ้าค่าความเชื่อมั่นใหม่น้อยกว่า 0 แสดงว่าทั้งเซตรายการที่มาก่อนและเซตรายการที่ตามมาแปรผกผันกันหรือสหสัมพันธ์เชิงลบ และถ้าค่าความเชื่อมั่นใหม่มากกว่า 0 แสดงว่าทั้งเซตรายการที่มาก่อนและเซตรายการที่ตามมาแปรผันตามกันหรือสหสัมพันธ์เชิงบวก (Liu et al., 2008)

ดังนั้นค่าความเชื่อมั่นใหม่มีคุณสมบัติ O3

<u>พิสูจน์คุณสมบัติ O4</u>

คุณสมบัติ O4 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าการประเมินความน่าสนใจของกฎความสัมพันธ์ $X \rightarrow Y$ ควรเท่ากับค่าการประเมินความน่าสนใจของกฎความสัมพันธ์ $\overline{X} \rightarrow \overline{Y}$

จาก $\text{NConf}(X \rightarrow Y) = P(X|Y) - P(X|\overline{Y})$

$$\text{NConf}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \overline{Y})}{P(\overline{Y})}$$

และ $\text{NConf}(\overline{X} \rightarrow \overline{Y}) = P(\overline{X}|\overline{Y}) - P(\overline{X}|Y)$

$$\text{NConf}(\overline{X} \rightarrow \overline{Y}) = \frac{P(\overline{X} \text{ and } \overline{Y})}{P(\overline{Y})} - \frac{P(\overline{X} \text{ and } Y)}{P(Y)}$$

ต้องการพิสูจน์ว่า $\text{NConf}(X \rightarrow Y) = \text{NConf}(\overline{X} \rightarrow \overline{Y})$

$$\frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \overline{Y})}{P(\overline{Y})} = \frac{P(\overline{X} \text{ and } \overline{Y})}{P(\overline{Y})} - \frac{P(\overline{X} \text{ and } Y)}{P(Y)}$$

คูณ $P(Y)P(\overline{Y})$ ทั้ง 2 ข้าง

$$P(\overline{Y})P(X \text{ and } Y) - P(Y)P(X \text{ and } \overline{Y}) = P(Y)P(\overline{X} \text{ and } \overline{Y}) - P(\overline{Y})P(\overline{X} \text{ and } Y)$$

เนื่องจาก $P(X \text{ and } \overline{Y}) = P(X) - P(X \text{ and } Y)$

$$P(\overline{Y})P(X \text{ and } Y) - P(Y)(P(X) - P(X \text{ and } Y)) = P(Y)P(\overline{X} \text{ and } \overline{Y}) - P(\overline{Y})P(\overline{X} \text{ and } Y)$$

เนื่องจาก $P(X \text{ and } \overline{Y}) = P(Y) - P(X \text{ and } Y)$

$$P(\overline{Y})P(X \text{ and } Y) - P(Y)(P(X) - P(X \text{ and } Y)) = P(Y)P(\overline{X} \text{ and } \overline{Y}) - P(\overline{Y})(P(Y) - P(X \text{ and } Y))$$

เนื่องจาก $P(\overline{X} \text{ and } \overline{Y}) = 1 - P(X) - P(Y) + P(X \text{ and } Y)$

$$P(\overline{Y})P(X \text{ and } Y) - P(Y)(P(X) - P(X \text{ and } Y)) = P(Y)(1 - P(X) - P(Y) + P(X \text{ and } Y)) - P(\overline{Y})(P(Y) - P(X \text{ and } Y))$$

$$P(\overline{Y})P(X \text{ and } Y) - P(X)P(Y) + P(Y)P(X \text{ and } Y) = P(Y) - P(X)P(Y) - P(Y)P(Y) + P(Y)P(X \text{ and } Y) - P(\overline{Y})(P(Y) - P(X \text{ and } Y))$$

เนื่องจาก $P(\overline{Y}) = 1 - P(Y)$

$$(1 - P(Y))P(X \text{ and } Y) - P(X)P(Y) + P(Y)P(X \text{ and } Y) = P(Y) - P(X)P(Y) - P(Y)P(Y) + P(Y)P(X \text{ and } Y) - (1 - P(Y))(P(Y) - P(X \text{ and } Y))$$

$$P(X \text{ and } Y) - P(X)P(Y) = P(Y) - P(X)P(Y) - P(Y) + P(X \text{ and } Y)$$

$$1 = 1 \quad \text{เป็นจริง}$$

ดังนั้นค่าความเชื่อมั่นใหม่มีคุณสมบัติ O4

<u>พิสูจน์คุณสมบัติ O5</u>

คุณสมบัติ O5 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M จะต้องไม่เปลี่ยนแปลงเมื่อ $n(\overline{X} \text{ and } \overline{Y})$ เปลี่ยนแปลง กล่าวคือค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะคำนวณมาจาก n(X) n(Y) หรือ/และ n(X and Y) เท่านั้น (ไม่มี $n(\overline{X} \text{ and } \overline{Y})$ หรือ N มาเกี่ยวข้อง)

จาก $\text{NConf}(X \rightarrow Y) = P(X|Y) - P(X|\overline{Y})$

$$\text{NConf}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \overline{Y})}{P(\overline{Y})}$$

$$\text{NConf}(X \rightarrow Y) = \frac{n(\text{X and Y})\text{N}}{n(\text{Y})\text{N}} - \frac{n(\text{X and } \overline{\text{Y}})\text{N}}{n(\overline{\text{Y}})\text{N}}$$

$$\text{NConf}(X \rightarrow Y) = \frac{n(\text{X and Y})}{n(\text{Y})} - \frac{n(\text{X and } \overline{\text{Y}})}{n(\overline{\text{Y}})}$$

$$\text{NConf}(X \rightarrow Y) = \frac{n(\text{X and Y})}{n(\text{Y})} - \frac{(n(\text{X}) - n(\text{X and Y}))}{(\text{N} - n(\text{Y}))}$$

มีพจน์ที่เกี่ยวข้องกับจำนวนทรานแซคชั่นทั้งหมด (N) คือพจน์ $(\text{N} - n(\text{Y}))$

ดังนั้นค่าความเชื่อมั่นใหม่ไม่มีคุณสมบัติ O5

<u>พิสูจน์คุณสมบัติ Q1</u>

คุณสมบัติ Q1 นั้นเป็นคุณสมบัติที่อธิบายว่าเมื่อกฎความสัมพันธ์นั้นมีความน่าสนใจสูงที่สุด หรือค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y ในฐานข้อมูลเท่ากับ 0 (หรือ ค่าความเชื่อมั่นเท่ากับ 1 นั่นเอง) แล้วค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะเป็นค่าคงที่ค่าใดค่าหนึ่งหรือเป็นค่าอนันต์

จาก $\text{NConf}(X \rightarrow Y) = \text{P}(\text{X}|\text{Y}) - \text{P}(\text{X}|\overline{\text{Y}})$

$$\text{NConf}(X \rightarrow Y) = \frac{\text{P}(\text{X and Y})}{\text{P}(\text{Y})} - \frac{\text{P}(\text{X and } \overline{\text{Y}})}{\text{P}(\overline{\text{Y}})}$$

เมื่อกฎความสัมพันธ์นั้นมีความน่าสนใจสูงที่สุด นั่นคือ $\text{P}(\text{X and } \overline{\text{Y}}) = 0$

จะได้ $$\text{NConf}(X \rightarrow Y) = \frac{\text{P}(\text{X and Y})}{\text{P}(\text{Y})}$$

$$\text{NConf}(X \rightarrow Y) = \frac{n(\text{X and Y})\text{N}}{n(\text{Y})\text{N}}$$

$$\text{NConf}(X \rightarrow Y) = \frac{n(\text{X and Y})}{n(\text{Y})}$$

เมื่อกฎความสัมพันธ์นั้นมีความน่าสนใจสูงที่สุด พจน์ n(X and Y) จะเป็นจำนวนเต็มบวกเท่านั้น (ไม่เป็น 0) และพจน์ n(Y) คือจำนวนเต็มบวก (ไม่เป็น 0)

จะได้ เมื่อกฎความสัมพันธ์นั้นมีความน่าสนใจสูงที่สุด ค่าความเชื่อมั่นใหม่จะมีค่าเป็น 1 ซึ่งเป็นค่าคงที่บวกเสมอ

ดังนั้นค่าความเชื่อมั่นใหม่มีคุณสมบัติ Q1

พิสูจน์คุณสมบัติ Q2

คุณสมบัติ Q2 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะต้องมีการลงลดเมื่อมีทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y ถูกเพิ่มเข้ามาในฐานข้อมูล โดยที่ลักษณะของการลดลงนั้นสามารถเป็นได้ 3 แบบคือ ลดลงเป็นเส้นตรง ลดลงเป็นพาราโบล่าคว่ำ หรือลดลงเป็นพาราโบล่าหงาย อย่างใดอย่างหนึ่ง

จาก $\text{NConf}(X \rightarrow Y) = P(X|Y) - P(X|\bar{Y})$

$$\text{NConf}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \bar{Y})}{P(\bar{Y})}$$

$$\text{NConf}(X \rightarrow Y) = \frac{n(X \text{ and } Y)N}{n(Y)N} - \frac{n(X \text{ and } \bar{Y})N}{n(\bar{Y})N}$$

$$\text{NConf}(X \rightarrow Y) = \frac{n(X \text{ and } Y)}{n(Y)} - \frac{n(X \text{ and } \bar{Y})}{n(\bar{Y})}$$

จะเห็นว่า เมื่อไม่มีทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y นั่นคือ $n(X \text{ and } \bar{Y}) = 0$

จะได้ $$\text{NConf}(X \rightarrow Y) = \frac{n(X \text{ and } Y)}{n(Y)}$$

เมื่อเริ่มมีทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y เพิ่มขึ้น นั่นคือ $n(X \text{ and } \bar{Y}) > 0$

จะได้ $$\text{NConf}(X \rightarrow Y) = \frac{n(X \text{ and } Y)}{n(Y)} - \frac{n(X \text{ and } \bar{Y})}{n(\bar{Y})}$$

จากสมการเส้นตรง $y = mx+c$

จะได้ว่าเมื่อเริ่มมีทรานแซคชั่นที่มีรายการ X แต่ไม่มีรายการ Y เพิ่มขึ้น สมการนี้เป็นสมการเส้นตรง

โดยที่ y คือ $\text{NConf}(X \rightarrow Y)$

x คือ $n(X \text{ and } \bar{Y})$

m คือ $-\frac{1}{n(\bar{Y})}$

c คือ $\frac{n(X \text{ and } Y)}{n(Y)}$

จะได้สมการเส้นตรงที่มีความชันเป็นลบ นั่นคือ $\text{NConf}(X \rightarrow Y)$ ลดลง เมื่อมี $n(X \text{ and } \bar{Y})$ เพิ่มขึ้น

ดังนั้นค่าความเชื่อมั่นใหม่มีคุณสมบัติ Q2 และมีลักษณะของการลดลงแบบเส้นตรง (ใส่หมายเลข 1 ในตารางที่ 2-4)

พิสูจน์คุณสมบัติ Q3

คุณสมบัติ Q3 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ควรจะเพิ่มขึ้นเมื่อจำนวนของทรานแซคชั่นทั้งหมด (N) เพิ่มขึ้นในขณะที่ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และ Y ค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ X และค่าความน่าจะเป็นในการพบทรานแซคชั่นที่มีรายการ Y คงที่

จาก $\text{NConf}(X \rightarrow Y) = P(X|Y) - P(X|\overline{Y})$

$$\text{NConf}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \overline{Y})}{P(\overline{Y})}$$

เนื่องจาก $P(X \text{ and } \overline{Y}) = P(X) - P(X \text{ and } Y)$

และ $P(\overline{Y}) = 1 - P(Y)$

จะได้ $$\text{NConf}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{(P(X) - P(X \text{ and } Y))}{(1 - P(Y))}$$

จะเห็นว่าเมื่อ N เพิ่มขึ้นในขณะที่ P(X and Y) P(X) และ P(Y) คงที่ ค่า $\text{NConf}(X \rightarrow Y)$ ก็คงที่

ดังนั้นค่าความเชื่อมั่นใหม่มีคุณสมบัติ Q3

พิสูจน์คุณสมบัติ S1

คุณสมบัติ S1 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ที่เขียนให้อยู่ในรูปของฟังก์ชั่นที่ขึ้นกับค่าสนับสนุนนั้นควรจะเพิ่มขึ้นเมื่อค่าสนับสนุนเพิ่มขึ้น เมื่อค่า $n(X)$ $n(\overline{X})$ $n(Y)$ และ $n(\overline{Y})$ คงที่

จาก $\text{NConf}(X \rightarrow Y) = P(X|Y) - P(X|\overline{Y})$

$$\text{NConf}(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \overline{Y})}{P(\overline{Y})}$$

สมมุติให้ $n(X) = a$, $n(\overline{X}) = N - a$, $n(Y) = b$, $n(\overline{Y}) = N - b$, $\text{Sup}(X \rightarrow Y) = x$

จะได้ว่า P(X and Y) = x, P($\overline{X}$ and Y) = ($\frac{b}{N}$) – x, P(X and $\overline{Y}$) = ($\frac{a}{N}$)) – x และ P($\overline{X}$ and $\overline{Y}$) = 1 – $\frac{a+b}{N}$) + x

ดังนั้น $$NConf(X \rightarrow Y) = \frac{xN}{b} - \frac{a - xN}{N - b}$$

$$NConf(X \rightarrow Y) = \frac{xN}{b} - \frac{a}{N - b} + \frac{xN}{N - b}$$

พจน์ $\frac{a}{N - b}$ คงที่ เนื่องจาก n(X) และ n($\overline{Y}$) คงที่

และเมื่อ x เพิ่ม พจน์ $\frac{xN}{b}$ และพจน์ $\frac{xN}{N - b}$ จะเพิ่มขึ้นด้วย เนื่องจาก n(Y) และ n($\overline{Y}$) คงที่

ดังนั้นเมื่อ x เพิ่ม ค่า NConf($X \rightarrow Y$) ก็จะเพิ่มด้วย

นั่นคือ Sup(X→Y) เพิ่ม ค่า NConf($X \rightarrow Y$) ก็จะเพิ่มด้วย

ดังนั้นค่าความเชื่อมั่นใหม่มีคุณสมบัติ S1 (ใส่หมายเลข 0 ในตารางที่ 2-4)

<u>พิสูจน์คุณสมบัติ S2</u>

คุณสมบัติ S1 นั้นเป็นคุณสมบัติที่อธิบายว่าค่าประเมินความน่าสนใจของกฎความสัมพันธ์ M ที่เขียนให้อยู่ในรูปของฟังก์ชั่นที่ขึ้นกับค่าความเชื่อมั่นนั้นควรจะเพิ่มขึ้นเมื่อค่าความเชื่อมั่นเพิ่มขึ้น เมื่อค่า n(X) n($\overline{X}$) n(Y) และ n($\overline{Y}$) คงที่

จาก $$NConf(X \rightarrow Y) = P(X|Y) - P(X|\overline{Y})$$

$$NConf(X \rightarrow Y) = \frac{P(X \text{ and } Y)}{P(Y)} - \frac{P(X \text{ and } \overline{Y})}{P(\overline{Y})}$$

สมมุติให้ n(X) = a, n($\overline{X}$) = N – a, n(Y) = b, n($\overline{Y}$) = N – b, Conf(X→Y) = y

จะได้ว่า P(X and Y) = $\frac{ay}{N}$, P($\overline{X}$ and Y) = $\frac{(b-ay)}{N}$, P(X and $\overline{Y}$) = $\frac{a(1-y)}{N}$ และ P($\overline{X}$ and $\overline{Y}$) = $1 - \frac{(a+b)}{N} + \frac{ay}{N}$

ดังนั้น $$NConf(X \rightarrow Y) = \frac{\frac{ay}{N}}{\frac{b}{N}} - \frac{\frac{a(1-y)}{N}}{\frac{N-b}{N}}$$

$$NConf(X \rightarrow Y) = \frac{ay}{b} - \frac{a(1-y)}{N-b}$$

$$NConf(X \rightarrow Y) = \frac{ay}{b} - \frac{a}{N-b} + \frac{ay}{N-b}$$

พจน์ $\frac{a}{N-b}$ คงที่ เนื่องจาก n(X) และ $n(\overline{Y})$ คงที่

และเมื่อ y เพิ่ม พจน์ $\frac{ay}{b}$ และพจน์ $\frac{ay}{N-b}$ จะเพิ่มขึ้นด้วย เนื่องจาก n(Y) และ $n(\overline{Y})$ คงที่

ดังนั้นเมื่อ y เพิ่ม ค่า $NConf(X \rightarrow Y)$ ก็จะเพิ่มด้วย

นั่นคือ Conf(X→Y) เพิ่ม ค่า $NConf(X \rightarrow Y)$ ก็จะเพิ่มด้วย

ดังนั้นค่าความเชื่อมั่นใหม่มีคุณสมบัติ S2 (ใส่หมายเลข 0 ในตารางที่ 2-4)

การทดสอบคุณสมบัติข้างต้นไม่รวมคุณสมบัติ Q4 และ Q5 เนื่องจากคุณสมบัติทั้ง 2 นั้นเป็นคุณสมบัติเชิงอัตวิสัย (Subjective Properties) หรือคุณสมบัติที่ขึ้นอยู่กับผู้ใช้และโดเมนที่นำไปใช้ ไม่สามารถตัดสินได้ว่ามีคุณสมบัติทั้ง 2 ข้อหรือไม่ถ้าไม่ได้อ้างอิงถึงโดเมนและวัตถุประสงค์ที่นำไปประยุกต์ใช้ (Geng and Hamilton, 2006; Heravi, 2009) กับค่าประเมินความน่าสนใจของกฎความสัมพันธ์ทั้งหมด 8 ค่าและค่าความเชื่อมั่นใหม่ การเปรียบเทียบคุณสมบัติของค่าประเมินความน่าสนใจของกฎความสัมพันธ์ 8 ค่าที่รวบรวมโดยคณะวิจัยของ Geng กับ Hamilton ในปี 2006 และคณะวิจัยของ Heravi ในปี 2009 (Geng and Hamilton, 2006; Heravi, 2009) และค่าความเชื่อมั่นใหม่ที่รวบรวมโดยผู้วิจัย แสดงดังตารางต่อไปนี้

ตารางที่ 2-4 แสดงการเปรียบเทียบคุณสมบัติของค่าประเมินความน่าสนใจของกฎความสัมพันธ์ทั้งหมด

	P1	P2	P3	P4	O1	O2	O3	O4	O5	Q1	Q2	Q3	S1	S2	รวม
ค่าสนับสนุน	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗	1	✗	0	0	6
ค่าความเชื่อมั่น	✗	✓	1	✗	✗	✗	✗	✗	✓	✓	1	✗	0	0	7
ค่าคอนวิคชั่น	✗	✓	2	✗	✗	✗	✗	✓	✗	✓	0	✗	0	0	7
ค่าลิฟท์	✗	✓	2	✗	✓	✗	✗	✗	✗	✗	2	✗	0	0	6
ค่าเลฟเวอเรจ	✗	✓	2	✗	✗	✗	✗	✗	✗	✗	1	✗	0	0	5
ค่าคัฟเวอเรจ	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	3	✗	1	0	3
ค่าสหสัมพันธ์	✓	✓	2	✗	✓	✗	✓	✓	✗	✗	0	✗	0	0	9
ค่าอัตราส่วนออดส์	✗	✓	2	✗	✓	✓	✗	✓	✗	✓	0	✗	4	0	9
ค่าความเชื่อมั่นใหม่	✓	✓	1	✗	✗	✓	✓	✓	✗	✓	1	✗	0	0	10

จากตารางข้างต้นเครื่องหมาย ✓ ในตารางหมายถึงค่าประเมินความน่าสนใจของกฎความสัมพันธ์ที่อยู่ในแถวนั้นมีคุณสมบัติของหลักนั้น เครื่องหมาย ✗ ในตารางหมายถึงค่าประเมินความน่าสนใจของกฎความสัมพันธ์ที่อยู่ในแถวนั้นไม่มีคุณสมบัติของหลักนั้น ในหลักของคุณสมบัติ P3 หมายเลข 0 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์เพิ่มขึ้นเมื่อค่าความน่าจะเป็นของเซตรายการที่มาก่อนลดลง หมายเลข 1 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์เพิ่มขึ้นเมื่อค่าความน่าจะเป็นของเซตรายการที่ตามมาลดลง หมายเลข 2 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์เพิ่มขึ้นเมื่อทั้งค่าความน่าจะเป็นของเซตรายการที่มาก่อนและค่าความน่าจะเป็นของเซตรายการที่ตามมาลดลงเท่านั้น ในหลักของคุณสมบัติ Q2 หมายเลข 0 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ลดลงแบบพาราโบล่าคว่ำ หมายเลข 1 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ลดลงแบบเส้นตรง หมายเลข 2 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ลดลงแบบพาราโบล่าหงาย หมายเลข 3 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ลดลงแต่ขึ้นอยู่กับพารามิเตอร์ หมายเลข 4 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์คงที่ หมายเลข 5 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์เพิ่มขึ้น หมายเลข 6 หมายถึง ค่าประเมินความน่าสนใจของกฎความสัมพันธ์เพิ่มหรือลดไม่แน่นอนขึ้นอยู่กับพารามิเตอร์ ในหลัก

ของคุณสมบัติ S1 และ S2 หมายเลข 0 หมายถึงค่าประเมินความน่าสนใจของกฎความสัมพันธ์เพิ่มขึ้นเมื่อค่าสนับสนุนเพิ่มขึ้น หมายเลข 1 หมายถึงค่าประเมินความน่าสนใจของกฎความสัมพันธ์คงที่เมื่อค่าสนับสนุนเพิ่มขึ้น หมายเลข 2 หมายถึงค่าประเมินความน่าสนใจของกฎความสัมพันธ์ลดลงเมื่อค่าสนับสนุนเพิ่มขึ้น หมายเลข 3 หมายถึงไม่สามารถประเมินได้ (not applicable) และหมายเลข 4 หมายถึงค่าประเมินความน่าสนใจของกฎความสัมพันธ์เพิ่มหรือลดขึ้นอยู่กับพารามิเตอร์ (Geng and Hamilton, 2006; Heravi, 2009) หลักสุดท้ายของตารางคือหลักที่แสดงการรวมจำนวนคุณสมบัติทั้งหมดที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์นั้นๆ มี

จากผลการพิสูจน์คุณสมบัติของค่าประเมินความน่าสนใจของกฎความสัมพันธ์ข้างต้นและตารางที่ 2-4 แสดงให้เห็นว่าค่าความเชื่อมั่นใหม่มีคุณสมบัติที่ค่าประเมินความน่าสนใจของกฎความสัมพันธ์ควรจะมี 10 คุณสมบัติจากทั้งหมด 14 คุณสมบัติ ซึ่งมากที่สุดเมื่อเทียบกับค่าประเมินความน่าสนใจของกฎความสัมพันธ์อื่นๆ โดยเฉพาะอย่างยิ่งการมีคุณสมบัติ O3 ของค่าความเชื่อมั่นใหม่จะทำให้การนำค่าความเชื่อมั่นใหม่ไปใช้นั้นจะสามารถขจัดการเกิดกฎความสัมพันธ์ที่มีเซตรายการที่มาก่อนและเซตรายการที่ตามที่มีความสัมพันธ์เชิงลบออกไปได้ ผู้วิจัยจึงเชื่อว่าถ้านำค่าความเชื่อมั่นใหม่นี้ไปประยุกต์ใช้กับการค้นหากฎความสัมพันธ์กับข้อมูลประเภทต่างๆรวมถึงข้อมูลซอฟต์แวร์อาร์ไควฟ์ แล้วน่าจะทำให้กฎความสัมพันธ์ที่ได้มาเป็นกฎความสัมพันธ์ที่น่าสนใจและช่วยลดการเกิดกฎความสัมพันธ์ที่เป็นผลบวกลวง (False Positive) ได้

2.6 การควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control, Version Control)

ซอฟต์แวร์ที่ถูกพัฒนาขึ้นในปัจจุบันนั้นมีความซับซ้อนและขนาดที่ใหญ่กว่าซอฟต์แวร์ที่ถูกพัฒนาขึ้นมาในอดีตเป็นอย่างมาก ความยากและความซับซ้อนเหล่านั้นถูกสะท้อนออกมาในการบริหารจัดการการพัฒนาและการบำรุงรักษาของซอฟต์แวร์นั้น แม้ว่าหลายองค์กรในปัจจุบันจะใช้ซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control Software, Version Control Software) คอยติดตามและจัดการกับพัฒนาการของความซับซ้อนของโครงการพัฒนาซอฟต์แวร์กันอย่างมาก แต่ทว่าแนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไข (Concept of Revision Control, Version Control) นั้นกลับเป็นศาสตร์ที่ได้ไม่ค่อยถูกพูดถึงและไม่ค่อยมีวัฒนาการมาก

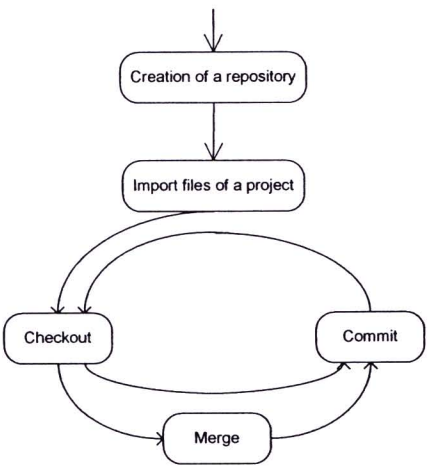
เท่าไรในตลอดทศวรรษที่ผ่านมา (Löh et al., 2007) ในหัวข้อนี้ได้เรียบเรียงวรรณกรรมที่เกี่ยวข้องกับการควบคุมการเปลี่ยนแปลงแก้ไข เริ่มตั้งแต่แนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไข ซอฟต์แวร์ที่ประยุกต์แนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไข และสุดท้ายจะกล่าวถึงระบบคอนเคอเรนท์เวอร์ชั่น (Concurrent Version System, CVS) ซึ่งเป็นซอฟต์แวร์โอเพนซอร์สที่ประยุกต์แนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไขที่ได้รับความนิยมสูงมากนานกว่าทศวรรษ (O'Sullivan et al., 2009)

2.6.1 แนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไข (Concept of Revision Control, Version Control)

การควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control, Version Control) เป็นลักษณะอย่างหนึ่งของการควบคุมเอกสาร (Documentation Control) แนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไข คือ การควบคุมพัฒนาการของเนื้อหาภายในเอกสารอิเล็กทรอนิกส์ตลอดช่วงอายุของเอกสาร รวมถึงการเรียกเนื้อหาในอดีตกลับคืน (Recovery) การบ่งชี้ข้อแตกต่างระหว่างเวอร์ชั่นของเนื้อหา และการให้รายละเอียดของพัฒนาการแต่ละครั้งด้วย (Tichy, 1982; Junqueira et al., 2008) การควบคุมการเปลี่ยนแปลงแก้ไขนั้นถูกนำไปประยุกต์ใช้ในทางวิศวกรรมแขนงต่างๆเพื่อการจัดการการพัฒนาที่ต่อเนื่องไปของเอกสารอิเล็กทรอนิกส์ เช่น ซอร์สโค้ดของโปรแกรมประยุกต์ พิมพ์เขียว แบบจำลองอิเล็กทรอนิกส์ และสารสนเทศสำคัญอื่นๆ ซึ่งพัฒนาโดยทีม การเปลี่ยนแปลงเอกสารเหล่านี้ในแต่ละครั้งจะถูกระบุโดยใช้การเพิ่มหมายเลขการเปลี่ยนแปลงแก้ไข (Revision Number) และมีการเชื่อมโยงกับผู้กระทำการเปลี่ยนแปลงแก้ไขด้วย

การไหลของกิจกรรมการควบคุมการเปลี่ยนแปลงแก้ไขเริ่มต้นจากการสร้างรีพอสิโทริ (Creation of a Pepository) สำหรับโครงการขึ้นมา จากนั้นนำเข้าแฟ้มข้อมูลทั้งหมดของโครงการ (Import Files of Project) ลงสูรีพอสิโทริที่สร้างไว้ แฟ้มข้อมูลที่นำเข้ามาครั้งแรกจะถูกกำหนดค่าเริ่มต้น (Initial Set) ให้มีหมายเลขการแก้ไขหรือหมายเลขเวอร์ชั่นเป็น 1 เมื่อมีความต้องการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลผู้ใช้ก็จะต้องทำการลงทะเบียนออก (Check out) แฟ้มข้อมูลนั้นออกมาจากรีพอสิโทริ การลงทะเบียนออกของแฟ้มข้อมูลก็คือการสำเนาแฟ้มข้อมูลเวอร์ชั่นล่าสุดจากรีพอสิโทริมาเป็นแฟ้มข้อมูลบนเครื่องของผู้ใช้ (Local File) เมื่อผู้ใช้เปลี่ยนแปลงแก้ไขแฟ้มข้อมูลเรียบร้อยแล้วจะต้องการลงทะเบียนเข้า (Check in) หรือคอมมิท (Commit) แฟ้มข้อมูลนั้นกลับเข้าสูรีพอสิโทริ การคอมมิทแต่ละครั้งจะมีผลให้หมายเลขการแก้ไขหรือหมายเลขเวอร์ชั่นมีค่าเพิ่มขึ้นไปเรื่อยๆ การไหลของกิจกรรมจะวนลูปเช่นนี้ไปเรื่อยๆตลอดช่วงอายุของโครงการ ใน

กรณีที่มีผู้ใช้หลายคนทำการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลเดียวกันในเวลาเดียวกัน (มีการลงทะเบียนออกของแฟ้มข้อมูลโดยผู้ใช้มากกว่า 1 คนในช่วงเวลาเดียวกัน) จะทำให้เกิดกิจกรรมที่สำคัญขึ้นอีกกิจกรรมคือการผสานเนื้อหา (Merge) การผสานเนื้อหาคือการปรับปรุง (Update) เนื้อหาภายในแฟ้มข้อมูลบนเครื่องผู้ใช้ที่แก้ไขไปแล้วกับเนื้อหาล่าสุดของแฟ้มข้อมูลนั้นในรีพอสิโทริซึ่งอาจถูกคอมมิทเวอร์ชั่นใหม่จากผู้ใช้อื่นในระหว่างที่ผู้ใช้กำลังแก้ไขอยู่ กล่าวคือเป็นการปรับปรุงเนื้อหาของแฟ้มข้อมูลนั้นบนเครื่องของผู้ใช้ (Local File) ให้สอดคล้องกับเวอร์ชั่นบนรีพอสิโทรินั่นเอง ผู้ใช้สามารถทำการผสานเนื้อหาได้โดยการเรียกใช้คำสั่งที่ชื่อว่า update (Tichy, 1982; Ambriola et al., 1990; Junqueira et al., 2008) การไหลของกิจกรรมการควบคุมการเปลี่ยนแปลงแก้ไขสามารถแสดงได้ดังรูปต่อไปนี้ (Junqueira et al., 2008)



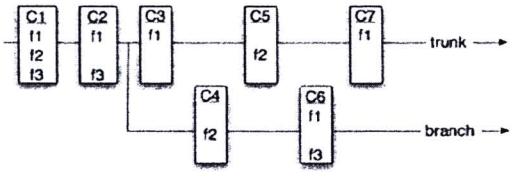
รูปที่ 2-1 แสดงการไหลของกิจกรรมการควบคุมการเปลี่ยนแปลงแก้ไข

แนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไขที่กล่าวไปข้างต้นนั้นเป็นแนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไขสำหรับเอกสารอิเล็กทรอนิกส์ทั่วไป สำหรับแนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไขในวิศวกรรมซอฟต์แวร์จะหมายถึงเทคนิคและเครื่องมือที่นำมาใช้ในการควบคุมพัฒนาการของแฟ้มข้อมูลซอร์สโค้ดและแฟ้มข้อมูลอื่นๆภายในโครงการพัฒนาซอฟต์แวร์ (Junqueira et al., 2008) แฟ้มข้อมูลซอร์สโค้ดนั้นเป็นแฟ้มข้อมูลที่มีคุณลักษณะเฉพาะตัวมากกว่าแฟ้มข้อมูลของเอกสารอิเล็กทรอนิกส์อื่นๆ ทำให้เทคนิคที่ใช้ในการควบคุมการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลซอร์ส โค้ดมีความละเอียดและเฉพาะตัวมากกว่าการควบคุมการเปลี่ยนแปลงแก้ไขของเอกสารอิเล็กทรอนิกส์อื่นๆ คุณลักษณะเฉพาะนั้นก็คือเนื้อหา (Contents) ภายในแฟ้มข้อมูลซอร์สโค้ดนั้นเป็นเนื้อหาที่มีโครงสร้างและโครงสร้างนั้นประกอบด้วยหน่วยย่อยที่

มีความสัมพันธ์กันอยู่ภายใน ด้วยเหตุนี้การควบคุมการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลซอร์สโค้ดจึงต้องมีการดำเนินการขั้นสูง (Advanced Operations) 2 อย่างคือการต่อกิ่ง (Branches) และการผสานกิ่ง (Merge) (Chadd et al., 2008)

คำว่า กิ่ง (Branch) ในที่นี้หมายถึง กลุ่มของเวอร์ชั่นต่างๆที่ถูกสร้างขึ้นมาด้วยเหตุผลในการทดสอบอะไรบางอย่างที่ยังไม่มั่นใจเพียงพอที่จะนำเวอร์ชั่นเหล่านี้ไปรวมกับเวอร์ชั่นหลักที่เสถียรแล้ว การต่อกิ่งเป็นการเปรียบเทียบการพัฒนาซอร์สโค้ดในลักษณะของต้นไม้ (Tree) คือการกำหนดให้ซอร์สโค้ดหลัก (เวอร์ชั่นของซอร์สโค้ดที่มั่นใจว่ามีความเสถียรสูงแล้ว) เป็นลำต้น (Trunk) ของต้นไม้ โดยปกติแล้วเมื่อนักพัฒนาแก้ไขหรือพัฒนาซอร์สโค้ดเพิ่มเติมเข้าไปจะทำให้เกิดเวอร์ชั่นใหม่ขึ้นมา เวอร์ชั่นใหม่ที่เสถียรเหล่านี้จะเรียงต่อกันไปเป็นลำดับ แต่ในกรณีที่นักพัฒนามีการแก้ไขหรือพัฒนาซอร์สโค้ดส่วนใดส่วนหนึ่ง ซึ่งเป็นการแก้ไขเพื่อทดสอบอะไรบางอย่างและยังไม่มั่นใจในความเสถียรของการแก้ไขครั้งนั้น เวอร์ชั่นใหม่ที่ถูกสร้างขึ้นมานี้จะถูกกำหนดให้เป็นกิ่ง (Branches) ต่อออกมาจากส่วนลำต้นหลักของต้นไม้ ทิศทางการพัฒนาจะขยายออกจากส่วนลำต้นของต้นไม้ การต่อกิ่งนี้เป็นเทคนิคที่นิยมใช้มากในการควบคุมการเปลี่ยนแปลงแก้ไขของโครงการพัฒนาซอฟต์แวร์ที่มีขนาดใหญ่ เทคนิควิธีการต่อกิ่งสามารถใช้เพื่อการทดสอบซอร์สโค้ดและเพิ่มเติมซอร์สโค้ดเมื่อเราต้องการที่จะขยายระบบหรือซอฟต์แวร์เพิ่มเติม ในขณะที่มีการพัฒนาส่วนกิ่งอยู่นั้นส่วนลำต้นก็ยังคงสามารถพัฒนาต่อออกไปได้เช่นกัน เมื่อพัฒนาส่วนที่เป็นกิ่งเสร็จแล้วและมั่นใจความเสถียรของกิ่งนั้นแล้ว นักพัฒนาสามารถที่จะนำส่วนกิ่งนั้นมารวมเข้ากับส่วนลำต้นหลักได้ เรียกว่าการผสานกิ่ง (Merge)

ถ้ากำหนดให้การเปลี่ยนแปลงแก้ไข (Change, Revision) คือการที่นักพัฒนาแก้ไข (alter) เพิ่ม (add) หรือ ลบ (delete) อย่างใดอย่างหนึ่งบนแฟ้มข้อมูลซอร์สโค้ด การคอมมิท (Commit) ก็คือเซตของการเปลี่ยนแปลงแก้ไขที่กระทำโดยนักพัฒนาคนเดียวกันในเวลาเดียวกันหรือใกล้เคียงกันโดยที่ไม่ใช่เซตว่างนั่นเอง รูปด้านล่างนี้แสดงตัวอย่างของการคอมมิท ลงบนลำต้นและบนกิ่งโดยกำหนดให้ สัญลักษณ์ C# คือการคอมมิทครั้งที่ # และสัญลักษณ์ f# คือการเปลี่ยนแปลงแก้ไขบนแฟ้มข้อมูลที่ # (Junqueira et al., 2008)



รูปที่ 2-2 แสดงตัวอย่างการคอมมิทลงบนลำต้นและบนกิ่ง

จากรูปที่ 2-2 ข้างต้นนี้ถ้ามีการผสานกิ่งเกิดขึ้นหลังจากการคอมมิทครั้งที่ 7 การดำเนินการของรีพอสิโทรีที่จะเกิดขึ้นก็คือ พิจารณาว่าตั้งแต่เกิดการต่อกิ่งนี้ขึ้นมาจนถึงการผสานกิ่งมีการเปลี่ยนแปลงแก้ไขกับแฟ้มข้อมูลใดบ้าง ถ้าการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลใดเกิดขึ้นทั้งบนกิ่งและบนลำต้น การเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลนั้นจะต้องถูกจับคู่กันแล้วนำไปวิเคราะห์เชิงลึกว่ามีการเปลี่ยนแปลงแก้ไขที่เกิดความขัดแย้ง (Conflict) ขึ้นหรือไม่ ถ้าไม่เกิดความขัดแย้งขึ้นเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลนั้นทั้งบนกิ่งและบนลำต้นก็จะสามารถนำมาผสานกันได้ทันที แต่ถ้าเกิดความขัดแย้งขึ้นจำเป็นจะต้องให้ผู้ที่ทำการผสานกิ่งเป็นผู้ตัดสินว่าจะแก้ไขความขัดแย้งนี้อย่างไร ตัวอย่างเช่น จากรูปข้างต้น ตั้งแต่จุดเริ่มต้นการต่อกิ่งจนถึงจุดการผสานกิ่งนั้นมีการคอมมิทบนลำต้นทั้งหมด 3 ครั้งคือ การคอมมิทครั้งที่ 3 การคอมมิทครั้งที่ 5 และการคอมมิทครั้งที่ 7 และมีการคอมมิทบนกิ่งทั้งหมด 2 ครั้งคือ การคอมมิทครั้งที่ 4 และการคอมมิทครั้งที่ 6 การเปลี่ยนแปลงแก้ไขบนแฟ้มข้อมูลที่ 2 เกิดขึ้นทั้งการคอมมิทบนกิ่ง (การคอมมิทครั้งที่ 4) และการคอมมิทบนลำต้น (การคอมมิทครั้งที่ 5) ดังนั้นการเปลี่ยนแปลงแก้ไขบนแฟ้มข้อมูลที่ 2 ทั้งสองอันนี้จะต้องถูกนำไปวิเคราะห์เชิงลึกต่อไป ถ้าพบว่าเกิดความขัดแย้งขึ้นจะต้องมีการตัดสินความขัดแย้งนั้น แต่ถ้าไม่มีการเปลี่ยนแปลงแก้ไขบนแฟ้มข้อมูลที่ 2 ทั้งสองอันจะถูกผสานเข้าด้วยกัน (Junqueira et al., 2008)

เทคนิคและขั้นตอนวิธีในการต่อกิ่ง การผสานกิ่ง และการเปรียบเทียบความแตกต่างระดับบรรทัดของแต่ละซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไขนั้น มีเทคนิคและขั้นตอนวิธีที่แตกต่างกันออกไปตามซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไข ซึ่งจะไม่กล่าวถึงในที่นี้

2.6.2 ซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control Software, Version Control Software)

ซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control Software, Version Control Software) คือ ซอฟต์แวร์ที่นำแนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไขมาประยุกต์ใช้จริงในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไขคือซอฟต์แวร์ที่ใช้ใน

จัดการการจัดเก็บ การค้นคืน การระบุและการผสานการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลซอร์สโค้ดของโปรแกรมประยุกต์ และสารสนเทศสำคัญอื่นๆที่พัฒนาขึ้นมาโดยทีมอย่างเป็นอัตโนมัติ ในซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไขนั้นจะมีการบันทึกแฟ้มข้อมูลซอร์สโค้ดทั้งโครงการเอาไว้ นอกจากนั้นข้อมูลที่เกี่ยวข้องกับการเปลี่ยนแปลงแก้ไขอื่นๆ อาทิเช่น ซอร์สโค้ดส่วนใดที่ถูกแก้ไข นักพัฒนาผู้บันทึกเวอร์ชั่นใหม่ของซอร์สโค้ด วันเวลาบันทึกเวอร์ชั่นใหม่ของซอร์สโค้ด และหมายเหตุของการบันทึกเวอร์ชั่นใหม่ของซอร์สโค้ดนั้นจะถูกบันทึกอยู่ในรูปของแฟ้มข้อมูลบันทึก (Log Files) แฟ้มข้อมูลบันทึกเหล่านี้ถูกแก้ไขใหม่ทุกครั้งที่มีนักพัฒนาทำการคอมมิท แฟ้มข้อมูลซอร์สโค้ดแต่ละเวอร์ชั่นในอดีตและแฟ้มข้อมูลบันทึกทั้งหมดจะถูกเรียกรวมกันว่า ซอฟต์แวร์อาร์ไคฟว์ (Software Archives) (Zimmermann et al., 2004; Zimmermann et al., 2005)

ซอฟต์แวร์แรกที่ได้ประยุกต์แนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไขขึ้นมาจริง คือ ระบบควบคุมซอร์สโค้ด (SCCS: Source Code Control System) ซึ่งถูกแนะนำขึ้นมาและออกจำหน่ายครั้งแรกในปี ค.ศ. 1972 โดยแบลแลบส์ (Bell Labs) (Rochkind et al., 1975) ในช่วงแรกนั้นระบบควบคุมซอร์สโค้ดถูกนำไปใช้อยู่เพียงแค่ในวงจำกัดและไม่ค่อยได้รับความนิยมมากเท่าไร (Baudis, 2009) หลังจากนั้นในปี ค.ศ. 1985 เป็นปีแรกที่ระบบควบคุมการเปลี่ยนแปลงแก้ไข (RCS: Revision Control System) ได้ถูกเผยแพร่ออกมาและสามารถนำไปใช้ได้โดยไม่เสียค่าใช้จ่าย (Tichy, 1985) หลังจากที่ถูกเผยแพร่ได้ไม่นานระบบควบคุมการเปลี่ยนแปลงแก้ไขได้ประสบความสำเร็จอย่างมากในอุตสาหกรรมพัฒนาซอฟต์แวร์ (Baudis, 2009) ถึงแม้ว่าในปัจจุบันระบบควบคุมการเปลี่ยนแปลงแก้ไขจะไม่ได้มีวิวัฒนาการใดๆเพิ่มขึ้นมาและไม่ได้ถูกนำไปใช้แล้วแต่ระบบควบคุมการเปลี่ยนแปลงแก้ไขยังคงได้รับการกล่าวถึงอยู่เรื่อยๆ ในฐานะเป็นต้นแบบของแนวคิดและรูปแบบแฟ้มข้อมูล (File Format) ของซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไขที่ดี (Baudis, 2009) ต่อมาในปี ค.ศ. 1990 ระบบคอนเคอเรนท์เวอร์ชั่น (CVS: Concurrent Versions System) ที่ถูกพัฒนาขึ้นมาโดยมีระบบควบคุมการเปลี่ยนแปลงแก้ไข (RCS, Revision Control System) เป็นต้นแบบได้ถูกเผยแพร่ออกมาและได้รับความนิยมอย่างมากและรวดเร็ว นอกจากซอฟต์แวร์ทั้ง 3 ซอฟต์แวร์ในข้างต้นแล้วหลังจากนั้นในช่วงทศวรรษที่ 1990 ก็มีซอฟต์แวร์ที่ประยุกต์แนวคิดของการควบคุมการเปลี่ยนแปลงแก้ไขเกิดขึ้นมาอีกอย่างมากมาย ทั้งที่เป็นซอฟต์แวร์เชิงพาณิชย์ (Commercial Software) และซอฟต์แวร์เสรี (Free/Open Source Software) ตัวอย่างซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไขเชิงพาณิชย์เช่น ClearCase (ปี ค.ศ. 1992), Visual SourceSafe (ปี ค.ศ. 1994), Perforce (ปี ค.ศ. 1995) และ Code Co-op

(ปี ค.ศ. 1997) ตัวอย่างซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไขที่เป็นซอฟต์แวร์เสรีเช่น CVSNT (ปี ค.ศ. 1998) และ Subversion (ปี ค.ศ. 2000)

เนื่องจากงานวิจัยนี้สนใจศึกษาข้อมูลซอฟต์แวร์อาร์ไคฟ์ที่ได้จากระบบคอนเคอเรนท์เวอร์ชั่นดังนั้นผู้วิจัยจึงเรียบเรียงรายละเอียดเกี่ยวกับระบบคอนเคอเรนท์เวอร์ชั่นรวมถึงข้อมูลซอฟต์แวร์อาร์ไคฟ์ของระบบคอนเคอเรนท์เวอร์ชั่นไว้ในหัวข้อถัดไป

2.6.3 ระบบคอนเคอเรนท์เวอร์ชั่น (Concurrent Versions System, CVS)

ระบบคอนเคอเรนท์เวอร์ชั่น (Concurrent Versions System, CVS) คือซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control Software, Version Control Software) ซอฟต์แวร์หนึ่งที่ได้รับความนิยมสูงสุดในปัจจุบัน (O'Sullivan et al., 2009) ระบบคอนเคอเรนท์เวอร์ชั่นนั้นถูกพัฒนาขึ้นมาโดยมีระบบควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control System, RCS) เป็นต้นแบบ และพัฒนาขึ้นมาอย่างโอเพนซอร์ส ดังนั้นระบบคอนเคอเรนท์เวอร์ชั่นนี้จึงสามารถบรรจุลง (download) มาใช้ได้โดยไม่เสียค่าใช้จ่าย ความสำคัญของระบบคอนเคอเรนท์เวอร์ชั่นคือระบบคอนเคอเรนท์เวอร์ชั่นนี้ถูกสร้างขึ้นมาให้เป็นส่วนประกอบ (component) ที่สำคัญส่วนประกอบหนึ่งที่ช่วยให้บรรลุถึงมาตรฐานของการจัดการค่าองค์ประกอบของซอฟต์แวร์ (SCM, Software Configuration Management) ซึ่งเป็น 1 ใน 6 กลุ่มกระบวนการหลัก (Process Area) ในระดับที่ 2 ของมาตรฐาน CMM (Capability Maturity Model) ได้ (Grune et al., 2006)

วิวัฒนาการของระบบคอนเคอเรนท์เวอร์ชั่นนั้นเริ่มต้นขึ้นในเดือนกรกฎาคมปี ค.ศ. 1986 Grune และคณะได้เริ่มต้นการเผยแพร่บางส่วนระบบคอนเคอเรนท์เวอร์ชั่น (Concurrent Versions System, CVS) ในรูปแบบของเชลล์สคริพท์ (Shell Script) ลงสู่กลุ่มข่าว (News-Group) ที่ชื่อว่า comp.sources.unix ต่อมาในเดือนเมษายนปี ค.ศ. 1989 Berliner และ Polk (Berliner et al., 1989) ได้ออกแบบและพัฒนาระบบคอนเคอเรนท์เวอร์ชั่นเพิ่มเติมซึ่งเวอร์ชั่นของ Berliner นี้ก็คือเวอร์ชั่นที่ใช้กันอยู่ในปัจจุบัน (Cederqvist, 2006; Grune et al., 2006) จนกระทั่งวันที่ 19 พฤศจิกายน ปี ค.ศ. 1990 ระบบคอนเคอเรนท์เวอร์ชั่นที่สมบูรณ์พร้อมใช้งานเวอร์ชั่นที่ 1 ได้ถูกเสนอเข้าสู่มูลนิธิซอฟต์แวร์เสรี (Free Software Foundation) เพื่อการพัฒนาและการเผยแพร่ต่อไป (Grune et al., 2006)

ระบบคอนเคอเรนท์เวอร์ชั่นเป็นซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไขที่ใช้สถาปัตยกรรมลูกข่าย-แม่ข่าย (Client-Server) การทำงานหลักของระบบคอนเคอเรนท์เวอร์ชั่น

คือ การเก็บประวัติการทำงานของแฟ้มข้อมูลต่างๆในโครงการพัฒนาซอฟต์แวร์ เพื่อช่วยแก้ปัญหาในการค้นหาความแตกต่างของซอร์สโค้ดพื้นฐานเดิมกับซอร์สโค้ดที่ทำการแก้ไขใหม่ นักพัฒนาสามารถที่จะทำการเปลี่ยนแปลงซอร์สโค้ด แก้ไขข้อผิดพลาดและรวม (Merge) ซอร์สโค้ดที่ตนพัฒนาอยู่กับซอร์สโค้ดล่าสุดของนักพัฒนาคนอื่นๆได้ ระบบคอนเคอเรนท์เวอร์ชั่นนั้นมีรูปแบบการทำงานแบบ 1 รีพอสิโทริ (Repository) แต่หลายๆ พื้นที่ทำงาน (Workspace) ภายใต้รูปแบบการทำงานนี้นักพัฒนาแต่ละคนสามารถนำซอร์สโค้ดมาพัฒนาเพียงงานเดียวหรือหลายๆงานก็ได้ เมื่อแก้ไขเรียบร้อยแล้วก็ทำการรวม (Merge) งานที่ทำเข้าด้วยกัน และสามารถช่วยให้นักพัฒนาค้นหาได้ว่าข้อผิดพลาดเกิดขึ้นนั้น เกิดขึ้นที่ไหนและกระทำโดยใคร โดยปกติแล้วการบันทึกเวอร์ชั่นทุกๆเวอร์ชั่นของแต่ละเอกสารเอาไว้นั้นเป็นการกระทำที่ค่อนข้างสิ้นเปลืองเนื้อที่บันทึก (Disk) เป็นอย่างมาก แต่ระบบคอนเคอเรนท์เวอร์ชั่นนั้นมีวิธีการบันทึกแต่ละเวอร์ชั่นของเอกสารภายในแฟ้มข้อมูลแฟ้มเดียวด้วยวิธีที่มีประสิทธิภาพ นั่นคือระบบคอนเคอเรนท์เวอร์ชั่นจะบันทึกเฉพาะส่วนที่แตกต่างกันของแต่ละเวอร์ชั่นเอาไว้เท่านั้น (Cederqvist, 2006)

เนื่องจากระบบคอนเคอเรนท์เวอร์ชั่น (CVS) เป็นระบบที่ถูกพัฒนาขึ้นมาโดยมีระบบควบคุมการเปลี่ยนแปลงแก้ไข (Revision Control System, RCS) เป็นต้นแบบ รูปแบบแฟ้มข้อมูล (File Format) ของระบบคอนเคอเรนท์เวอร์ชั่นจึงใช้รูปแบบเดียวกับระบบควบคุมการเปลี่ยนแปลงแก้ไขทั้งหมดรวมถึงรูปแบบของแฟ้มข้อมูลบันทึก (CVS Log File) ด้วย การดึงแฟ้มข้อมูลบันทึกของโครงการที่เก็บอยู่บนรีพอสิโทริสามารถทำได้โดยการใช้คำสั่ง cvs log และแฟ้มข้อมูลบันทึกจะถูกทำสำเนาและส่งกลับมายังเครื่องที่ใช้คำสั่งนี้ (Fischer et al., 2003) ตัวอย่างบางส่วนของแฟ้มข้อมูลบันทึกของคอนเคอเรนท์เวอร์ชั่นแสดงดังรูปด้านล่างนี้ (Fischer et al., 2003)

```
RCS file: /cvsroot/mozilla/layout/html/style/src/nsCSSFrameConstructor.cpp,v
Working file: nsCSSFrameConstructor.cpp
head: 1.804
branch:
locks: strict
access list:
symbolic names:
        MOZILLA_1_3a_RELEASE: 1.800
        NETSCAPE_7_01_RTM_RELEASE: 1.727.2.17
        PHOENIX_0_5_RELEASE: 1.800
        ...
        RDF_19990305_BASE: 1.46
        RDF_19990305_BRANCH: 1.46.0.2
keyword substitution: kv
total revisions: 976;   selected revisions: 976
description:
----------------------------
revision 1.804
date: 2002/12/13 20:13:16;  author: doe@netscape.com;  state: Exp;  lines: +15 -47
Don't set NS_BLOCK_SPACE_MGR and NS_BLOCK_WRAP_SIZE on ...
----------------------------
...
----------------------------
revision 1.638
date: 2001/09/29 02:20:52;  author: doe@netscape.com;  state: Exp;  lines: +14 -4
branches:  1.638.4;
bug 94341 keep a separate pseudo frame list for a new pseudo block or inline frame ...
----------------------------
....
=============================================================================
```

รูปที่ 2-3 แสดงตัวอย่างแฟ้มข้อมูลบันทึกของคอนเคอเรนท์เวอร์ชั่น

แฟ้มข้อมูลบันทึกของระบบคอนเคอเรนท์เวอร์ชั่นประกอบด้วยหลายส่วน (Sections) โดยที่แต่ละส่วนจะอธิบายถึงเวอร์ชั่นในอดีตของแต่ละแฟ้มข้อมูลที่อยู่ในโครงการ (1 ส่วนต่อ 1 แฟ้มข้อมูล) ส่วนแต่ละส่วนจะถูกแบ่งออกจากกันด้วยบรรทัดของเครื่องหมายเท่ากับ (=) ภายในแต่ละส่วนจะประกอบด้วยแอททริบิวต์ (Attributes) และค่าของแอททริบิวต์นั้นๆ แอททริบิวต์ที่สำคัญและถูกนำไปใช้ในการวิเคราะห์ต่างๆ มีรายละเอียดดังนี้ (Fischer et al., 2003)

- แอททริบิวต์ RCS file คือ พาธ (Path) ที่ระบุตำแหน่งของแฟ้มข้อมูลที่สัมพันธ์กับส่วนนี้ที่บันทึกอยู่บนรีพอสิโทริของระบบคอนเคอเรนท์เวอร์ชั่น โดยอ้างอิงจากตำแหน่งราก (Root) ของรีพอสิโทริ จากรูปที่ 2-3 ระบุว่าส่วนนี้เป็นส่วนของแฟ้มข้อมูลชื่อ nsCSSFrameConstructor.cpp ซึ่งมีพาธอยู่ที่ /cvsroot/mozilla/layout/html/style/src/nsCSSFrameConstructor.cpp
- แอททริบิวต์ symbolic names คือ รายการของคู่ชื่อแท็ก (Tag Name) กับหมายเลขการเปลี่ยนแปลงแก้ไข ที่นักพัฒนาตั้งชื่อไว้เพื่อสะดวกในการอ้างอิงถึง จากรูปที่ 2-3 ระบุว่าแฟ้มข้อมูลชื่อ nsCSSFrameConstructor.cpp ถูกตั้งชื่อแท็กไว้หลายชื่อ ตัวอย่างเช่นแท็กชื่อ MOZILLA_1_3a_RELEASE คู่กับหมายเลขการเปลี่ยนแปลงแก้ไขที่ 1.800

- แอททริบิวต์ description คือ รายการของการเปลี่ยนแปลงแก้ไขที่เกิดขึ้นกับแฟ้มข้อมูลนี้ เริ่มตั้งแต่ที่แฟ้มข้อมูลนี้ได้ถูกลงคอมมิทเข้าสูรีพอสิโทริจนถึงเวอร์ชั่นล่าสุดของแฟ้มข้อมูลนี้ นอกจากการเปลี่ยนแปลงแก้ไขที่ถูกบันทึกลงบนลำต้นหลัก (Main Trunk) แล้วทุกๆการเปลี่ยนแปลงแก้ไขที่ถูกบันทึกลงบนกิ่ง (Branches) ก็ถูกบันทึกไว้ในส่วนนี้ด้วยเช่นกัน การเปลี่ยนแปลงแก้ไขแต่ละการเปลี่ยนแปลงแก้ไขจะถูกแบ่งออกจากกันด้วยบรรทัดเครื่องหมายยัติภังค์ (-) ภายในแต่ละส่วนย่อยนี้อธิบายการเปลี่ยนแปลงแก้ไขทั้งหมดที่เคยเกิดขึ้นกับแฟ้มข้อมูล ประกอบด้วย 1) หมายเลขการเปลี่ยนแปลงแก้ไข (Revision Number) 2) วันและเวลา (Date) ที่คอมมิทการเปลี่ยนแปลงแก้ไขนี้เข้ามา 3) นักพัฒนา (Authof) ที่เป็นผู้คอมมิท 4) สเตท (Atate) ระบุถึงสถานภาพในขณะนั้นของแฟ้มข้อมูล มีค่าที่เป็นไปได้ 2 ค่า คือ Exp หมายถึง สถานภาพทดสอบ (Experimental State) และ Dead หมายถึง แฟ้มข้อมูลถูกลบไปแล้ว 4) บรรทัด (Lines) ระบุจำนวนบรรทัดที่ถูกเพิ่มเข้าไป (นำหน้าด้วยเครื่องหมายบวก) และจำนวนบรรทัดที่ถูกลบออกไป (นำหน้าด้วยเครื่องหมายลบ) ของแฟ้มข้อมูลสำหรับการคอมมิทครั้งนี้ 5) รายการหมายเลขกิ่ง (Branches) คือรายการของหมายเลขการเปลี่ยนแปลงแก้ไขบนกิ่งที่มีการเปลี่ยนแปลงแก้ไขนี้เป็นจุดต่อกิ่ง และ 6) บรรทัดสุดท้ายของส่วนการเปลี่ยนแปลงแก้ไขระบุข้อความหมายเหตุ (Comment) ที่ผู้พัฒนาเขียนระบุไว้ในการคอมมิท ตัวอย่างรายการการเปลี่ยนแปลงแก้ไขหนึ่งจากรูปที่ 2-3 คือรายการการเปลี่ยนแปลงแก้ไขที่มีหมายเลขการเปลี่ยนแปลงแก้ไขที่ 1.804 บันทึกวันที่ 13 ธันวาคม ค.ศ. 2002 เวลา 20 นาฬิการ 13 นาที 16 วินาที บันทึกโดย doe@netscape.com อยู่ในสถานภาพทดสอบ การเปลี่ยนแปลงแก้ไขครั้งนี้มีบรรทัดที่ถูกเพิ่มเข้าไปจำนวน 15 บรรทัด บรรทัดที่ถูกลบออกไป 47 บรรทัด และมีข้อความหมายเหตุจากผู้บันทึกว่า Don't set NS_BLOCK_SPACE_MGR and NS_BLOCK_WARP_SIZE on ...

2.7 การประยุกต์ใช้การทำเหมืองข้อมูลด้วยเทคนิคค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟ์ (Applying Association Rule Discovery in Software Archive)

ในช่วงต้นของการคิดค้นและพัฒนาแนวคิดการทำเหมืองข้อมูลด้วยด้วยเทคนิคการค้นหากฎความสัมพันธ์ (Association Rule Mining) นั้น การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ถูกพัฒนามาเพื่อการค้นหารูปแบบความสัมพันธ์ของพฤติกรรมการซื้อสินค้าของลูกค้า ตัวอย่างเช่นการค้นหาว่าลูกค้าที่ซื้อหนังสือ ก. มักจะซื้อหนังสืออะไรด้วย จากฐานข้อมูล

รายการซื้อหนังสือขนาดใหญ่ นอกจากความสามารถในการค้นหารูปแบบความสัมพันธ์ของพฤติกรรมการซื้อสินค้าของลูกค้าแล้ว ต่อจากนั้นมาไม่นานนักเริ่มมีนักวิจัยหลายคณะนำการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์มาประยุกต์ใช้กับข้อมูลในแขนงต่างๆ เช่น ข้อมูลเครือข่ายโทรคมนาคม ข้อมูลการจัดการความเสี่ยง ข้อมูลการควบคุมคลังสินค้า และข้อมูลทางพันธุกรรมของสิ่งมีชีวิต เป็นต้น (Kotsiantis et al., 2006) นอกจากนั้นการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ยังสามารถประยุกต์ใช้กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Software Archive) ของขั้นตอนการพัฒนาซอฟต์แวร์ในวงจรชีวิตการพัฒนาซอฟต์แวร์ได้อีกด้วย คณะนักวิจัยต่างๆนำข้อมูลซอฟต์แวร์อาร์ไคฟว์มาวิเคราะห์ในหลากหลายรูปแบบต่างๆกัน เช่น การวิเคราะห์เพื่อทำความเข้าใจพัฒนาการของการพัฒนาซอฟต์แวร์ (Software Evolution) (Ball et al., 1997) การวิเคราะห์เพื่อดักจับพัฒนาการของการเชื่อมโยงกัน (Evolution Coupling) ระหว่างคลาส (Bieman et al., 2003) หรือระหว่างไฟล์ (Gall et al.,1998; Burch et al., 2005) ต่างๆในระหว่างการพัฒนาโปรแกรม การวิเคราะห์เพื่อดักจับพัฒนาการของการเกิดขึ้นต่อกันระหว่างไฟล์พร้อมระบุลำดับการเปลี่ยนแปลง (Burch et al., 2005) การวิเคราะห์เพื่อดักจับพัฒนาการของการเกิดขึ้นต่อกันอย่างละเอียดระหว่างส่วนย่อยภายในโปรแกรม (Program Entities) อย่างเช่น ระหว่างฟังก์ชั่นหรือระหว่างตัวแปร (ซึ่งจัดว่าละเอียดกว่าการพิจารณาเพียงแค่ระดับระหว่างคลาสหรือระหว่างไฟล์) (Zimmermann et al., 2004) การวิเคราะห์เพื่อค้นหารูปแบบการเรียกใช้ซอฟต์แวร์ไลบรารี่ (Software Libraries) ที่ถูกต้องเพื่อการนำรูปแบบเหล่านั้นกลับมาใช้ใหม่ (Michail, 2000) และรูปแบบการเรียกใช้ซอฟต์แวร์ไลบรารี่ที่ผิดและนำไปสู่การเกิดข้อผิดพลาดได้ (Li et al., 2005; Livshits et al., 2005; Williams et al., 2005) งานวิจัยในอดีตเหล่านี้แสดงให้เห็นถึงประโยชน์ของการทำเหมืองข้อมูลด้วยกฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ ซึ่งมีส่วนช่วยในการทำงานร่วมกันของนักพัฒนาซอฟต์แวร์ระหว่างขั้นตอนการพัฒนาซอฟต์แวร์ในวงจรชีวิตการพัฒนาซอฟต์แวร์เป็นอย่างมาก ประโยชน์ของการทำเหมืองข้อมูลดังกล่าวได้แก่ 1) แนะนำและคาดการณ์ล่วงหน้าว่านักพัฒนาซอฟต์แวร์ควรจะแก้ไขคลาส ไฟล์ หรือฟังก์ชั่นใดต่อไปหลังจากที่ได้แก้ไขคลาส ไฟล์ หรือฟังก์ชั่นหนึ่งไปแล้ว 2) ป้องกันการเกิดข้อผิดพลาดจากการแก้ไขคลาส ไฟล์ หรือฟังก์ชั่นใดอย่างไม่สมบูรณ์ 3) สามารถดักจับการเกิดของการขึ้นต่อกันระหว่างคลาส ไฟล์ หรือฟังก์ชั่นได้ โดยเฉพาะอย่างยิ่งการขึ้นต่อกันที่ไม่สามารถดักจับได้ในระหว่างขั้นตอนการวิเคราะห์และออกแบบซอฟต์แวร์ (Zimmermann et al., 2004)

ในงานวิจัยนี้สนใจเฉพาะการประยุกต์ใช้การทำเหมืองข้อมูลด้วยกฎความสัมพันธ์จากข้อมูลซอฟต์แวร์อาร์ไคฟว์ในขั้นตอนการพัฒนาซอฟต์แวร์ โดยจะเน้นเฉพาะความสัมพันธ์ในระดับ

ที่ละเอียดที่สุด นั่นก็คือในระดับความสัมพันธ์ระหว่างฟังก์ชั่นหรือเมธอด รูปแบบการเรียกใช้ฟังก์ชั่นหรือเมธอดในคลาส (Function-Call-Usage Pattern) คือเซตหรือบัญชีรายการของการเรียกใช้ฟังก์ชั่นหรือเมธอดในคลาสที่พบอยู่ในซอร์สโค้ดของซอฟต์แวร์ รูปแบบการเรียกใช้ฟังก์ชั่นหรือเมธอดในคลาสเหล่านี้มักจะเกิดขึ้นมาจากสัญชาตญาณและองค์ความรู้สามัญของนักพัฒนาซอฟต์แวร์ รูปแบบการเรียกใช้ฟังก์ชั่นหรือเมธอดในคลาสบางรูปแบบก็เกิดขึ้นมาโดยที่นักพัฒนาซอฟต์แวร์ไม่รู้ตัว ซึ่งทั้งหมดนี้มักจะไม่ได้ถูกนำมาจัดทำเป็นเอกสารอย่างเป็นทางการและอยู่นอกเหนือความสามารถของการตรวจหาจุดบกพร่องของระบบตรวจจุดบกพร่องภายในซอฟต์แวร์ด้วย

ในช่วงปี 2005 คณะวิจัยของ Li และคณะวิจัยของ Livshits (Li et al., 2005; Livshits et al., 2005) ได้ทำการประยุกต์เทคนิคการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์ในการค้นหารูปแบบการเรียกใช้ฟังก์ชั่นหรือเมธอดในคลาสของระบบซอฟต์แวร์ขนาดใหญ่เพื่อที่จะแก้ไขปัญหาดังที่กล่าวไว้ งานวิจัยของคณะวิจัยทั้ง 2 ได้แสดงให้เห็นว่าการประยุกต์เทคนิคการทำเหมืองข้อมูลด้วยกฎความสัมพันธ์ในการค้นหารูปแบบการเรียกใช้ฟังก์ชั่นหรือเมธอดในคลาสนั้นเป็นเทคนิคที่ค่อนข้างมีประสิทธิภาพมากแต่ในบางกรณีนั้นสามารถทำให้เกิดผลลัพธ์ของการค้นหาที่เป็นผลบวกลวง (False Positive) เป็นจำนวนมาก กล่าวคือการใช้เทคนิคดังกล่าวอาจก่อให้เกิดกฎความสัมพันธ์ที่เป็นไปได้ออกมาโดยความจริงแล้วไม่ได้มีกฎความสัมพันธ์นั้นอยู่จริงๆ

นอกจากการนั้นการใช้เทคนิคการค้นหากฎความสัมพันธ์ก็ยังคงมีข้อด้อยที่สำคัญในการค้นหารูปแบบที่เรียกใช้ฟังก์ชั่นหรือเมธอดในคลาสก็คือ ธรรมชาติการเรียกใช้ฟังก์ชั่นหรือเมธอดในคลาสนั้นลำดับของการเรียกใช้ฟังก์ชั่นหรือเมธอดในคลาสย่อมมีความสำคัญ เช่น เมธอด open() ย่อมถูกเรียกใช้งานก่อนเมธอด close() แต่การใช้เทคนิคการค้นหากฎความสัมพันธ์ไม่สามารถบ่งชี้ถึงลำดับได้ (Huzefa Kagdi et al., 2007)

ต่อมาไม่นานงานวิจัยของ Burch และคณะ (Burch et al., 2005) ได้ทำการประยุกต์การทำเหมืองข้อมูลด้วยเทคนิคการค้นหารูปแบบลำดับ (Sequential pattern Mining) กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ เพื่อแก้ไขข้อบกพร่องการใช้เทคนิคค้นหากฎความสัมพันธ์ที่ไม่ได้ให้ความสำคัญกับลำดับของการเรียกฟังก์ชั่นหรือเมธอดในคลาส ผลลัพธ์ที่ได้จากการใช้เทคนิคการค้นหารูปแบบลำดับนั้นมีความแม่นยำสูงมากและให้ผลลัพธ์ที่เป็นผลบวกลวงน้อยกว่าเทคนิคอื่นๆ แต่อย่างไรก็ดีการทำเหมืองข้อมูลด้วยเทคนิคการค้นหารูปแบบลำดับนั้นมาพร้อมกับค่าใช้จ่ายในการประมวลผลที่สูงมาก ด้วยเหตุนี้ การทำเหมืองข้อมูลเพื่อค้นหารูปแบบที่มีความสัมพันธ์กันใน

ข้อมูลซอฟต์แวร์อาร์ไคฟว์ในทางปฏิบัตินั้นเหมาะกับการใช้เทคนิคการค้นหากฎความสัมพันธ์มากกว่าการใช้เทคนิคการค้นหารูปแบบลำดับถึงแม้ว่าการใช้เทคนิคการค้นหารูปแบบลำดับจะให้ผลลัพธ์ที่แม่นยำกว่าก็ตาม (Burch et al., 2005)

2.8 ขั้นตอนวิธีการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Data Mining in Software Archives)

วัตถุประสงค์ของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Data Mining in Software Archives) นั้นก็คือ การสร้างเซตของคำแนะนำในการเปลี่ยนแปลงแก้ไขให้กับนักพัฒนาในระหว่างขั้นตอนการพัฒนาซอฟต์แวร์เมื่อนักพัฒนาได้ทำให้เกิดเหตุการณ์ใดเหตุการณ์หนึ่งเกิดขึ้น การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์มีขั้นตอนหลักทั้งหมด 2 ขั้นตอน คือ 1) การจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Preparing Data for Mining in Software Archives) และ 2) การทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Data Mining in Software Archives) รายละเอียดของแต่ละขั้นตอนอธิบายได้ดังต่อไปนี้

2.8.1 การจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Preparing Data for Mining in Software Archives)

ในปี ค.ศ. 1997 คณะวิจัยของ Ball เป็นคณะวิจัยแรกที่ได้เริ่มทำงานวิจัยเกี่ยวกับข้อมูลซอฟต์แวร์อาร์ไคฟว์และได้พบว่าข้อมูลซอฟต์แวร์อาร์ไคฟว์ซึ่งประกอบด้วยแฟ้มข้อมูลซอร์สโค้ดและแฟ้มข้อมูลบันทึกของซอฟต์แวร์ควบคุมการเปลี่ยนแปลงแก้ไขนั้นสามารถแสดงได้ถึงพัฒนาการของการพัฒนาซอฟต์แวร์หรือระบบนั้นๆ และยังสามารถแสดงถึงการเชื่อมโยงกันหรือความสัมพันธ์กันระหว่าง 2 คลาส (Class) ภายในซอฟต์แวร์หรือระบบได้ด้วย นอกจากนั้นยังแสดงให้เห็นถึงความสัมพันธ์ของสิ่งต่างๆ (aspects) ในขั้นตอนการพัฒนาซอฟต์แวร์ที่มีผลต่อการเปลี่ยนแปลงข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Ball et al., 1997)

งานวิจัยของ Ball และคณะในปี ค.ศ. 1997 นี้ถือเป็นจุดเริ่มต้นแรกให้เกิดงานวิจัยเกี่ยวกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ขึ้นมาอีกมากมายในเวลาต่อมา ซึ่งสามารถยกตัวอย่างได้ดังต่อไปนี้

1) การประยุกต์ใช้การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ เพื่อช่วยเหลือการทำงานของนักพัฒนาซอฟต์แวร์อาทิเช่น คาดการณ์และแนะนำนักพัฒนาซอฟต์แวร์ว่าควรจะต้องแก้ไขซอร์สโค้ดส่วนไหนต่อไป แสดงการเชื่อมโยงกันของคลาสที่เกิดขึ้นในระหว่างการพัฒนาซึ่งไม่สามารถดักจับพบได้ในช่วงของการออกแบบซอฟต์แวร์ และป้องกันไม่ให้เกิดข้อผิดพลาดที่เกิดขึ้นจากการแก้ไขซอร์สโค้ดไม่สมบูรณ์ (Zimmermann et al., 2004; Ying et al., 2004; Livshits et al., 2005; Weißgerber et al., 2005; Yu et al., 2007)
2) การประยุกต์ใช้การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ เพื่อค้นหารูปแบบการเรียกใช้ซอฟต์แวร์ไลบรารี่ (Software Libraries) ที่ถูกต้องเพื่อการนำรูปแบบเหล่านั้นกลับมาใช้ใหม่ (Michail, 2000) และค้นหารูปแบบการเรียกใช้ซอฟต์แวร์ไลบรารี่ที่ผิดและนำไปสู่การเกิดข้อผิดพลาดได้ (Li et al., 2005; Livshits et al., 2005; Williams et al., 2005)
3) การจินตทัศน์ผลของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ เพื่อแสดงให้นักพัฒนาซอฟต์แวร์เห็นถึงพัฒนาการของการเชื่อมโยงกัน (Evoluation Coupling) ของคลาสที่เกิดขึ้นในระหว่างการพัฒนาซอฟต์แวร์ (Burch et al., 2005; Voinea et al., 2005; Voinea et al., 2006; Weissgerber et al., 2007)

การทำเหมืองข้อมูลกับข้อมูลประเภทต่างๆนั้นจะต้องมีขั้นตอนสามัญขั้นตอนหนึ่งที่ต้องทำก่อนเป็นอันดับแรกเสมอนั้นก็คือขั้นตอนการจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูล (Preparing Data for Mining) หรือที่เรียกว่า ขั้นตอนก่อนกระบวนการทำเหมืองข้อมูล (Preprocessing) งานวิจัยทุกงานวิจัยเกี่ยวกับการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ที่กล่าวไปในข้างต้นนั้นก็ต้องผ่านการจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Preparing Data for Mining in Software Archives) ด้วยเช่นกัน ขั้นตอนการจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลนี้ถือเป็นขั้นตอนที่มีความสำคัญมากสำหรับการทำเหมืองข้อมูลในทุกๆประเภท เพราะขั้นตอนการจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลนั้นจะส่งผลกระทบโดยตรงกับคุณภาพของการวิเคราะห์ผลลัพธ์ที่จะได้มาจากการทำเหมืองข้อมูล

ในหัวข้อนี้จะกล่าวถึงขั้นตอนสำคัญ 4 ขั้นตอนที่จะเกิดขึ้นในการจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ ที่ถูกเสนอขึ้นมาโดย Zimmermann และคณะ ในปี

2004 (Zimmermann et al., 2004) และเป็นขั้นตอนวิธีที่มีงานวิจัยที่เกี่ยวข้องกับการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟ์วหลายงายวิจัย (Zimmermann et al., 2005; Livshits et al., 2005; Williams et al., 2005; Breu et al., 2006; Weißgerber et al, 2006) นำไปใช้ ขั้นตอนทั้ง 4 ขั้นตอนได้แก่ 1) การสกัดข้อมูล (Data Extraction) 2) การซ่อมแซมทรานแซคชั่น (Restoring Transactions) 3) การระบุการเปลี่ยนแปลงแก้ไขในระดับเอนทิตี้ (Mapping Changes to Entities) และ 4) การกำจัดสิ่งแปลกปลอม (Data Cleaning) แต่ก่อนที่จะเริ่มการอธิบายรายละเอียดของการทำงานในแต่ละขั้นตอนข้างต้น ผู้วิจัยจำเป็นต้องนิยามคำศัพท์และสัญลักษณ์ที่เกี่ยวข้องกับการอธิบายขั้นตอนการทำงานดังต่อไปนี้

- เอนทิตี้ (Entity) คือ เอกลักษณ์หรือสิ่งที่ผู้วิจัยสนใจศึกษา ในที่นี้คำว่า เอนทิตี้ สามารถหมายถึง แฟ้มข้อมูลเอกสาร คลาส เมธอดหรือฟังก์ชัน และตัวแปร นิยามให้เอนทิตี้ e ถูกเขียนอยู่ในรูปแบบ (c, i, p) โดยที่ c คือหมวดหรือประเภทของเอนทิตี้นั้น (syntactic category) i คือชื่อของเอนทิตี้นั้น (identifier) และ p คือเอนทิตี้ที่เป็นเอนทิตี้แม่ของเอนทิตี้นั้นหรือใช้สัญลักษณ์ ... แทนในกรณีที่เอนทิตี้นั้นคือเอนทิตี้รากหรือในกรณีที่ต้องการละไว้ในฐานที่เข้าใจ ตัวอย่างของเอนทิตี้เช่น (method, initDefaults(), (Class, Comp, (file, Comp.java,...))) แสดงถึงเมธอดชื่อ initDefaults() ของคลาส Comp ในแฟ้มข้อมูล Comp.java
- การเปลี่ยนแปลงแก้ไข (Changes, Revisions) คือ เหตุการณ์ที่มีนักพัฒนาแก้ไขเอนทิตี้ใดๆ คำว่า เปลี่ยนแปลงแก้ไข ในที่นี้สามารถแสดงได้ 3 มิติ คือ 1) การเปลี่ยนแปลงเอนทิตี้ (alter) ใช้สัญลักษณ์ alter(e) แทนการเปลี่ยนแปลงอะไรบางอย่างภายในเอนทิตี้ e 2) การเพิ่มลงในเอนทิตี้ (add to) ใช้สัญลักษณ์ add_to(e) แทนการเพิ่มเอนทิตี้ใหม่เข้าไปในเอนทิตี้ e 3) การลบออกจากเอนทิตี้ (delete from) ใช้สัญลักษณ์ del_from(e) แทนการลบเอนทิตี้ใดๆออกจากเอนทิตี้ e
- ทรานแซคชั่น (Transaction) คือ เซตของการเปลี่ยนแปลงแก้ไขที่เกิดขึ้นพร้อมกันและถูกคอมมิทเข้าสู่ระบบ โดยหนึ่งทรานแซคชั่นใดๆจะเป็นของนักพัฒนาเพียงคนเดียวเท่านั้น ตัวอย่างของทรานแซคชั่นเช่น T = {alter(method, initDefaults(), ...), alter(field, fKeys[], ...), add_to(file, Comp.java, ...)} การ

เปลี่ยนแปลงแก้ไขที่อยู่ภายในทรานแซคชั่นบางครั้งอาจถูกเรียกว่า รายการ (item)

- เหตุการณ์ (Situation) คือเซตของการเปลี่ยนแปลงแก้ไขใดๆ ใช้สัญลักษณ์ Q แทนเหตุการณ์ ตัวอย่างของเหตุการณ์เช่น Q = {alter(method, initDefaults(), ...)}
- กฎความสัมพันธ์ (Association Rules) ที่ได้จากการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ คือกฎความสัมพันธ์ที่ตอบข้อถามที่ว่า *ถ้านักพัฒนาเปลี่ยนแปลงแก้ไข (เปลี่ยนแปลง เพิ่มลง หรือ ลบออก) เอนทิตี้ใดเอนทิตี้หนึ่งแล้วนักพัฒนาคนนั้นควรจะต้องเปลี่ยนแปลงแก้ไขอะไรด้วย* ตัวอย่างของกฎความสัมพันธ์สามารถเขียนได้ดังนี้ {alter(field, fKeys[], ...)} → {alter(method, initDefaults(), ...), alter(file, plug.properties, ...)} แทนกฎความสัมพันธ์ที่ว่าเมื่อมีการเปลี่ยนแปลงตัวแปรชื่อ fKey[] แล้วจะมีการเปลี่ยนแปลงเมธอดชื่อ initDefaults() และการเปลี่ยนแปลงแฟ้มข้อมูลชื่อ plug.properties ด้วย
- เซตของคำแนะนำสำหรับเหตุการณ์ Q (The Set of Suggestions for Situation Q) คือ เซตของการเปลี่ยนแปลงแก้ไขที่นักพัฒนาควรจะทำตามหลังจากที่นักพัฒนาได้เปลี่ยนแปลงแก้ไขตามเหตุการณ์ Q โดยอ้างอิงมาจากเซตของกฎความสัมพันธ์ R เซตของคำแนะนำสำหรับเหตุการณ์ Q สามารถเขียนเป็นสัญลักษณ์ได้คือ $apply_R(Q) = U_{(Q \rightarrow x_2) \in R} x_2$ ตัวอย่างเซตของคำแนะนำสำหรับเหตุการณ์ Q เช่น $apply_R(Q)$ = {alter(method, initDefaults(), ...), add_to(file, Comp.java, ...)}

เมื่อได้ทำความเข้าใจกับนิยามและสัญลักษณ์ของคำศัพท์แล้ว ส่วนต่อไปนี้คือการอธิบายรายละเอียดของการทำงานในแต่ละขั้นตอนของการจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์

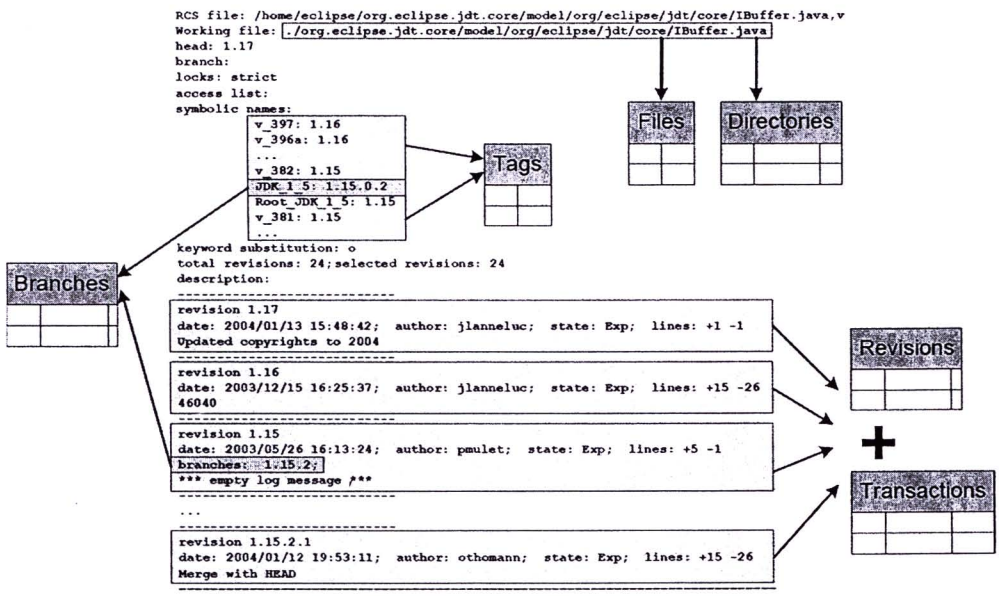
2.8.1.1 การสกัดข้อมูล (Data Extraction)

จุดประสงค์หนึ่งของการจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ คือ เพื่อให้สามารถเข้าถึงข้อมูลภายในระบบคอนเคอเรนท์เวอร์ชั่นได้อย่างรวดเร็วในขณะที่

กำลังทำเหมืองข้อมูล วิธีการที่ดีที่สุดที่จะบรรลุจุดประสงค์นั้นได้ก็คือการสำเนาแฟ้มข้อมูลบันทึกจากรีพอสิโทริของระบบคอนเคอเรนท์เวอร์ชั่น ทำการสกัดข้อมูลจากแฟ้มข้อมูลบันทึกนั้น และนำมาบันทึกไว้บนฐานข้อมูลของผู้วิจัยเอง (Zimmermann et al., 2004)

โดยทั่วไปแล้ว ข้อมูลที่จะถูกสกัดออกมาจะออกมาเป็นอย่างไรนั้นจะขึ้นอยู่กับว่ามีความต้องการวิเคราะห์ข้อมูลอะไร ตัวอย่างเช่น ถ้ามีความต้องการที่จะวิเคราะห์พัฒนาการของซอฟต์แวร์ (Software Evolution) ก็จะให้ความสนใจกับข้อมูลทุกอย่างที่บันทึกเอาไว้รวมถึงแฟ้มข้อมูลที่ถูกลบไปแล้วด้วย (Zimmermann et al., 2004) แต่ถ้ามีความต้องการที่จะวิเคราะห์เพื่อให้คำแนะนำกับนักพัฒนาในการแก้ไขแฟ้มข้อมูลที่เกี่ยวข้องกัน ก็จะให้ความสนใจกับข้อมูลที่บันทึกเอาไว้ในปัจจุบันเท่านั้น (Zimmermann et al., 2004)

การสกัดข้อมูลของระบบคอนเคอเรนท์เวอร์ชั่นเริ่มต้นจากการเรียกใช้คำสั่ง CVS log จากไดเรคทอรี่ราก (Root Directory) ของโครงการพัฒนาซอฟต์แวร์ที่ต้องการ ผลลัพธ์ที่ส่งกลับคืนมาก็คือแฟ้มข้อมูลซอร์สโค้ด (Source Code Files) และแฟ้มข้อมูลบันทึก (Log File) ของระบบคอนเคอเรนท์เวอร์ชั่นที่บันทึกอยู่บนรีพอสิโทริ แฟ้มข้อมูลบันทึกที่ได้มาจะถูกนำมาวิเคราะห์รูปแบบไวยากรณ์ (Parse) และถูกนำไปบันทึกลงฐานข้อมูลตามแผนภาพข้างล่างนี้ (Zimmermann et al., 2004)



รูปที่ 2-4 แสดงการทำงานของขั้นตอนการสกัดข้อมูล (Data Extraction)

ข้อมูลที่ได้จากการสกัดข้อมูลจากแฟ้มข้อมูลบันทึกก็คือ 1) แฟ้มข้อมูล (Files) ทั้งหมดที่อยู่ในโครงการนี้ ทั้งที่เป็นแฟ้มข้อมูลซอร์สโค้ดและแฟ้มข้อมูลอื่นๆ ด้วย 2) ไดเรคทอรี่ (Directories) ทั้งหมดที่อยู่ในโครงการนี้ 3) การเปลี่ยนแปลงแก้ไขแฟ้มข้อมูล (Revisions) 4) ทรานแซคชั่นของการเปลี่ยนแปลงแก้ไข (Transactions) 5) การตั้งชื่อแท็ก (Tags) และ 6) การต่อกิ่ง (Branches) รายละเอียดของขั้นตอนการสกัดข้อมูลสามารถอธิบายได้ดังต่อไปนี้ (Zimmermann et al., 2004)

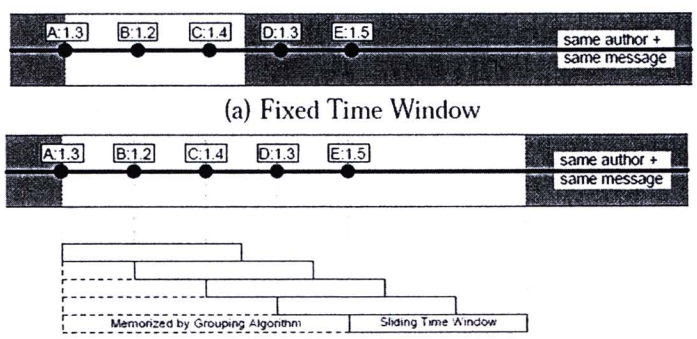
- แอททริบิวต์ RCS file ของแต่ละส่วน (Sections) ในแฟ้มข้อมูลบันทึกสามารถสกัดข้อมูลออกมาเป็นรายชื่อและรายละเอียดของแฟ้มข้อมูล (Files) และไดเรคทอรี่ (Directories) ทั้งหมดของโครงการ ตัวอย่างจากรูปข้างต้นจะได้ แฟ้มข้อมูลชื่อ IBuffer.java และไดเรคทอรี่ ./org.eclipse.jdt.core/Model/org/eclipse/jdt/core/
- แอททริบิวต์ description ประกอบด้วยส่วนย่อยหลายส่วนแต่ละส่วนแสดงถึงการเปลี่ยนแปลงแก้ไขแต่ละครั้งที่เกิดขึ้น ข้อมูลในแต่ละส่วนย่อยนี้สามารถสกัดข้อมูลออกมาเป็นรายการการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูล (Revisions) ได้ ตัวอย่างจากรูปข้างต้นจะได้ รายการการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลทั้งหมด 4 รายการ คือ รายการการเปลี่ยนแปลงแก้ไขหมายเลข 1.15.2.1 1.15 1.16 และ 1.17
- รายการการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลที่ได้มาข้างต้นจะถูกนำมาพิจารณาว่ารายการใดบางที่เกิดขึ้นในเวลาเดียวกันและเกิดขึ้นโดยนักพัฒนาคนเดียวกันจะถูกรวมกันไว้เป็นทรานแซคชั่นของการเปลี่ยนแปลงแก้ไข (Transactions) เดียวกัน ตัวอย่างจากรูปข้างต้นจะได้ว่ามีทรานแซคชั่นทั้ง 4 ทรานแซคชั่นและแต่ละทรานแซคชั่นประกอบด้วยรายการการเปลี่ยนแปลงแก้ไขเพียง 1 รายการ
- แอททริบิวต์ symbolic name ในแต่ละส่วนย่อยของแอททริบิวต์ description ในแฟ้มข้อมูลบันทึกสามารถสกัดข้อมูลออกมาเป็นรายชื่อของแท็ก (Tags) ที่นักพัฒนาตั้งไว้ให้กับการเปลี่ยนแปลงแก้ไขนั้นๆได้ ตัวอย่างจากรูปข้างต้นจะได้ว่าสำหรับแฟ้มข้อมูล IBuffer.java มีการตั้งชื่อแท็กทั้งหมดหลายชื่อและมีชื่อหนึ่งคือ V_397 ที่ตั้งไว้ให้กับรายการการเปลี่ยนแปลงแก้ไขหมายเลข 1.16
- การต่อกิ่ง (Branches) สามารถสกัดมาจากรายการการเปลี่ยนแปลงแก้ไขและชื่อแท็ก ตัวอย่างจากรูปข้างต้น เช่น ชื่อแท็ก JDK_1_5 ตั้งให้รายการการเปลี่ยนแปลงแก้ไขหมายเลข 1.15.0.2 จะได้ชื่อของการต่อกิ่งนี้เป็น 1.15.2 ส่วนจุดต่อกิ่งนั้นได้มาจากตารางแฮช (Hash Map) ที่ใช้ชื่อของการต่อกิ่งเป็นคีย์

2.8.1.2 การซ่อมแซมทรานแซคชั่น (Restoring Transactions)

ระบบคอนเคอเรนท์เวอร์ชั่นโดยทั่วไปนั้นจะไม่มีการบันทึกเอาไว้ว่าแฟ้มข้อมูลใดบ้างที่ถูกเปลี่ยนแปลงแก้ไขร่วมกันในการคอมมิทแต่ละครั้ง แต่ว่าข้อมูลการเปลี่ยนแปลงแก้ไขดังกล่าวมีความจำเป็นต่อการนำมาวิเคราะห์ข้อมูลซอฟต์แวร์อาร์ไคฟว์ (Zimmermann et al., 2004; Gall et al., 2003) วิธีการให้ได้มาซึ่งข้อมูลการเปลี่ยนแปลงแก้ไขที่เกิดร่วมกันในแต่ละการคอมมิทก็คือการเข้าไปอ่านข้อมูลที่ถูกบันทึกเอาไว้ในแฟ้มข้อมูลบันทึก (Log File) บนเครื่องแม่ข่ายของระบบคอนเคอเรนท์เวอร์ชั่นว่ามีข้อความบันทึก (Log Message) ใดบ้างที่ระบุการเปลี่ยนแปลงแก้ไขทั้งหมดที่เกิดจากนักพัฒนาคนเดียวกันและเกิดขึ้นในเวลาเดียวกัน ข้อมูลการเปลี่ยนแปลงแก้ไขที่เกิดจากนักพัฒนาคนเดียวกันและเกิดขึ้นในเวลาเดียวกันจะถูกนำมาเปลี่ยนเป็นทรานแซคชั่น 1 ทรานแซคชั่นแล้วบันทึกลงในตาราง Transactions บนฐานข้อมูล คำว่า ในเวลาเดียวกัน ในที่นี้นั้นหมายรวมถึงในเวลาที่ใกล้เคียงกันด้วย เนื่องจากการคอมมิทในแต่ละครั้งอาจใช้เวลาในการดำเนินการหลายวินาทีหรือหลายนาที โดยเฉพาะอย่างยิ่งการคอมมิททีละหลายๆแฟ้มข้อมูล (Zimmermann et al., 2004) ดังนั้นในแต่ทางปฏิบัติแล้วนอกจากการพิจารณาที่การคอมมิทในเวลาเดียวกันแล้วยังต้องมีวิธีการพิจารณาที่การคอมมิทระหว่างช่วงของเวลา (Time Interval) เดียวกันด้วย วิธีการพิจารณาการเปลี่ยนแปลงแก้ไขระหว่างช่วงของเวลานั้นมี 2 วิธีคือ

1) วิธีกำหนดกรอบเวลาที่แน่นอน (Fixed Time Windows) คือ การกำหนดกรอบของช่วงเวลามากสุดที่จะพิจารณาว่าการเปลี่ยนแปลงแก้ไขดังกล่าวนั้นอยู่ภายในทรานแซคชั่นเดียวกันหรือไม่ โดยกำหนดให้กรอบของช่วงเวลาจะเริ่มต้นขึ้นใหม่เมื่อมีการเปลี่ยนแปลงแก้ไขใหม่เกิดขึ้นครั้งแรกและกรอบจะคงที่เช่นนี้จนจบการพิจารณา การเปลี่ยนแปลงแก้ไขใดที่อยู่ในกรอบนี้ทั้งหมดจะถูกพิจารณาว่าอยู่ในทรานแซคชั่นเดียวกัน วิธีนี้เป็นวิธีที่ง่ายและสะดวกในการนำไปประยุกต์ใช้ วิธีการพิจารณาที่การเปลี่ยนแปลงแก้ไขระหว่างช่วงของเวลาเดียวโดยกำหนดกรอบเวลาที่แน่นอนถูกนำไปใช้ในงานวิจัยของ Ubranic´ และคณะ (Ubranic´ et al., 2003) และในงานวิจัยของ Gall และคณะในปี ค.ศ. 2003 (Gall et al., 2003)
2) วิธีเลื่อนกรอบเวลา (Sliding Time Windows) คือ การกำหนดช่องว่างระหว่างการเปลี่ยนแปลงแก้ไข 2 ครั้งที่มากที่สุด จุดเริ่มต้นของกรอบของช่วงเวลาจะถูกเลื่อนไปที่การเปลี่ยนแปลงแก้ไขครั้งต่อไปเสมอตราบใดที่การเปลี่ยนแปลงแก้ไขครั้งต่อไปนั้นมี

จุดเริ่มต้นอยู่ภายในกรอบเวลาของการเปลี่ยนแปลงแก้ไขครั้งก่อนหน้า ดังนั้นการพิจารณาด้วยวิธีเลื่อนกรอบเวลานี้จะสามารถรู้จำ (Recognize) การคอมมิทที่ใช้ระยะเวลาในการดำเนินการนานกว่าจะสมบูรณ์ได้ดีกว่าการพิจารณาด้วยวิธีกำหนดกรอบเวลาที่แน่นอน วิธีเลื่อนกรอบเวลานี้มีต้นกำเนิดมาจากโปรแกรมประยุกต์ที่มีชื่อว่า cvs2cl (http://www.red-bean.com/cvs2cl) และ CVSps (http://www.cobite.com/cvsps) (Zimmermann et al., 2004)



รูปที่ 2-5 แสดงการพิจารณาช่วงเวลาของการคอมมิทด้วยวิธีกำหนดกรอบเวลาที่แน่นอนและวิธีเลื่อนกรอบเวลา

รูปที่ 2-5 แสดงการพิจารณาช่วงเวลาของการคอมมิทด้วยวิธีกำหนดกรอบเวลาที่แน่นอนและวิธีเลื่อนกรอบเวลา (Zimmermann et al., 2004) ส่วน (a) แสดงการพิจารณาด้วยวิธีกำหนดกรอบเวลาที่แน่นอน บริเวณสีขาวเป็นบริเวณที่อยู่ภายในกรอบเวลาที่กำหนดเอาไว้แน่นอน ดังนั้นการแก้ไขเปลี่ยนแปลงที่แฟ้มข้อมูล A เวอร์ชั่นที่ 1.3 แฟ้มข้อมูล B เวอร์ชั่นที่ 1.2 และแฟ้มข้อมูล C เวอร์ชั่นที่ 1.4 จะถูกพิจารณาว่าเกิดขึ้นพร้อมกันและอยู่ภายในทรานแซคชั่นเดียวกัน รูปที่ 2-5 ส่วน (b) แสดงการพิจารณาด้วยวิธีเลื่อนกรอบเวลา โดยเริ่มต้นกรอบเวลาที่มีช่วงแน่นอนเริ่มต้นที่จุดการแก้ไขเปลี่ยนแปลงที่แฟ้มข้อมูล A เวอร์ชั่นที่ 1.3 และกรอบเวลาถูกเลื่อนไปเรื่อยๆ จนสิ้นสุดดังรูป ดังนั้นการแก้ไขเปลี่ยนแปลงที่แฟ้มข้อมูล A เวอร์ชั่นที่ 1.3 แฟ้มข้อมูล B เวอร์ชั่นที่ 1.2 แฟ้มข้อมูล C เวอร์ชั่นที่ 1.4 แฟ้มข้อมูล D เวอร์ชั่นที่ 1.3 และ แฟ้มข้อมูล E เวอร์ชั่นที่ 1.5 จะถูกพิจารณาว่าเกิดขึ้นพร้อมกันและอยู่ภายในทรานแซคชั่นเดียวกัน

โดยปกติ การใช้วิธีเลื่อนกรอบเวลาสำหรับทุกการเปลี่ยนแปลงแก้ไข $\alpha_1, \alpha_2, \ldots, \alpha_k$ (เรียงตามลำดับเวลาที่บันทึก (time(α_i))) ที่เป็นส่วนหนึ่งของทรานแซคชั่น T เดียวกันนั้น จะต้องอยู่ภายใต้เงื่อนไข

$$\forall \alpha_i \in T : author(\alpha_i) = author(\alpha_1)$$

$$\forall \alpha_i \in T : \log\_message(\alpha_i) = \log\_message(\alpha_1)$$

$$\forall i \in \{2, \dots, k\} : |time(\alpha_i) - time(\alpha_{i-1})| \leq 200sec$$

นอกจากนั้นการเปลี่ยนแปลงแก้ไขเวอร์ชั่นของแต่ละแฟ้มข้อมูลจะปรากฏอยู่บน 1 ทรานแซคชั่นได้เพียงครั้งเดียว เนื่องจากระบบคอนเคอเรนท์เวอร์ชั่นไม่อนุญาตให้มีการคอมมิทการเปลี่ยนแปลงเวอร์ชั่นของแฟ้มข้อมูลเดียวกัน 2 ครั้งในเวลาเดียวกันได้ ดังนั้นจึงมีเงื่อนไขเพิ่มอีก 1 ข้อดังนี้

$$\forall \alpha_a, \alpha_b \in T : \alpha_a \neq \alpha_b \rightarrow file(\alpha_a) \neq file(\alpha_b)$$

ขั้นตอนวิธีในการรวมกลุ่ม (grouping) การเปลี่ยนแปลงแก้ไขเวอร์ชั่นของแต่ละแฟ้มข้อมูลให้มาเป็นทรานแซคชั่นนั้น เป็นขั้นตอนวิธีที่เรียบง่ายและตรงไปตรงมา ดังนี้ (Zimmermann et al., 2004)

1) เรียงลำดับการเปลี่ยนแปลงแก้ไขเวอร์ชั่นของแต่ละแฟ้มข้อมูลตาม เวลา ชื่อของนักพัฒนาที่ทำการเปลี่ยนแปลง และข้อความในแฟ้มข้อมูลบันทึก (log messager)
2) เริ่มต้นทรานแซคชั่นที่ i โดยวนซ้ำพิจารณาแต่ละ การเปลี่ยนแปลงแก้ไขเวอร์ชั่นของแต่ละแฟ้มข้อมูล ถ้ากรอบเวลาของทรานแซคชั่นปัจจุบันจบลง หรือ ชื่อนักพัฒนา เวลา และ/หรือ ข้อความในแฟ้มข้อมูลบันทึกของการเปลี่ยนแปลงแก้ไขเวอร์ชั่นของแฟ้มข้อมูลที่ i แตกต่างจากชื่อนักพัฒนา เวลา และ/หรือ ข้อความในแฟ้มข้อมูลบันทึกของการเปลี่ยนแปลงแก้ไขเวอร์ชั่นของแฟ้มข้อมูลที่ i-1 แล้ว ถือเป็นการสิ้นสุดทรานแซคชั่นที่ i
3) วนซ้ำข้อ 2 ไปจนกว่าจะหมดข้อมูลเปลี่ยนแปลงแก้ไขเวอร์ชั่นของแต่ละแฟ้มข้อมูล

งานวิจัยของ Zimmermann และคณะในปี ค.ศ. 2004 (Zimmermann et al., 2004) งานวิจัยของ Livshits และคณะในปี ค.ศ. 2005 (Livshits et al., 2005) งานวิจัยของ Weißgerber และคณะในปี ค.ศ. 2005 (Weißgerber et al., 2005) และปี ค.ศ. 2007 (Weißgerber et al., 2007) เลือกพิจารณาการเปลี่ยนแปลงแก้ไขเวอร์ชั่นระหว่างช่วงของเวลาด้วยวิธีเลื่อนกรอบเวลาและกำหนดช่วงของกรอบเวลาอยู่ที่ 200 วินาที

2.8.1.3 การระบุการเปลี่ยนแปลงแก้ไขในระดับเอนทิตี้ (Mapping Changes to Entities)

ข้อมูลที่ถูกจัดเก็บไว้ในรีพอสิโทริของระบบคอนเคอเรนท์เวอร์ชั่นนั้นมีเพียงข้อมูลแฟ้มข้อมูลทุกแฟ้มข้อมูลในโครงการและข้อมูลการเปลี่ยนแปลงแก้ไขในระดับแฟ้มข้อมูล (File) หรือคลาส (Class) ที่เก็บอยู่ในรูปของแฟ้มข้อมูลบันทึก (Log File) เท่านั้น แต่ไม่มีบันทึกว่าการเปลี่ยนแปลงแก้ไขที่เกิดขึ้นนั้นเกิดขึ้นกับฟังก์ชั่น (Function) หรือ เมธอด (Method) ใดบ้าง มีตัวแปร (Variable) ใดถูกเพิ่มเข้ามา แก้ไข หรือถูกลบออกไปบ้าง (Zimmermann et al., 2004)

การเปรียบเทียบความแตกต่างอย่างละเอียดถึงในระดับตัวแปร และ ฟังก์ชั่น หรือ เมธอด นี้มีวิธีการหลายวิธี วิธีการอย่างง่ายๆก็คือการเปรียบเทียบในระดับแฟ้มข้อมูล โดยการประยุกต์ใช้ฟังก์ชั่นดิฟฟ์ (diff) (Miller et al., 1985) กับแต่ละบรรทัดของซอร์สโค้ดระหว่างแฟ้มข้อมูลของเวอร์ชั่นเก่ากับแฟ้มข้อมูลเวอร์ชั่นใหม่ ผลลัพธ์ที่ได้คือส่วนของโค้ดที่มีการเปลี่ยนแปลง ส่วนที่มีการเพิ่มเข้าใหม่และส่วนที่ถูกลบออกไป วิธีการนี้มีข้อเสียที่สำคัญอยู่ 2 ประการคือ 1) คุณภาพของผลลัพธ์ที่ได้จะขึ้นอยู่กับคุณภาพของฟังก์ชั่นดิฟฟ์ (diff) ที่นำมาใช้ 2) วิธีการนี้ระบุความแตกต่างได้แค่ในระดับบรรทัดของซอร์สโค้ด ไม่สามารถระบุได้ว่าเป็นตรงไหนของบรรทัดนั้น (Zimmermann et al., 2004)

วิธีการที่มีความแม่นยำสูงกว่าแต่ก็มีค่าใช้จ่ายในการคำนวณสูงวิธีหนึ่ง คือการกำหนดเอนทิตี้ (ตัวแปร ฟังก์ชั่นหรือเมธอด คลาส แฟ้มข้อมูล) ทั้งหมดภายในแฟ้มข้อมูลทั้ง 2 เวอร์ชั่นโดยการนำไปผ่านตัววิเคราะห์ไวยากรณ์ (Parser) จากนั้นก็ทำการเปรียบเทียบซอร์สโค้ดของเอนทิตี้เดียวกันใน 2 เวอร์ชั่น หรือกล่าวคือเป็นการประยุกต์ใช้ฟังก์ชั่นดิฟฟ์ (diff) ในระดับของเอนทิตี้นั่นเอง วิธีการนี้สามารถดำเนินการได้ดังต่อไปนี้ (Zimmermann et al., 2004)

1) กำหนดเซต E_1 คือเซตของเอนทิตี้ที่มีอยู่ทั้งหมดในเวอร์ชั่น r_1 ของแฟ้มข้อมูล และกำหนดเซต E_2 คือเซตของเอนทิตี้ที่มีอยู่ทั้งหมดในเวอร์ชั่น r_2 ของแฟ้มข้อมูลเดียวกัน
2) เอนทิตี้ที่ถูกเพิ่มเข้ามาใหม่สามารถหาได้จาก $E_2 - E_1$
3) เอนทิตี้ที่ถูกลบออกไปสามารถหาได้จาก $E_1 - E_2$
4) ทุกๆเอนทิตี้ที่อยู่ในเซต $E_1 \cap E_2$ อาจจะเป็นเอนทิตี้ที่มีการเปลี่ยนแปลงภายใน การตัดสินว่าเอนทิตี้ใดบ้างที่มีการเปลี่ยนแปลงแก้ไขสามารถทำได้โดยการประยุกต์ใช้ฟังก์ชั่นดิฟฟ์ (diff) กับซอร์สโค้ดของเอนทิตี้นั้นๆของทั้ง 2 เวอร์ชั่น

ภายในแพลตฟอร์ม (Platform) ของอิคลิพซ์ (Eclipse) นั้นมีการจัดเตรียมโครงร่าง (Framework) สำหรับการเปรียบเทียบความแตกต่างระหว่าง 2 ซอร์สโค้ดใดๆที่มีประสิทธิภาพสูง และสามารถนำไปประยุกต์เพิ่มเติมได้ วิธีการทั้ง 2 วิธีการที่ได้กล่าวไปข้างต้นนั้นสามารถนำมาประยุกต์ใช้จริงได้โดยเรียกใช้ 2 โครงร่างดังต่อไปนี้ (Zimmermann et al., 2004)

1) โครงร่างเรนจ์ดิฟเฟอร์เรนเซอร์ (Range Differencer) คลาสเรนจ์ดิฟเฟอร์เรนเซอร์ (RangeDifferencer Class) ทำการเปรียบเทียบความแตกต่างของแฟ้มข้อมูล 2 เวอร์ชั่น โดยใช้โทเคน (Token) เป็นฐาน วิธีการนี้อ้างอิงวิธีการในการเปรียบเทียบมาจากขั้นตอนวิธีดิฟฟ์ (diff Algorithm) ดังเดิมของ Miller และ Myers (Miller et al., 1985) โทเคนแต่ละโทเคนจะถูกสร้างขึ้นมาโดยคลาสที่มีการอิมพลีเมนต์ (Implementing) อินเตอร์เฟส (Interface) ชื่อว่า ไอโทเคนคอมพาราเตอร์ (ITokenComparator interface) ตัวอย่างคลาสที่มีการอิมพลีเมนต์โทเคนคอมพาราเตอร์ คือ คลาสดอคไลน์คอมพาราเตอร์ (DocLineComparator Class) ที่สามารถคำนวณความแตกต่างของแต่ละบรรทัดในคลาสได้และให้ผลลัพธ์ออกมาเป็นรายการ (List object) ของบรรทัดที่แตกต่างกันระหว่าง 2 เวอร์ชั่นของแฟ้มข้อมูลที่นำมาเปรียบเทียบกัน
2) โครงร่างสตรัคเชอร์เมิร์จวิวเวอร์ (Structure Merge Viewer) คลาสดิฟเฟอร์เรนเซอร์ (Differrencer Class) ทำการเปรียบเทียบความแตกต่างของแฟ้มข้อมูล 2 เวอร์ชั่นโดยใช้โครงสร้างลำดับขั้น (hierarchical Structure) เป็นฐาน ผลลัพธ์ที่ได้จากการเปรียบเทียบคือต้นไม้ที่อธิบายการความแตกต่างระหว่าง 2 เวอร์ชั่นของแฟ้มข้อมูลอย่างละเอียด โครงสร้างลำดับขั้นแต่ละโครงสร้างที่เป็นตัวแทนของแฟ้มข้อมูลนั้นๆจะถูกสร้างขึ้นมาโดยคลาสที่มีการอิมพลีเมนต์อินเตอร์เฟสชื่อว่าไอสตรัคเชอร์ครีเอเตอร์ (IStructureCreator Interface) แต่ถ้าไม่ต้องการสร้างคลาสที่มีการอิมพลีเมนต์อินเตอร์เฟสไอสตรัคเชอร์ครีเอเตอร์ มาใช้เองก็สามารถใช้คลาสจาวาสตรัคเชอร์ครีเอเตอร์ (JavaStructureCreator Class) ที่แพลตฟอร์มอิคลิพซ์จัดเตรียมไว้ให้แล้วสำหรับการสร้างโครงสร้างลำดับขั้นของคลาสแฟ้มข้อมูลภาษาจาวา (Java)

2.8.1.4 การกำจัดสิ่งแปลกปลอม (Data Cleaning)

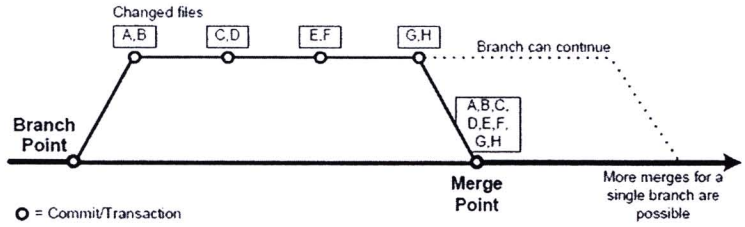
ในหัวข้อก่อนหน้านี้อธิบายถึงการสกัดข้อมูล การซ่อมแซมทรานแซคชั่น และการแปลงการเปลี่ยนแปลงแก้ไขไปสู่ระดับเอนทิตี้ ซึ่งผลลัพธ์ที่ได้มานั้นยังมีสิ่งแปลกปลอม (Noise) ปะปนมาด้วย ขั้นตอนการกำจัดสิ่งแปลกปลอม (Data Cleaning) เป็นขั้นตอนที่เข้าไปตรวจสอบข้อมูลทั้งหมดเพื่อค้นหาสิ่งแปลกปลอมและกำจัดสิ่งแปลกปลอมเหล่านั้นออกไป ลักษณะของข้อมูลทรานแซคชั่นที่จะถูกระบุว่าเป็นสิ่งแปลกปลอมมีอยู่ 2 ลักษณะคือ 1) ทรานแซคชั่นขนาดใหญ่ (Large Transactions) และ 2) ทรานแซคชั่นการผสานกิ่ง (Merge Transactions) รายละเอียดของสิ่งแปลกปลอมและวิธีการในการกำจัดสิ่งแปลกปลอมทั้ง 2 ลักษณะนี้สามารถอธิบายได้ดังต่อไปนี้ (Zimmermann et al., 2004)

- ทรานแซคชั่นขนาดใหญ่ (Large Transactions)

ทรานแซคชั่นขนาดใหญ่เป็นเหตุการณ์ปกติที่สามารถเกิดขึ้นได้จริงกับข้อมูลจริงในทุกประเภท ข้อมูลที่ได้จากระบบคอนเคอเรนท์เวอร์ชั่นนั้นอาจมีข้อมูลที่ไม่เกี่ยวกับข้องกับการนำไปวิเคราะห์มาปะปนอยู่ด้วย ข้อมูลที่ไม่เกี่ยวข้องกับการวิเคราะห์อาทิ ข้อความที่ระบุการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลที่เป็นการแก้ไขโครงสร้างภายในของแต่ละแฟ้มข้อมูลเองที่เกิดขึ้นต่อเนื่องกันแต่ไม่ได้มีความสัมพันธ์เชิงตรรกะระหว่างแฟ้มข้อมูลเหล่านั้น เช่น ข้อความ “Chage #include filenames from <foo.h> [sign] to <openssl.h>.” ที่ปรากฏต่อเนื่องกัน 552 ครั้งสำหรับการแก้ไขแฟ้มข้อมูล 552 แฟ้มข้อมูล ข้อความลักษณะดังกล่าวจะถูกระบุว่าเป็นสิ่งแปลกปลอมที่อาจส่งผลให้ผลลัพธ์ของการวิเคราะห์ผิดพลาดได้ วิธีกำจัดสิ่งแปลกปลอมลักษณะนี้ทำได้โดยการกรอง (filter out) ทรานแซคชั่นที่มีขนาดใหญ่เกินกว่าค่า N ที่กำหนดไว้ออก

- ทรานแซคชั่นการผสานกิ่ง (Merge Transactions)

ในระหว่างที่นักพัฒนากำลังพัฒนาซอฟต์แวร์ด้วยระบบคอนเคอเรนท์เวอร์ชั่นอยู่นั้นอาจมีการสร้างกิ่ง (Branches) ของเวอร์ชั่นขึ้นมาหลากหลายกิ่ง กิ่งเหล่านั้นบางกิ่งอาจจะถูกผสานกิ่ง (Merge) เข้ากับลำต้นเวอร์ชั่นหลักในอนาคต แต่บางกิ่งของเวอร์ชั่นก็ถูกปล่อยทิ้งเอาไว้



รูปที่ 2-6 แสดงตัวอย่างการต่อกิ่งและผสานกิ่ง

รูปที่ 2-6 แสดงตัวอย่างการต่อกิ่งและการผสานกิ่งอย่างง่ายในระบบคอนเคอเรนท์เวอร์ชั่น กิ่งที่ต่อออกมานั้นประกอบด้วยการคอมมิททั้งหมด 4 ทรานแซคชั่น ได้แก่ ทรานแซคชั่น {A, B} ทรานแซคชั่น {C, D} ทรานแซคชั่น {E, F} และทรานแซคชั่น {G, H} แฟ้มข้อมูล A B C D E F G และ H เหล่านี้จะถูกเปลี่ยนแปลงแก้ไขอีกครั้งตอนที่มีการผสานกิ่งเข้ากับลำต้นเกิดเป็นทรานแซคชั่น {A, B, C, D, E, F, G, H} ที่จุดผสาน (Merge Point) และเรียกทรานแซคชั่น {A, B, C, D, E, F, G, H} ว่า ทรานแซคชั่นการผสานกิ่ง

ทรานแซคชั่นการผสานกิ่งถูกระบุว่าเป็นสิ่งแปลกปลอมด้วยสาเหตุสำคัญ 2 สาเหตุคือ 1) ทรานแซคชั่นการผสานกิ่งเป็นทรานแซคชั่นที่ประกอบด้วยการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูลที่ไม่มีความสัมพันธ์เกี่ยวข้องกันอย่างแท้จริง ตัวอย่างเช่น ทรานแซคชั่นการผสานกิ่งข้างต้นมีการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูล B และการเปลี่ยนแปลงแก้ไขแฟ้มข้อมูล C อยู่ภายใน ซึ่งในความเป็นจริงทั้ง 2 แฟ้มข้อมูลไม่มีความสัมพันธ์เกี่ยวข้องกัน 2) ทรานแซคชั่นการผสานกิ่งถูกสร้างมาจากการนำการเปลี่ยนแปลงแก้ไขที่มีทั้งหมดบนกิ่งนั้นมาเรียงต่อกัน ดังนั้นทรานแซคชั่นการผสานกิ่งจึงเป็นทรานแซคชั่นที่มีความซ้ำซ้อนกับทรานแซคชั่นอื่นที่มีอยู่บนกิ่งนั้นอยู่แล้ว

จากเหตุผลทั้ง 2 ข้อข้างต้น การนำทรานแซคชั่นการผสานกิ่งไปใช้ในการทำเหมืองข้อมูลด้วย อาจทำให้ผลลัพธ์ที่ได้จากการทำเหมืองข้อมูลผิดพลาดได้ ดังนั้นทรานแซคชั่นการผสานกิ่งที่จะเกิดขึ้นทุกครั้งที่มีการผสานกิ่งเกิดขึ้นจะต้องถูกค้นหาให้พบและกำจัดออก

2.8.2 **การทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์** (Data Mining in Software Archives)

หลังจากที่ข้อมูลการเปลี่ยนแปลงแก้ไขภายในรีพอสิโทริของระบบคอนเคอเรนท์เวอร์ชั่นได้ถูกนำมาดำเนินการตามขั้นตอนการจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์

อาร์ไคฟ์ที่กล่าวไปข้างต้นและได้ผลลัพธ์ของการดำเนินการออกมาเป็นข้อมูลรายการทรานแซคชั่นแล้ว ข้อมูลรายการทรานแซคชั่นเหล่านั้นจะถูกนำมาเป็นข้อมูลเข้าสำหรับการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟ์ (Data Mining in Software Archives) ผลลัพธ์หรือข้อมูลออกที่ได้จากการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟ์ก็คือ กฎความสัมพันธ์ที่สามารถตอบข้อถามที่ว่า *ถ้านักพัฒนาเปลี่ยนแปลงแก้ไข (เปลี่ยนแปลง เพิ่มลง หรือ ลบออก) เอนทิตี้ใดเอนทิตี้หนึ่งแล้วนักพัฒนาคนนั้นควรจะต้องเปลี่ยนแปลงแก้ไขอะไรด้วย* ตัวอย่างเช่น

{alter(field, fKeys[], ...)} → {alter(method, initDefaults(), ...), alter(file, plug.properties, ...)}

กฎความสัมพันธ์ในข้างต้นนั้นหมายความว่า เมื่อไรก็ตามที่นักพัฒนามีการเปลี่ยนแปลงอะไรบางอย่างกับตัวแปรชื่อ fkey[] แล้วนักพัฒนาควรจะต้องไปเปลี่ยนแปลงแก้ไขอะไรบางอย่างในเมธอดชื่อ initDefaults() และเปลี่ยนแปลงแก้ไขอะไรบางอย่างในแฟ้มข้อมูลชื่อ plug.properties

โดยทั่วไปแล้ว กฎความสัมพันธ์ r จะอยู่ในรูปของคู่ (x_1, x_2) โดยที่เซตรายการ x_1 และ x_2 ไม่มีส่วนซ้อนทับกัน (disjoint) หรืออยู่ในรูปของ $x_1 \rightarrow x_2$ ซึ่ง x_1 จะถูกเรียกว่าเซตรายการที่มาก่อน (Antecedent Itemset) และ x_2 จะถูกเรียกว่าเซตรายการที่ตามมา (Consequent Itemset) กฎความสัมพันธ์แต่ละกฎนั้นถูกสร้าง (derived) ขึ้นมาจากการพิจารณาความน่าจะเป็นจากทรานแซคชั่นที่เกิดขึ้นจริงในอดีต วิธีที่นิยมนำมาใช้ในการประเมินความน่าสนใจของแต่ละกฎความสัมพันธ์คือการพิจารณาค่าสนับสนุน (Support) และค่าความเชื่อมั่น (Confidence) ของกฎความสัมพันธ์ แต่สำหรับการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟ์นั้นจะใช้วิธีการประเมินความน่าสนใจของแต่ละกฎความสัมพันธ์โดยการพิจารณาที่ค่าสนับสนุนนับ (Support Count) และค่าความเชื่อมั่น (Confidence) ของกฎความสัมพันธ์ (Zimmermann et al., 2005) นิยามของค่าสนับสนุนนับ และค่าความเชื่อมั่น ของกฎความสัมพันธ์สำหรับการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟ์นั้น สามารถอธิบายได้ดังนี้

- ค่าสนับสนุนนับ (Support Count) คือ จำนวนของทรานแซคชั่นที่กฎความสัมพันธ์นั้นปรากฏอยู่ ตัวอย่างเช่น สมมุติให้ตัวแปร fKey[] เคยปรากฏว่าถูกเปลี่ยนแปลงแก้ไขไปทั้งหมด 11 ทรานแซคชั่น ภายใน 11 ทรานแซคชั่นนั้นมี 10 ทรานแซคชั่นที่มีการเปลี่ยนแปลงแก้ไขเมธอด initDefaults() และแฟ้มข้อมูล plug.properties ด้วย ดังนั้นค่าสนับสนุนนับของกฎความสัมพันธ์ {alter(field, fKeys[], ...)} → {alter(method, initDefaults(), ...), alter(file, plug.properties, ...)} เท่ากับ 10 การทำเหมืองข้อมูลกับ

ข้อมูลซอฟต์แวร์อาร์ไคฟว์นั้นใช้ค่าสนับสนุนนับ แทนที่การใช้ค่าสนับสนุนอย่างในการทำเหมืองข้อมูลทั่วๆไปเพราะสาเหตุสำคัญ 2 ประการคือ 1) ค่าสนับสนุนนับ สามารถสื่อสารให้นักพัฒนาเข้าใจได้ดีกว่าค่าสนับสนุน กล่าวคือ ค่าสนับสนุนนับ เท่ากับ 10 หมายถึงที่ผ่านมาในอดีตเคยมีเหตุการณ์ตามกฎความสัมพันธ์เกิดขึ้นมาแล้วทั้งหมด 10 ครั้ง แต่ค่าสนับสนุน เท่ากับ 0.000145 นั้นไม่สามารถสื่อสารอะไรได้เลยถ้าไม่ได้บอกข้อมูลเพิ่มเติมว่าจำนวนทรานแซคชั่นที่มีอยู่ทั้งหมดนั้นเท่ากับเท่าไร 2) ค่าสนับสนุนนับ สามารถถูกนำไปใช้ในการวิเคราะห์อื่นๆได้ ตัวอย่างเช่น นอกจากการใช้กฎความสัมพันธ์ในการให้คำแนะนำนักพัฒนาในระหว่างการพัฒนาแล้ว กฎความสัมพันธ์สามารถบอกได้ถึงพัฒนาการของการเชื่อมโยงกันระหว่างแฟ้มข้อมูล (Evoluation Coupling) ได้ ในการระบุพัฒนาการของการเชื่อมโยงกันระหว่างแฟ้มข้อมูลนั้นใช้แค่เพียงค่าสนับสนุนนับก็สามารถระบุได้ เพราะจำนวนทรานแซคชั่นทั้งหมดมีผลต่อการระบุพัฒนาการของการเชื่อมโยงกันระหว่างแฟ้มข้อมูลน้อยมาก (Zimmermann et al., 2005)

- ค่าความเชื่อมั่น (Confidence) คือ ค่าความน่าจะเป็นที่จะพบกฎความสัมพันธ์ในทรานแซคชั่นที่มีเซตรายการที่มาก่อนอยู่ จากตัวอย่างข้างต้นค่าความเชื่อมั่นของกฎความสัมพันธ์ {alter(field, fKeys[], ...)} → {alter(method, initDefaults(), ...), alter(file, plug.properties, ...)} เท่ากับ 10/11 = 0.909

นิยามการเขียนกฎความสัมพันธ์พร้อมระบุค่าสนับสนุนนับ และค่าความเชื่อมั่น ในรูปสัญลักษณ์ดังนี้ r [s;c] โดยที่ r แทนกฎความสัมพันธ์ s แทนค่าสนับสนุนนับของกฎความสัมพันธ์นั้น และ c แทนค่าความเชื่อมั่นของกฎความสัมพันธ์นั้น ตัวอย่างเช่น {alter(field, fKeys[], ...)} → {alter(method, initDefaults(), ...)} [4;0.57]

สมมุติให้นักพัฒนามีการเปลี่ยนแปลงแก้ไขบางอย่างลงไป ทำให้เกิดเซตรายการของการเปลี่ยนแปลงแก้ไขขึ้นมา นิยามเซตรายการของการเปลี่ยนแปลงแก้ไขดังกล่าวว่าเป็นเซตเหตุการณ์ (Situation) และใช้สัญลักษณ์ Q แทนเหตุการณ์ เซตเหตุการณ์จะถูกปรับปรุง (Update) ทุกครั้งที่นักพัฒนาทำการบันทึกการเปลี่ยนแปลงแก้ไขลงสู่เครื่องลูกข่ายของนักพัฒนาเอง เมื่อนักพัฒนามั่นใจในเวอร์ชั่นใหม่ของแต่ละแฟ้มข้อมูลที่นักพัฒนาแก้ไขไปแล้วทำการคอมมิทเพื่อบันทึกการเปลี่ยนแปลงแก้ไขทั้งหมดลงสู่รีพอสิโทริของระบบคอนเคอเรนท์เวอร์ชั่น เซตเหตุการณ์สุดท้ายก็จะถูกเพิ่มเข้าไปฐานข้อมูลทรานแซคชั่น ตัวอย่างของเซตเหตุการณ์เช่น

Q = {alter(field, fKeys[], ...)}

สมการข้างต้นแสดงเซตเหตุการณ์ที่ประกอบด้วย 1 รายการการเปลี่ยนแปลงแก้ไข ระบุว่านักพัฒนาได้ทำการเปลี่ยนแปลงเอนทิตี้ระดับตัวแปรชื่อ fKey[] ในแฟ้มข้อมูล ComparePerferencePage.java (ในสมการข้างต้นละการเขียนถึงเอนทิตี้แฟ้มข้อมูล ComparePerferencePage.java เอาไว้โดยแทนที่ด้วย ... เพื่อความสะดวกในการเขียน)

จุดมุ่งหมายในการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ของงานวิจัยนี้คือ การให้คำแนะนำนักพัฒนาว่าควรจะเปลี่ยนแปลงแก้ไขที่ใดต่อไปเมื่อนักพัฒนาทำให้เกิดเหตุการณ์นั้นๆ ขึ้น ในหัวข้อต่อไปนี้จะอธิบายถึงขั้นตอนวิธีในการสร้างกฎความสัมพันธ์จากการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ และอธิบายการสร้างคำแนะนำจากกฎความสัมพันธ์ที่ได้มา

2.8.2.1 การสร้างกฎความสัมพันธ์จากการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์

ขั้นตอนวิธีในการสร้างกฎความสัมพันธ์ที่ได้รับความนิยมมากที่สุดก็คือ ขั้นตอนวิธีอพริโอริ (Apriori algorithm) ที่ถูกนำเสนอโดย Agrawal และ Srikant ในปี ค.ศ. 1994 (Agrawal and Srikant, 1994) ขั้นตอนวิธีอพริโอรินั้นจำเป็นต้องระบุค่าสนับสนุนขั้นต่ำ (Minimum Support) (สำหรับงานวิจัยนี้เป็นค่าสนับสนุนนับขั้นต่ำ (Minimum Support Count) แทน) และค่าความเชื่อมั่นขั้นต่ำ (Minimum Confidence) เพื่อใช้ในการคำนวณผลลัพธ์ของขั้นตอนวิธีอพริโอริคือเซตของกฎความสัมพันธ์ทั้งหมดที่มีค่าสนับสนุนนับมากกว่าค่าสนับสนุนนับขั้นต่ำและมีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นขั้นต่ำ

การนำขั้นตอนวิธีอพริโอริมาประยุกต์ใช้กับการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์โดยคำนวณหากฎความสัมพันธ์ทั้งหมดเอาไว้ก่อนล่วงหน้าและสร้างคำแนะนำที่เหมาะสมกับเหตุการณ์ที่นักพัฒนาสร้างขึ้น แต่วิธีการดังกล่าวสามารถทำให้เกิดปัญหาตามมาได้ เนื่องจากเหตุการณ์ใหม่ๆที่เกิดขึ้นหลังจากการสร้างกฎความสัมพันธ์เก็บไว้แล้วจะไม่ได้ถูกนำไปสร้างเป็นทรานแซคชั่นและนำไปเป็นส่วนหนึ่งของข้อมูลที่ใช้ในการทำเหมืองข้อมูล ดังนั้นเมื่อเวลาผ่านไปเซตของกฎความสัมพันธ์ที่สร้างเอาไว้ล่วงหน้าจะได้ถูกต้องตรงกับความเป็นจริง วิธีที่ถูกต้องก็คือกฎความสัมพันธ์ทั้งหมดจะต้องถูกคำนวณใหม่อยู่เสมอทุกครั้งที่นักพัฒนาทำให้เกิดเหตุการณ์ใหม่ๆ แต่ในความเป็นจริงการคำนวณโดยใช้ขั้นตอนวิธีอพริโอริกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ใช้เวลานานมาก (ในกรณีโครงการใหญ่ๆและผ่านระยะเวลาการพัฒนามานานๆ ต้องใช้เวลาในการคำนวณหลายวัน) ปี ค.ศ. 2005 Zimmermann และคณะได้เสนอวิธีการในปรับปรุงขั้นตอนวิธีอพริ

โอริมาเพื่อให้เหมาะสม (Optimization) กับการทำเหมืองข้อมูลของข้อมูลซอฟต์แวร์อาร์ไคฟว์ โดยการเพิ่มข้อกำหนด 2 ข้อให้กับขั้นตอนวิธีอพริโอริ (Zimmermann et al., 2005) ดังนี้

- การค้นหากฎความสัมพันธ์เฉพาะกฎที่มีเซตรายการที่มาก่อนที่ต้องการเท่านั้น กล่าวคือ การทำเหมืองข้อมูลจะเกิดขึ้นในขณะที่ผู้พัฒนากำลังทำให้เกิดเหตุการณ์ขึ้นและค้นหาเฉพาะกฎความสัมพันธ์ที่มีเซตที่เหตุการณ์ Q ที่นักพัฒนาพึ่งจะกระทำเป็นเซตรายการที่มาก่อนของกฎ (Srikant et al., 1997) การกระทำเช่นนี้ทำให้การคำนวณทั้งหมดใช้เวลาเพียงเล็กน้อยเท่านั้น ข้อดีอีกประการหนึ่งของการใช้การค้นหากฎความสัมพันธ์เฉพาะกฎที่มีเซตรายการที่มาก่อนคือ วิธีการนี้สามารถกระทำได้แบบเพิ่มขึ้นต่อเนื่อง (Incrementally) ได้ จึงทำให้เหตุการณ์ใหม่ๆที่พึ่งการจะถูกนำไปสร้างเป็นทรานแซคชั่นและมีผลต่อการคำนวณหากฎความสัมพันธ์ในครั้งถัดไป
- การกำหนดให้ทุกกฎความสัมพันธ์ที่ค้นหามีเซตรายการที่ตามมาเพียง 1 รายการเท่านั้น การกระทำนี้ทำให้การทำเหมืองข้อมูลเร็วขึ้นกว่าเดิมมาก ดังนั้นกฎความสัมพันธ์ที่ได้หลังจากที่นักพัฒนากระทำให้เกิดเหตุการณ์ Q ก็คือ $Q \rightarrow \{e\}$ โดยที่ e คือสมาชิกของเซตรายการที่ตามมา การกำหนดให้ทุกกฎความสัมพันธ์ที่ค้นหามีเซตรายการที่ตามมาเพียงรายการเดียวนี้ไม่ได้ส่งผลกระทบให้ผลลัพธ์ของคำแนะนำที่จะได้นั้นผิดเพลี้ยนไปเพราะถึงแม้ว่าเซตของคำแนะนำที่ได้จะมีสมาชิกมากกว่า 1 รายการแต่นักพัฒนาก็สามารถกระทำตามคำแนะนำได้เพียงครั้งละ 1 รายการอยู่ดี หรือการพิสูจน์ทางคณิตศาสตร์ดังนี้ กำหนดให้ รายการ $e \in x_2$ ของกฎความสัมพันธ์ $Q \rightarrow x_2[s;c]$ ดังนั้นย่อมมีกฎความสัมพันธ์ r ที่มีเซตรายการที่ตามมารายการเดียวคือ กฎความสัมพันธ์ $Q \rightarrow \{e\}[s_r;c_r]$ ซึ่ง $s_r \geq s$ และ $c_r \geq c$ เนื่องจาก $Q \cup \{e\} \subseteq Q \cup x_2$ ทำให้ จำนวนรายการของ $Q \cup \{e\} \geq Q \cup x_2$

2.8.2.2 การสร้างคำแนะนำจากกฎความสัมพันธ์ (Generating Suggestions for Situation)

การสร้างเซตของคำแนะนำ (Suggestions) จากเซตของกฎความสัมพันธ์ R สำหรับเหตุการณ์ Q ที่นักพัฒนากระทำออกมาสามารถนิยามให้อยู่ในรูปของการยูเนียน (Union) ของเซตรายการที่ตามมาของกฎความสัมพันธ์ R ที่มีเซตรายการที่มาก่อนตรงกับเซตเหตุการณ์ Q ดังนี้

$$apply_R(Q) = \bigcup_{(Q \rightarrow \{x_2\}) \in R} x_2$$

ตัวอย่างการสร้างคำแนะนำจากกฎความสัมพันธ์ {alter(field, fKeys[], ...)} → {alter(method, initDefaults(), ...), alter(file, plug.properties, ...)} สำหรับเหตุการณ์ Q จะทำให้ได้ผลลัพธ์คือเซตของคำแนะนำคือ {alter(method, initDefaults(), ...), alter(file, plug.properties, ...)} โดยที่คำแนะนำการเปลี่ยนแปลงแก้ไขที่อยู่ภายในเซตของคำแนะนำนั้นจะถูกเรียงลำดับตามค่าสนับสนุนนับ และค่าความเชื่อมั่น จำนวนของคำแนะนำภายในเซตของคำแนะนำนี้จะขึ้นอยู่กับการกำหนดค่าสนับสนุนนับ และค่าความเชื่อมั่นขั้นต่ำ โดยปกติแล้วมักจะเริ่มต้นกำหนดค่าสนับสนุนนับขั้นต่ำเป็น 1 และค่าความเชื่อมั่นขั้นต่ำเป็น 0.1

ในงานวิจัยของ Zimmermann และคณะ (Zimmermann et al., 2005) ได้ตั้งข้อสันนิษฐานไว้ว่า การให้คำแนะนำในการเปลี่ยนแปลงแก้ไขกับนักพัฒนานั้น คำแนะนำที่จะได้รับความสนใจจากนักพัฒนาก็คือคำแนะนำที่อยู่ใน 10 อันดับแรกเท่านั้น ดังนั้นในการสร้างเซตของคำแนะนำจึงควรให้ความสนใจกฎความสัมพันธ์ที่อยู่ใน 10 อันดับแรกโดยเรียงจากค่าสนับสนุนนับและค่าความเชื่อมั่นเท่านั้น ดังนั้นผู้วิจัยจึงกำหนดสมการในการสร้างเซตของคำแนะนำดังแสดงในสมการต่อไปนี้

กำหนดให้ q คือ ข้อสอบถาม (Query) ที่ประกอบด้วยเซตเหตุการณ์ (Situation) Q และเซตผลลัพธ์ที่คาดไว้ (Expected Result) E และเขียนให้อยู่ในรูป q = (Q, E)

R คือ เซตของกฎความสัมพันธ์ที่อยู่ในรูปแบบ Q→{x} โดยที่ x คือรายการการเปลี่ยนแปลงแก้ไข และกำหนดให้ R_{10} คือเซตของกฎความสัมพันธ์ที่มีระดับความน่าสนใจสูงสุด 10 กฎแรกซึ่งเรียงลำดับด้วยค่าความเชื่อมั่น โดยที่ $R_{10} \subset R$

A_q คือ เซตของรายการการเปลี่ยนแปลงแก้ไข x ที่ได้จากกฎความสัมพันธ์ใน

เซต R_{10} ที่สอดคล้องกับเซตเหตุการณ์ Q ของข้อสอบถาม q ซึ่งสามารถเขียนในรูป $A_q = apply_{R_{10}}(Q)$ ดังนั้นขนาดของเซต A_q จะน้อยกว่าหรือเท่ากับ 10 เสมอ

$$A_q = apply_{R_{10}}(Q)$$

สมมุติว่าเมื่อนักพัฒนาเห็นเซตของคำแนะนำแล้ว นักพัฒนาจะตัดสินใจดำเนินการเปลี่ยนแปลงแก้ไขตามคำแนะนำแรกเสมอ (คำแนะนำที่มีค่าความเชื่อมั่นสูงที่สุด) เมื่อนักพัฒนากระทำตามคำแนะนำนั้นก็จะทำให้เกิดเหตุการณ์ใหม่ขึ้นและเหตุการณ์ใหม่นั้นก็จะถูกนำไปสร้างเซตของคำแนะนำใหม่จากกฎความสัมพันธ์ที่ได้จากการทำเหมืองข้อมูลกับทรานแซคชั่นทั้งหมดที่รวมเอาเหตุการณ์ครั้งก่อนหน้าเอาไว้ด้วย

ขั้นตอนวิธีที่กล่าวมาทั้งหมดในหัวข้อ 2.8.1 และ 2.8.2 นี้ ได้ถูกนำไปสร้างเป็นระบบให้คำแนะนำสำหรับนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์ชื่อโปรแกรมประยุกต์อีโรส (eROSE) โดย Zimmermann และคณะ (Zimmermann et al., 2005) โปรแกรมประยุกต์อีโรสประกอบด้วย 3 ส่วน คือ 1) ส่วนการจัดเตรียมข้อมูลเพื่อการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ 2) ส่วนการทำเหมืองข้อมูลกับข้อมูลซอฟต์แวร์อาร์ไคฟว์ และ 3) ส่วนการให้คำแนะนำนักพัฒนาระหว่างการพัฒนาซอฟต์แวร์

2.9 การวัดประสิทธิภาพของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์

การทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์นั้นสามารถกระทำได้ในสถานการณ์ที่แตกต่างกัน 3 สถานการณ์ ได้แก่ 1) สถานการณ์การนำทาง (Navigation) คือ สถานการณ์ที่นักพัฒนามีการเปลี่ยนแปลงแก้ไขที่เอนทิตี้หนึ่งแล้ว ระบบจะให้คำแนะนำกับนักพัฒนาให้แก้ไขเอนทิตี้ใดต่อไปได้ถูกต้องหรือไม่ 2) สถานการณ์การป้องกันการเกิดข้อผิดพลาด (Error Prevention) คือ สถานการณ์ที่นักพัฒนามีการเปลี่ยนแปลงแก้ไขที่เอนทิตี้หลายๆเอนทิตี้ต่อเนื่องกันแต่ยังขาดการเปลี่ยนแปลงแก้ไขเอนทิตี้อีกหนึ่งเอนทิตี้จึงจะสมบูรณ์ ระบบจะให้คำแนะนำกับนักพัฒนาให้แก้ไขเอนทิตี้ที่เหลือนั้นได้ถูกต้องหรือไม่ 3) สถานการณ์การ

เปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure) คือ สถานการณ์ที่นักพัฒนามีการเปลี่ยนแปลงแก้ไขที่เอนทิตี้หลายๆเอนทิตี้ต่อเนื่องกันจนสมบูรณ์แล้ว ระบบจะให้คำแนะนำที่เป็นผลบวกลวง (False Positive) เป็นผลลัพธ์แก่นักพัฒนาหรือไม่

การวัดประสิทธิภาพของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ใช้วิธีการวัดประสิทธิภาพเหมือนกับการวัดประสิทธิภาพของการค้นคืนข้อมูล (Information Retrieval) (Zimmermann et al., 2005) ก็คือการคำนวณหาค่าความถูกต้อง (Precision) และค่าเรียกคืน (Recall) ของแต่ละหน่วยทดลอง จากนั้นนำค่าความถูกต้อง (Precision) และค่าเรียกคืน (Recall) ที่ได้มาคำนวณหาค่าเอฟเมสเซอร์ (F-measure) ซึ่งค่าเอฟเมสเซอร์นี้ก็คือค่าประสิทธิภาพของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ในสถานการณ์การนำทาง (Navigation) และสถานการณ์การป้องกันการเกิดข้อผิดพลาด (Error Prevention) นั่นเอง นอกนั้นผู้วิจัยยังสามารถคำนวณค่าผลสะท้อนกลับ (Feedback) หรือค่าร้อยละของข้อสอบถามที่ไม่ได้ให้เซตรายการการเปลี่ยนแปลงแก้ไขที่ถูกดึงขึ้นมาเป็นเซตว่างกับข้อสอบถามทั้งหมด ซึ่งค่าผลสะท้อนกลับนี้ก็คือค่าประสิทธิภาพของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ในสถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้ว (Closure)

การใช้ค่าเอฟเมสเซอร์เป็นค่าที่ใช้การระบุระดับประสิทธิภาพของการค้นคืนข้อมูลเป็นที่นิยมในงานวิจัยที่เกี่ยวข้องกับการค้นคืนข้อมูล เช่น งานวิจัยการขยายคำในข้อสอบถามโดยรวมกฎความสัมพันธ์กับสิ่งที่ศึกษาร่วมกับเทคนิคการค้นคืนสารสนเทศ (Song et al., 2005) และงานวิจัยการจัดกลุ่มเอกสารทางเว็บโดยใช้เซตรายการที่มากที่สุด (Zhuang and Dai, 2004) เป็นต้น นอกจากนั้นงานวิจัยของ Methanias และคณะ (Methanias et al., 2009) ที่ทำการเปรียบเทียบประสิทธิภาพของการทำเหมืองข้อมูลบนข้อมูลซอฟต์แวร์อาร์ไคฟว์ของโครงการซอฟต์แวร์สิ่งแวดล้อมอุตสาหกรรม (Industrial Environment) ก็ใช้ค่าเอฟเมสเซอร์เป็นค่าที่ใช้การระบุระดับประสิทธิภาพเช่นกัน ค่าความถูกต้อง (Precision) ค่าเรียกคืน (Recall) และค่าเอฟเมสเซอร์ (F-measure) นั้นสามารถคำนวณได้ดังนี้

กำหนดให้ q คือ ข้อสอบถาม (Query) ที่ประกอบด้วยเซตเหตุการณ์ (Situation) Q และเซตผลลัพธ์ที่คาดไว้ (Expected Result) E และเขียนให้อยู่ในรูป $q = (Q, E)$

R คือ เซตของกฎความสัมพันธ์ที่อยู่ในรูปแบบ $Q \rightarrow \{x\}$ โดยที่ x คือรายการ

การเปลี่ยนแปลงแก้ไข และกำหนดให้ R_{10} คือเซตของกฎความสัมพันธ์ที่มีระดับความน่าสนใจสูงสุด 10 กฎแรกซึ่งเรียงลำดับด้วยค่าความเชื่อมั่น โดยที่ $R_{10} \subset R$

A_q คือ เซตของรายการการเปลี่ยนแปลงแก้ไข x ที่ได้จากกฎความสัมพันธ์ในเซต R_{10} ที่สอดคล้องกับเซตเหตุการณ์ Q ของข้อสอบถาม q ซึ่งสามารถเขียนในรูป $A_q = apply_{R_{10}}(Q)$ ดังนั้นขนาดของเซต A_q จะน้อยกว่าหรือเท่ากับ 10 เสมอ

2.9.1 ค่าความถูกต้อง (Precision)

ค่าความถูกต้อง (Precision) เป็นสัดส่วนรายการการเปลี่ยนแปลงแก้ไขที่ถูกดึงขึ้นมาแล้วตรงกับรายการการเปลี่ยนแปลงแก้ไขที่อยู่ในเซตผลลัพธ์ที่คาดไว้ เทียบกับรายการการเปลี่ยนแปลงแก้ไขทั้งหมดที่ถูกดึงขึ้น ดังสมการต่อไปนี้ (Zimmermann et al., 2005)

กำหนดให้ $|A_q \cap E|$ คือ จำนวนรายการการเปลี่ยนแปลงแก้ไขที่ถูกดึงขึ้นมาแล้วตรงกับรายการการเปลี่ยนแปลงแก้ไขที่อยู่ในเซตผลลัพธ์ที่คาดไว้

$|A_q|$ คือ จำนวนรายการการเปลี่ยนแปลงแก้ไขที่ถูกดึงขึ้นมา

$$precision = \frac{|A_q \cap E|}{|A_q|}$$

ในกรณีที่เซตรายการการเปลี่ยนแปลงแก้ไขที่ถูกดึงขึ้นมาเป็นเซตว่าง ($|A_q| = 0$) จะกำหนดให้ค่าความถูกต้องมีค่าเท่ากับ 1

2.9.2 ค่าเรียกคืน (Recall)

ค่าเรียกคืน (Recall) เป็นสัดส่วนรายการการเปลี่ยนแปลงแก้ไขที่ถูกดึงขึ้นมาแล้วตรงกับรายการการเปลี่ยนแปลงแก้ไขที่อยู่ในเซตผลลัพธ์ที่คาดไว้ เทียบกับรายการการเปลี่ยนแปลงแก้ไขที่อยู่ในเซตผลลัพธ์ที่คาดไว้ ดังสมการต่อไปนี้ (Zimmermann et al., 2005)

กำหนดให้ $|A_q \cap E|$ คือ จำนวนรายการการเปลี่ยนแปลงแก้ไขที่ถูกดึงขึ้นมาแล้วตรงกับรายการการเปลี่ยนแปลงแก้ไขที่อยู่ในเซตผลลัพธ์ที่คาดไว้

|E| คือ จำนวนรายการการเปลี่ยนแปลงแก้ไขที่อยู่ในเซตผลลัพธ์ที่คาดไว้

$$recall = \frac{|A_q \cap E|}{|E|}$$

ในกรณีที่เซตผลลัพธ์ที่คาดไว้เป็นเซตว่าง (|E| = 0) จะกำหนดให้ค่าความถูกต้องมีค่าเท่ากับ 1

2.9.3 ค่าเอฟเมสเซอร์ (F-measure)

ค่าเอฟเมสเซอร์ (F-measure) หรือค่าเอฟเมสเซอร์แบบถ่วงน้ำหนัก (Weighted Harmonic mean of Precision and Recall) ถูกเสนอขึ้นโดย Rijsbergen ในปี ค.ศ. 1979 (Rijsbergen, 1979) คือ ค่าที่ใช้ในการประเมินความถูกต้องแม่นยำ (accuracy) ของการทดสอบ ซึ่งพิจารณาจากทั้งค่าความถูกต้อง (Precision) และค่าเรียกคืน (Recall) ของการทดสอบโดยมีการถ่วงน้ำหนักให้กับค่าทั้งสองด้วย ค่าเอฟเมสเซอร์มีพิสัยอยู่ระหว่าง 0 กับ 1 ค่าเอฟเมสเซอร์ที่เท่ากับ 1 แสดงว่ามีความถูกต้องแม่นยำมากที่สุด ส่วนค่าเอฟเมสเซอร์ที่เท่ากับ 0 แสดงว่ามีความถูกต้องแม่นยำน้อยที่สุด สมการรูปทั่วไปของค่าเอฟเมสเซอร์แสดงได้ดังต่อไปนี้ (Tsunenori, 2003)

กำหนดให้ F คือ ค่าเอฟเมสเซอร์ (F-measure)

Precision คือ ค่าความถูกต้อง

Recall คือ ค่าเรียกคืน

β คือ ค่าอัตราส่วนน้ำหนักของค่าความถูกต้องต่อค่าเรียกคืน

$$F_\beta = \frac{(1+\beta^2) * precision * recall}{\beta^2 * precision + recall}$$

โดยที่ ค่า β หรือค่าอัตราส่วนน้ำหนักของค่าความถูกต้องต่อค่าเรียกคืนเป็นจำนวนจริงบวก ตัวอย่างของการกำหนดค่า เช่น ค่า β = 1 หมายความว่าให้น้ำหนักของค่าความถูกต้องกับค่าเรียกคืนมีน้ำหนักความสำคัญเท่ากัน ค่า β = 2 หมายถึงกำหนดให้ค่าความถูกต้องมีน้ำหนักความสำคัญเป็นสองเท่าเมื่อเทียบกับค่าเรียกคืน และค่า β = 0.5 หมายถึงกำหนดให้ค่าเรียกคืนมีน้ำหนักความสำคัญเป็นสองเท่าเมื่อเทียบกับค่าความถูกต้อง เป็นต้น สำหรับที่กำหนดให้ β = 1 นั้นค่าเอฟเมสเซอร์จะถูกเรียกว่า ค่าบาลานซ์เอฟเมสเซอร์ (balanced F-score) หรือค่าเอฟหนึ่ง

สกอร์ (F_1 score) หรือก็คือค่าเอฟเมสเซอร์ที่ถ่วงน้ำหนักอย่างสมดุลนั่นเอง สมการของค่าเอฟหนึ่งสกอร์แสดงได้ดังนี้ (Tsunenori, 2003)

กำหนดให้ F_1 คือ ค่าเอฟหนึ่งสกอร์ (F_1 score)

Precision คือ ค่าความถูกต้อง

Recall คือ ค่าเรียกคืน

$$F_1 = \frac{2 * precision * recall}{precision + recall}$$

ค่าเอฟเมสเซอร์เป็นค่าที่นิยมใช้ในการวัดประสิทธิภาพของการค้นคืนข้อมูล (Information Retrieval) อย่างแพร่หลาย (Tsunenori, 2003) งานวิจัยที่เกี่ยวข้องกับการค้นหากฎความสำคัญหลายงานวิจัย (Beil et al., 2002; Geyer-Schulz and Hahsler, 2002) แนะนำให้ใช้ค่าเอฟหนึ่งสกอร์หรือค่าเอฟเมสเซอร์ที่ให้น้ำหนักของค่าความถูกต้องและค่าเรียกคืนอย่างสมดุล

ในปี 2005 งานวิจัยของ Zimmermann และคณะ (Zimmermann et al., 2005)) ทำการเปรียบเทียบประสิทธิภาพของการทำเหมืองข้อมูลบนข้อมูลซอฟต์แวร์อาร์ไคฟ์กับโครงการพัฒนาระบบปฏิบัติการคอมพิวเตอร์ต่างๆ (Operating system) และกล่าวว่า จุดมุ่งหมายของการทดสอบระบบให้คำแนะนำนักพัฒนาในระหว่างการพัฒนาซอฟต์แวร์คือค่าความถูกต้องที่สูง (ค่าใกล้เคียง 1) และค่าเรียกคืนที่สูง (ค่าใกล้เคียง 1) นั่นคือต้องการให้ระบบสามารถแนะนำคำแนะนำทั้งหมด (ค่าเรียกคืนเท่ากับ 1) และแต่ละคำแนะนำนั้นถูกต้องหรือตรงกับเซตผลลัพธ์ที่คาดไว้ทั้งหมด (ค่าความถูกต้องเท่ากับ 1) ดังนั้นงานวิจัยของ Zimmermann และคณะจึงใช้ค่าเฉลี่ยฮาร์โมนิคของค่าความถูกต้องและค่าเรียกคืน (Harmonic mean of Precision and Recall) หรือก็คือค่าเอฟเมสเซอร์ที่ให้น้ำหนักของค่าความถูกต้องและค่าเรียกคืนอย่างสมดุล เป็นค่าประเมินประสิทธิภาพของการทำเหมืองข้อมูล (Zimmermann et al., 2005)

ต่อมาในปี 2009 งานวิจัยของ Methanias และคณะ (Methanias et al., 2009) ทำการเปรียบเทียบประสิทธิภาพของการทำเหมืองข้อมูลบนข้อมูลซอฟต์แวร์อาร์ไคฟ์ของโครงการซอฟต์แวร์สิ่งแวดล้อมอุตสาหกรรม (Industrial Environment) ใน 3 สถานการณ์เช่นเดียวกัน และแนะนำให้ใช้ค่าเอฟเมสเซอร์ที่ให้น้ำหนักของค่าความถูกต้องและค่าเรียกคืนอย่างสมดุลสำหรับการประเมินประสิทธิภาพในสถานการณ์การนำทางและสถานการณ์การป้องกันข้อผิดพลาด (Methanias et al., 2009)

2.9.4 ค่าผลสะท้อนกลับ (Feedback)

ค่าผลสะท้อนกลับ (Feedback) คือ ค่าร้อยละของเซตรายการการเปลี่ยนแปลงแก้ไขที่ถูกดึงขึ้นมาที่ไม่ใช่เซตว่างกับเซตรายการการเปลี่ยนแปลงแก้ไขที่ถูกดึงขึ้นมาทั้งหมด ค่าผลสะท้อนกลับแสดงได้ดังสมการต่อไปนี้ (Zimmermann et al., 2005)

กำหนดให้ $feedbac$ คือ ค่าผลสะท้อนกลับ

|Z*| คือ จำนวนข้อสอบถามที่อยู่ในเซตของข้อสอบถามที่มีเซตของคำแนะนำที่ไม่เป็นเซตว่าง โดยที่ Z* = {q | q = (Q, E) ∈ Z, $apply_{R_{10}}(Q) \neq \emptyset$}

|Z| คือ จำนวนข้อสอบถามทั้งหมด โดยที่ Z = {q | q = (Q, E) }

$$feedback = \frac{|Z^*|}{|Z|}$$

เมื่อนำค่าผลสะท้อนกลับมาใช้ประเมินในสถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้วจะสามารถแสดงให้เห็นถึงร้อยละของการเกิดการแจ้งเตือนที่ผิด (False Alarm) หรือการให้คำแนะนำที่เป็นผลบวกลวงนั่นเอง เนื่องจากค่าผลสะท้อนกลับมีพิสัยอยู่ระหว่าง 0 กับ 1 ค่าผลสะท้อนกลับที่มีค่าเท่ากับ 0 หมายถึงไม่มีข้อสอบถามใดเลยที่ให้คำแนะนำที่เป็นผลบวกลวงออกมาในสถานการณ์นี้นั่นคือมีประสิทธิภาพดีที่สุด และค่าผลสะท้อนกลับที่มีค่าเท่ากับ 1 หมายถึงข้อสอบถามทั้งหมดให้คำแนะนำที่เป็นผลบวกลวงออกมานั่นคือมีประสิทธิภาพไม่ดีที่สุด ดังนั้นการวัดประสิทธิภาพของการทำเหมืองข้อมูลด้วยเทคนิคการค้นหากฎความสัมพันธ์กับข้อมูลซอฟต์แวร์อาร์ไคฟว์ในสถานการณ์การเปลี่ยนแปลงแก้ไขที่สมบูรณ์แล้วนั้นจะต้องเปรียบเทียบค่าผลสะท้อนกลับและค่าผลสะท้อนกลับที่น้อยกว่าจะมีความหมายว่ามีประสิทธิภาพดีกว่า (Zimmermann et al., 2005)