บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

เมื่อต้องการเปิดเผยข้อมูลใดๆ สู่สาธารณะ ข้อมูลเหล่านั้นอาจต้องทำการปกปิดข้อมูลเพื่อปกป้องความเป็นส่วนตัวของข้อมูลก่อน โดยการปกปิดข้อมูลนั้นสามารถทำได้หลายรูปแบบหนึ่งในนั้นคือการแปลงข้อมูลให้มีคุณสมบัติ *k*-Anonymity แต่วิธีการทำให้ข้อมูลมีคุณสมบัติ *k*-Anonymity มีอยู่หลายวิธี ในวิทยานิพนธ์ฉบับนี้สนใจขั้นตอนวิธี MCCRT (Minimum Classification Correction Rate Transformation algorithm) เนื่องจากขั้นตอนวิธีนี้ให้ผลลัพธ์ที่เหมาะสมกับการนำไปใช้งานในการจำแนกแบบความสัมพันธ์ (Associative Classification) และขั้นตอนวิธี MCCRT นี้ยังมีประสิทธิภาพกับประสิทธิผลที่ดีเมื่อเทียบกับการหาแบบ Exhaustive Search ซึ่งใช้เวลาในการประมวลผลที่มากกว่า

2.1 เทคนิค *k*-Anonymity

ในการเผยแพร่ข้อมูลหลังจากปกปิดหรือลบคอลัมน์ที่เป็นตัวบ่งชี้บุคคล (Identifier) ตัวข้อมูลอาจเหลือกลุ่มของคอลัมน์ที่เป็นตัวบ่งชี้ข้อมูลบุคคลทางอ้อม (Quasi-identifier) ซึ่งสามารถนำไปใช้ในการระบุตัวบุคคลอีกครั้งได้ (Re-Identified) ปัญหาของการระบุตัวบุคคลอีกครั้งจะเกิดขึ้นได้ถ้าข้อมูลที่ถูกระบุตัวบุคคลอีกครั้งมีความเป็นเอกลักษณ์ (ระเบียนที่กลุ่มของคอลัมน์ที่เป็นตัวบ่งชี้บุคคลทางอ้อมมีค่าของข้อมูลไม่ซ้ำกับระเบียนอื่น) กล่าวคือมีระเบียนที่ตัวบ่งชี้บุคคลทางอ้อมมีค่าไม่เหมือนระเบียนใดๆในฐานข้อมูลนั้นๆ ระเบียนเหล่านี้ถูกนำไปตรวจสอบกับข้อมูลจากแหล่งอื่นได้ แต่ถ้าระเบียนเหล่านี้มีการซ้ำกันจะทำให้ไม่สามารถระบุตัวบุคคลอีกครั้งได้

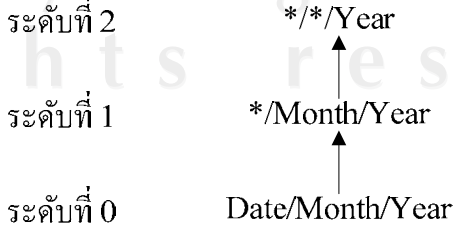
หลักการแปลงข้อมูลให้มีคุณสมบัติ *k*-Anonymity [3] ก็คือการแปลงข้อมูลในแต่ละระเบียนให้ทุกระเบียนมีค่าของตัวบ่งชี้บุคคลทางอ้อมเหมือนกันอย่างน้อย *k* ระเบียน ในการแปลงข้อมูลนั้น สามารถกำหนดขั้นตอนการแปลงข้อมูลตามลำดับขั้น (Hierarchy) ของแต่ละคอลัมน์ไว้ล่วงหน้าโดยอาจกำหนดมาจากผู้เชี่ยวชาญหรือจากการพิจารณาตามความเหมาะสมตามลักษณะค่าของข้อมูล แล้วจึงทำการเลือกกลุ่มคอลัมน์และระดับของขั้นตอนการแปลงข้อมูลตามลำดับขั้นของคอลัมน์เหล่านั้นมาทำการแปลงข้อมูลในแต่ละระเบียน ซึ่งเมื่อทำการแปลงข้อมูลแล้ว ข้อมูลทุก

ระเบียนต้องมีคุณสมบัติ *k*-Anonymity ต่อไปนี้เป็นตัวอย่างข้อมูลและขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้น

ตารางที่ 2.1 ตัวอย่างข้อมูลการวินิจฉัยโรค

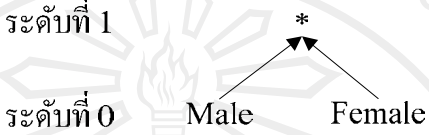
Tuple-ID	BirthDate	Sex	Weight	Height	Career	Diag
1	14/2/2520	Female	48	160	B1	Flu
2	28/2/2520	Female	49	156	B1	Flu
3	19/5/2520	Male	55	161	B1	Fever
4	31/5/2520	Male	55	169	B1	Fever
5	26/3/2522	Female	60	162	A2	Flu
6	26/3/2522	Female	56	161	A2	Fever
7	29/3/2522	Male	51	172	A3	Flu2009
8	30/3/2522	Male	52	175	A3	Flu2009

ตารางที่ 2.1 เป็นตารางตัวอย่างข้อมูลของการวินิจฉัยโรค มีคอลัมน์ Diag เป็นผลการวินิจฉัยและเป็นคอลัมน์คลาส ตารางนี้ถูกปกปิดตัวบ่งชี้บุคคลแล้วโดยการลบคอลัมน์ Name และ Address แต่ยังเหลือตัวบ่งชี้บุคคลทางอ้อมได้แก่คอลัมน์ BirthDate, Sex, Weight, High และ Career ในการกำหนดขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้น (Hierarchy) ของแต่ละคอลัมน์นั้นสามารถพิจารณาจากชนิดหรือรูปแบบของค่าข้อมูลของคอลัมน์นั้นๆได้ เช่น ข้อมูลที่อยู่ในรูปแบบของวันที่ กำหนดให้ไม่แสดงวันที่ในระดับที่ 1 และไม่แสดงวันที่และเดือนในระดับที่ 2 ดังรูปที่ 2.1 หากข้อมูลเป็นเพียงค่า 2 ค่า กำหนดให้ใช้เครื่องหมาย “*” แทนในระดับที่ 1 ดังรูปที่ 2.2 หากข้อมูลเป็นค่าตัวเลขอาจกำหนดให้เป็นช่วงของตัวเลขโดยใช้ช่วงที่กว้างขึ้นในระดับที่สูงขึ้นดังรูปที่ 2.3 และ 2.4 หรือหากข้อมูลเป็นชุดตัวอักษรอาจใช้การแทนตัวอักษรทีละตัวด้วยเครื่องหมาย “*” ดังรูปที่ 2.5 เป็นต้น



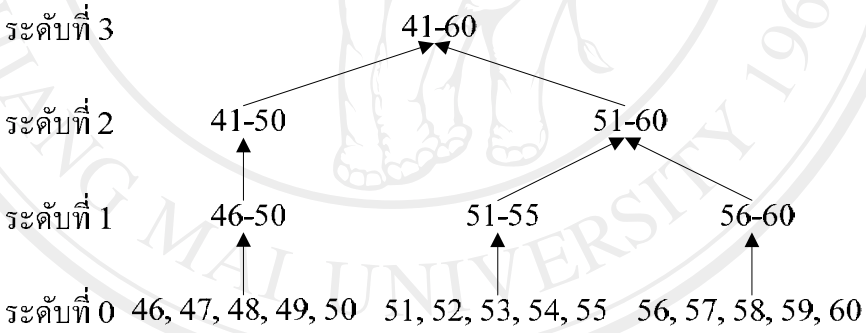
รูปที่ 2.1 ขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ BirthDate

รูปที่ 2.1 เป็นขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ BirthDate ซึ่งเป็นคอลัมน์ที่มีค่าของข้อมูลอยู่ในรูปแบบวันที่ โดยกำหนดให้ไม่แสดงค่าบางส่วนเพิ่มขึ้นในระดับที่สูงขึ้นในที่นี้ให้ไม่แสดงวันในระดับที่ 1 และไม่แสดงเดือนในระดับที่ 2 ในการกำหนดขั้นตอนการแปลงค่าตามลำดับขั้นของวันที่นั้น ยังสามารถทำได้โดยการแทนค่าในรูปแบบอื่น เช่นการแทนค่าของวันที่ให้เป็นสัปดาห์ที่แทน หรือแทนที่เดือนด้วยเลขที่ไตรมาสเป็นต้น



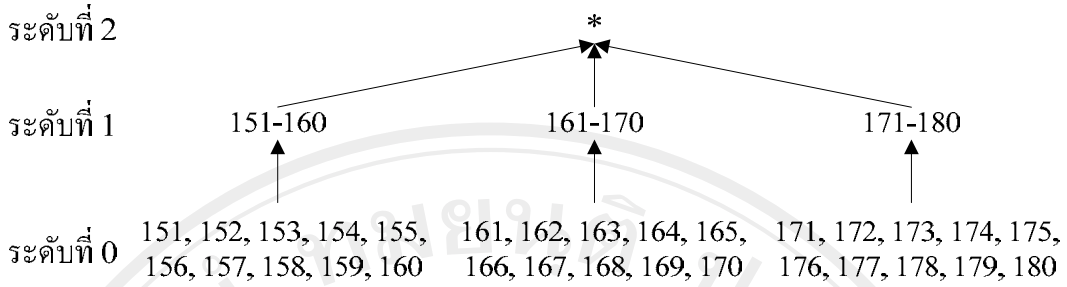
รูปที่ 2.2 ขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ Sex

รูปที่ 2.2 เป็นขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ Sex เป็นคอลัมน์ที่มีค่าเพียงสองค่าคือ Male กับ Female ในกรณีนี้สามารถกำหนดทำการแปลงค่าโดยแปลงเป็นค่าที่ไม่รู้จัก (Unknown Value) หรือ “*” ในระดับที่ 1



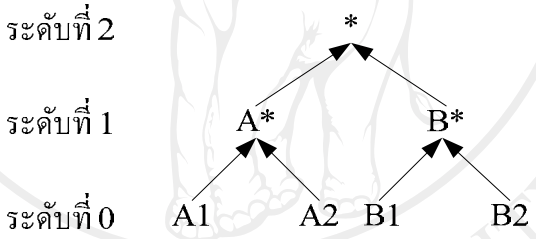
รูปที่ 2.3 ขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ Weight

รูป 2.3 เป็นขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ Weight เป็นคอลัมน์ที่มีค่าของข้อมูลเป็นตัวเลข โดยทำการกำหนดขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นด้วยวิธีการแบ่งช่วงให้กับค่าของตัวเลข ให้ช่วงของตัวเลขมีขนาดที่กว้างขึ้นในระดับที่สูงขึ้น กล่าวคือในระดับที่ 1 มีค่าความกว้างของช่วงเท่ากับ 5 ในระดับที่ 2 มีค่าความกว้างของช่วงเท่ากับ 10 และระดับที่ 3 มีค่าความกว้างของช่วงเท่ากับ 20



รูปที่ 2.4 ขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ Height

รูป 2.4 เป็นขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ Height จะมีการกำหนดขั้นตอนการแปลงค่าตามลำดับขั้นคล้ายกับคอลัมน์ Weight แต่ที่ระดับบนสุดคือระดับที่ 2 ได้กำหนดให้ค่าเป็น “*” ซึ่งการกำหนดค่าของระดับบนสุดเป็น “*” นี้ สามารถกล่าวได้ว่าคือการลบค่าของคอลัมน์นี้ออกไปจะแตกต่างกับที่ระดับบนสุดที่ยังบ่งบอกค่าที่มีความหมายไว้ แต่ลักษณะทางกายภาพไม่แตกต่างกันเพราะค่าที่ระดับบนสุดยังเป็นค่าที่มีค่าเดียว



รูปที่ 2.5 ขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ Career

รูป 2.5 เป็นขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของคอลัมน์ Career มีค่าเป็นตัวอักษรผสมกับตัวเลขโดยตัวอักษรบ่งบอกถึงกลุ่มของอาชีพส่วนตัวเลขเป็นตัวบ่งบอกถึงรายละเอียดเช่น A บ่งบอกว่าเป็นกลุ่มอาชีพข้าราชการ ส่วนตัวเลขจะบ่งบอกว่าเป็นตำแหน่งอะไร เช่น 1 เป็นอาจารย์ เป็นต้น ในลักษณะนี้สามารถกำหนดขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นโดยการปกปิดตัวอักษรทีละตัว โดยใช้เครื่องหมาย “*” แทนค่าตัวอักษร 1 ตัวและแทนค่าตัวอักษรมากขึ้นในระดับที่สูงขึ้น

หลังจากการกำหนดขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นของทุกคอลัมน์แล้ว จะทำการแปลงค่าข้อมูลไปตามระดับที่กำหนดไว้ที่ละระเบียน ระดับการแปลงค่าของข้อมูลแต่ละระเบียนนี้เรียกว่า ระดับการเจนเนอรัลไลเซชัน (Generalization Level : *GL*) สามารถเขียนแทนด้วย <(BirthDate, 1), (Sex, 0), (Weight, 0),(Height, 0), (Career, 0)> โดยในวิทยานิพนธ์นี้ใช้

วิธีการแปลงค่าข้อมูลแบบ Full Domain ดังที่ [4] ได้นิยามไว้ คือการแปลงค่าของข้อมูลทุกระเบียนของแต่ละคอลัมน์ให้อยู่ในระดับเดียวกันตามที่กำหนดไว้ในระดับการเจนเนอรัลไลเซชัน *GL* กล่าวคือทุกระเบียนใช้ระดับการเจนเนอรัลไลเซชัน *GL* เดียวกัน ตัวอย่างเช่นการแปลงค่าของข้อมูลในคอลัมน์ BirthDate ให้อยู่ในระดับที่ 1 ของขั้นตอนการแปลงค่าตามลำดับขั้น กล่าวคือ ระดับการเจนเนอรัลไลเซชัน *GL* เท่ากับ <(BirthDate, 1), (Sex, 0), (Weight, 0),(Height, 0), (Career, 0)> จะได้ ข้อมูลใหม่ออกมาตามตัวอย่างตารางที่ 2.2

ตารางที่ 2.2 ตารางข้อมูลหลังการแปลงค่าโดยระดับการเจนเนอรัลไลเซชัน *GL* เท่ากับ <(BirthDate, 1), (Sex, 0), (Weight, 0),(Height, 0), (Career, 0)>

Tuple-ID	BirthDate	Sex	Weight	Height	Career	Diag
1	*/2/2520	Female	48	160	B1	Flu
2	*/2/2520	Female	49	156	B1	Flu
3	*/5/2520	Male	55	161	B1	Fever
4	*/5/2520	Male	55	169	B1	Fever
5	*/3/2522	Female	60	162	A2	Flu
6	*/3/2522	Female	56	161	A2	Fever
7	*/3/2522	Male	51	172	A3	Flu2009
8	*/3/2522	Male	52	175	A3	Flu2009

ตารางที่ 2.3 ตารางข้อมูลหลังการเปลี่ยนแปลงค่าซึ่งมีคุณสมบัติ *2*-Anonymity โดย *GL* เท่ากับ <(BirthDate, 1), (Sex, 1), (Weight, 1),(Height, 1), (Career, 0)>

Tuple-ID	BirthDate	Sex	Weight	Height	Career	Diag
1	*/2/2520	*	46-50	151-160	B1	Flu
2	*/2/2520	*	46-50	151-160	B1	Flu
3	*/5/2520	*	51-55	161-170	B1	Fever
4	*/5/2520	*	51-55	161-170	B1	Fever
5	*/3/2522	*	56-60	161-170	A2	Flu
6	*/3/2522	*	56-60	161-170	A2	Fever

ตารางที่ 2.3 ตารางข้อมูลหลังการเปลี่ยนแปลงค่าซึ่งมีคุณสมบัติ *2*-Anonymity โดย *GL* เท่ากับ <(BirthDate, 1), (Sex, 1), (Weight, 1),(Height, 1), (Career, 0)> (ต่อ)

7	*/3/2522	*	51-55	171-180	A3	Flu2009
8	*/3/2522	*	51-55	171-180	A3	Flu2009

ตารางที่ 2.3 เป็นตารางข้อมูลหลังการแปลงค่าของข้อมูลโดยใช้ค่าระดับเจนเนอรัลไลเซชัน *GL* เท่ากับ <(BirthDate, 1), (Sex, 1), (Weight, 1),(Height, 1), (Career, 0)> ซึ่งตารางข้อมูลนี้มีคุณสมบัติของ *2*-Anonymity โดยทุกระเบียนมีจะมีระเบียนอื่นที่ค่าของข้อมูลเหมือนกันรวมระเบียนตัวเองอย่างน้อย 2 ระเบียนซึ่งตารางที่ 2.3 และ 2.4 ได้แรเงากลุ่มระเบียนที่มีค่าเหมือนกันไว้เพื่อง่ายต่อการพิจารณา ทุกกลุ่มระเบียนจะมีจำนวน 2 ระเบียน เมื่อพิจารณาตารางที่ 2.3 จะเห็นว่ามีลักษณะที่คล้ายกับตารางที่ 2.4 แต่ค่าของระเบียนจะแตกต่างไปเนื่องจากเป็นค่าที่มาจากการใช้ระดับการเจนเนอรัลไลเซชัน *GL* คนละตัวในการแปลงค่าของข้อมูล

ตารางที่ 2.4 ตารางข้อมูลหลังการเปลี่ยนแปลงค่าซึ่งมีคุณสมบัติ *2*-Anonymity โดย *GL* เท่ากับ <(BirthDate, 1), (Sex, 0), (Weight, 1),(Height, 1), (Career, 1)>

Tuple-ID	BirthDate	Sex	Weight	Height	Career	Diag
1	*/2/2520	Female	46-50	151-160	B*	Flu
2	*/2/2520	Female	46-50	151-160	B*	Flu
3	*/5/2520	Male	51-55	161-170	B*	Fever
4	*/5/2520	Male	51-55	161-170	B*	Fever
5	*/3/2522	Female	56-60	161-170	A*	Flu
6	*/3/2522	Female	56-60	161-170	A*	Fever
7	*/3/2522	Male	51-55	171-180	A*	Flu2009
8	*/3/2522	Male	51-55	171-180	A*	Flu2009

ในการหาระดับการเจนเนอรัลไลเซชัน *GL* ที่ทำให้ข้อมูลมีคุณสมบัติ *k*-Anonymity นั้น เมื่อกำหนดค่า *k* แล้วจะเห็นว่าต้องทำการทดสอบระดับการเจนเนอรัลไลเซชัน *GL* ทีละค่าว่าข้อมูลใหม่ที่ได้จากระดับการเจนเนอรัลไลเซชัน *GL* ดังกล่าวมีคุณสมบัติ *k*-Anonymity หรือไม่และมัก

ปรากฏว่าข้อมูลหนึ่งๆสามารถมีคุณสมบัติ *k*-Anonymity ได้จากระดับการเจนเนอรัลไลเซชัน *GL* หลายค่า เช่นตัวอย่างตารางที่ 2.3 และตารางที่ 2.4 จะเห็นว่าตารางเหล่านี้มีคุณสมบัติของ *k*-Anonymity โดยที่ *k* เท่ากับ 2 แต่มีค่าของข้อมูลในแต่ละระเบียนต่างกันเมื่อเทียบตาม Tuple-ID เนื่องจากข้อมูลทั้ง 2 ตารางมาจากการใช้ระดับการเจนเนอรัลไลเซชัน *GL* ที่ต่างกัน ฉะนั้นการเผยแพร่ข้อมูลต้องเลือกว่าควรใช้ข้อมูลใหม่จากการแปลงค่าโดยระดับการเจนเนอรัลไลเซชัน *GL* ค่าใดที่เหมาะสมที่สุด

2.2 เทคนิค (α , *k*)-Anonymity

จากตารางที่ 2.4 จะเห็นว่าในกลุ่ม Tuple-ID {5, 6} นั้นมีค่าในคอลัมน์คลาสซึ่งในที่นี้คือคอลัมน์ Diag อยู่สองค่าคือ Flu และ Fever แต่ในกลุ่ม Tuple-ID {1, 2}, Tuple-ID {3, 4} และ Tuple-ID {7, 8} มีค่าของคอลัมน์คลาสในแต่ละกลุ่มเพียงค่าเดียว ถ้ามีการเทียบค่าจากตารางข้อมูลได้เป็น 3 กลุ่มหลังจะทำให้ทราบค่าคลาสได้ ยกตัวอย่างเช่น ถ้ามีข้อมูลจากแหล่งอื่นให้ค่าว่านายสมชาย เกิดวันที่ 29/3/2522 น้ำหนัก 51 กิโลกรัม ส่วนสูง 172 และมีอาชีพเป็นข้าราชการ ถ้าทราบว่านายสมชายเคยมารับการรักษาที่โรงพยาบาลที่ทำการเผยแพร่ตารางข้อมูล 2.4 นี้ ทำให้สามารถทำข้อมูลนายสมชายมาตรวจสอบกับตารางข้อมูล 2.4 ซึ่งจะได้ว่านายสมชายมีข้อมูลเหมือนกับระเบียนที่ Tuple-ID เท่ากับ 7 และ 8 ถึงแม้จะไม่อาจแน่ใจได้ว่านายสมชายมี Tuple-ID เท่าไหร่แต่จะเห็นว่าทั้ง 2 ระเบียนนั้นเป็นโรคเดียวกันคือ Flu2009 ซึ่งแสดงให้เห็นว่านายสมชายถูกวินิจฉัยว่าเป็นโรค Flu2009

จากปัญหานี้จึงได้มีการคิดค้นและนำเสนอรูปแบบ (α , *k*)-Anonymity ขึ้นมา โดยนอกเหนือจากคุณสมบัติของ *k* แล้วยังเพิ่มคุณสมบัติของค่า α ขึ้นมาซึ่งค่านี้เป็นตัวเลขจำนวนจริงที่มีค่าอยู่ระหว่าง 0 ถึง 1 โดยค่า α นี้ใช้เป็นตัวกำหนดความถี่ของการซ้ำกันของค่าคลาสเฝ้าระวังในแต่ละกลุ่มโดย จำนวนระเบียนในกลุ่มที่คู่กับค่าคลาสเฝ้าระวังหารด้วยจำนวนระเบียนในกลุ่มต้องมีค่าน้อยกว่าหรือเท่ากับ α

2.3 ขั้นตอนวิธี MCCRT

การแปลงค่าของข้อมูลเพื่อให้ข้อมูลมีคุณสมบัติ *k*-Anonymity ทำให้คุณภาพของข้อมูลลดลง ดังนั้นจึงมีการนำตัววัดคุณภาพข้อมูลเข้ามาใช้ เพื่อวัดว่าข้อมูลหลังการแปลงค่าข้อมูลจากระดับเจนเนอรัลไลเซชันค่าใดมีความเหมาะสมที่สุด แต่การเลือกใช้ตัววัดคุณภาพของข้อมูลต้องพิจารณาจากงานที่จะนำข้อมูลหลังการแปลงข้อมูลไปใช้ ตัววัดคุณภาพข้อมูลจึงต้องเหมาะสมกับ

งานดังกล่าว โดย [3] ได้ใช้ค่า C_{FCM} เป็นตัววัดคุณภาพของข้อมูล ซึ่งค่า C_{FCM} ของข้อมูลใดๆมีค่าน้อยแสดงว่าข้อมูลนั้นมีความบิดเบือนน้อยและมีความเหมาะสมกับงาน Associative Classification มาก

ในการหาระดับการเจนเนอรัลไลเซชัน *GL* ที่ใช้ในการแปลงค่าข้อมูลเพื่อให้ข้อมูลมีคุณสมบัติ *k*-Anonymity และมีค่า C_{FCM} น้อยนั้น ถ้าใช้ขั้นตอนวิธีการค้นหาแบบละเอียด (Exhaustive Search) ต้องหาระดับการเจนเนอรัลไลเซชัน *GL* ที่เป็นไปได้ทั้งหมด แล้วทำการทดสอบว่าข้อมูลหลังการแปลงตามระดับการเจนเนอรัลไลเซชัน *GL* ค่าใดบ้างที่มีคุณสมบัติ *k*-Anonymity แล้วจึงนำข้อมูลดังกล่าวไปหาค่า C_{FCM} และสุดท้ายจะทำการเลือกข้อมูลหลังการแปลงที่มีค่า C_{FCM} น้อยที่สุด ปัญหาในการแปลงค่าข้อมูลเพื่อให้ข้อมูลมีคุณสมบัติ *k*-Anonymity และเหมาะสมกับงาน Associative Classification ที่สุดนี้เป็นปัญหาแบบยาก (NP-Hard) ใน [3] จึงได้คิดขั้นตอนวิธี MCCRT ซึ่งใช้แนวทางศึกษาสำนึก (Heuristic) ในการแก้ปัญหาซึ่งขั้นตอนวิธีนี้มีประสิทธิภาพที่ดีกว่าขั้นตอนวิธีการค้นหาแบบละเอียดและให้ประสิทธิผลที่ใกล้เคียงกับขั้นตอนวิธีการค้นหาแบบละเอียด

ขั้นตอนวิธี MCCRT เริ่มต้นด้วยการนำข้อมูลที่ต้องการแปลงค่ามาคำนวณค่าอัตราความถูกต้องในการจำแนก *CCR* (Classification Correction Rate) ของแต่ละคอลัมน์ จากนั้นทำการเพิ่มค่าระดับของคอลัมน์ในระดับการเจนเนอรัลไลเซชัน *GL* ทีละคอลัมน์ขึ้นไปจนถึงระดับบนสุด โดยเรียงจากคอลัมน์ที่มีค่าอัตราความถูกต้องในการจำแนก *CCR* น้อยไปหาอัตราความแม่นยำในการจำแนก *CCR* มาก ในกรณีที่มีค่าอัตราความถูกต้องในการจำแนก *CCR* เท่ากันให้เลือกเพิ่มระดับกับคอลัมน์ที่มีค่าความสูงของขั้นตอนการแปลงค่าข้อมูลตามลำดับขั้นมากกว่าก่อน ทำการเพิ่มระดับไปจนพบระดับการเจนเนอรัลไลเซชัน *GL* ที่ทำให้ข้อมูลหลังการแปลงมีคุณสมบัติ *k*-Anonymity

ค่าอัตราความถูกต้องในการจำแนก *CCR* ของคอลัมน์ใดๆ เป็นค่าที่แสดงค่าความแม่นยำในการบ่งบอกค่าคลาสจากค่าข้อมูลในคอลัมน์นั้นๆ การคำนวณหาค่าอัตราความถูกต้องในการจำแนก *CCR* ของแต่ละคอลัมน์สามารถทำโดยการหาจำนวนระเบียนของคู่ระหว่างค่าของข้อมูลในคอลัมน์กับค่าคลาสที่ผ่านค่าขีดแบ่ง 2 ค่าคือค่าค่าสนับสนุนขั้นต่ำ (Minimum Support) กับค่าความเชื่อมั่นขั้นต่ำ (Minimum Confidence) หารด้วยจำนวนระเบียนทั้งหมด ตัวอย่างเช่นการหาค่าอัตราความความถูกต้องในการจำแนก *CCR* ของแต่ละคอลัมน์ในตารางที่ 2.5 เริ่มจากคอลัมน์ Sex ถ้ากำหนดให้ค่าสนับสนุนขั้นต่ำมีค่าเท่ากับ 2 ระเบียนคิดเป็น 20% และให้ค่าความเชื่อมั่นขั้นต่ำเท่ากับ 65% คู่ระหว่างค่าของคอลัมน์กับค่าคลาสที่ผ่านค่าขีดแบ่งคือ Female $\rightarrow$ Flu ซึ่งมี 3

ระเบียน ฉะนั้นมีค่าสนับสนุนเท่ากับ 3 และระเบียนที่มีค่าเป็น Female มีทั้งหมด 4 ระเบียนฉะนั้นค่าความเชื่อมั่นเท่ากับ $\frac{3}{4}$ คิดเป็น 75% ดังนั้นอัตราความถูกต้องในการจำแนก *CCR* ของคอลัมน์ Sex เท่ากับ $\frac{3}{8}$ คิดเป็น 37% โดย 3 คือจำนวนระเบียนทั้งหมดของคู่ที่ผ่านค่าขีดแบ่ง 8 คือจำนวนระเบียนทั้งหมดของข้อมูล

ต่อมาหาค่าอัตราความถูกต้องในการจำแนก *CCR* ในคอลัมน์ Career มี 2 คู่ที่ผ่านค่าขีดแบ่งคือ A3 $\rightarrow$ Flu2009 มี 2 ระเบียน ฉะนั้นค่าสนับสนุนเท่ากับ 2 ค่า A3 มีทั้งหมด 2 ระเบียนฉะนั้นค่าความเชื่อมั่นเท่ากับ $\frac{2}{2}$ คิดเป็น 100% คู่ต่อมา B1 $\rightarrow$ Fever มี 2 ระเบียนฉะนั้นค่าสนับสนุนเท่ากับ 2 B1 มีทั้งหมด 3 ระเบียนฉะนั้นค่าความเชื่อมั่นเท่ากับ $\frac{2}{3}$ คิดเป็น 66% ดังนั้นอัตราความถูกต้องในการจำแนก *CCR* ของคอลัมน์ Career เท่ากับ $\frac{4}{8}$ คิดเป็น 50%โดย 4 คือจำนวนระเบียนทั้งหมดของคู่ที่ผ่านค่าขีดแบ่ง 8 คือจำนวนระเบียนทั้งหมดของข้อมูล

ตารางที่ 2.5 ตัวอย่างข้อมูลเพื่อใช้ในการทำ MCCRT

Tuple-ID	Sex	Career	Diag
1	Female	B2	Flu
2	Female	B1	Flu
3	Male	B1	Fever
4	Male	B1	Fever
5	Female	A2	Flu
6	Female	A2	Fever
7	Male	A3	Flu2009
8	Male	A3	Flu2009

จากขั้นตอนวิธี MCCRT เมื่อได้ค่าอัตราความถูกต้องในการจำแนก *CCR* ของแต่ละคอลัมน์แล้ว จะนำคอลัมน์มาเรียงตามค่าอัตราความถูกต้องในการจำแนก *CCR* จากน้อยไปหามากลงไปในระดับการเจนเนอรัลไลเซชัน *GL* จะได้ระดับการเจนเนอรัลไลเซชัย *GL* เท่ากับ <(Sex,

0), (Career, 0)> แล้วทำการแปลงค่าข้อมูลจาก คอลัมน์ Sex เพิ่มขึ้นไปทีละระดับตามการแปลงค่าข้อมูลตามลำดับขั้นขึ้นไปจนถึงระดับบนสุดแล้วจึงค่อยแปลงค่าในคอลัมน์ถัดมาในระดับการเจนเนอรัลไลเซชัน *GL* แปลงค่าข้อมูลและเพิ่มระดับไปจนข้อมูลมีคุณสมบัติ *k*-Anonymity ซึ่งในที่นี้ถ้ากำหนดให้ *k* เท่ากับ 2 จะได้ผลลัพธ์การแปลงข้อมูลจากขั้นตอนวิธี MCCRT ตามตาราง 2.6 ซึ่งมีค่าระดับการเจนเนอรัลไลเซชัน *GL* เท่ากับ <(Sex, 1), (Career, 1)>

ตาราง 2.6 ข้อมูลที่มีคุณสมบัติ *2*-Anonymity โดย *GL* เท่ากับ <(Career, 1), (Sex, 0)>

Tuple-ID	Sex	Career	Diag
1	*	B*	Flu
2	*	B*	Flu
3	*	B*	Fever
4	*	B*	Fever
5	*	A*	Flu
6	*	A*	Fever
7	*	A*	Flu2009
8	*	A*	Flu2009

2.4 เทคนิค IncSpan

เมื่อนำฐานข้อมูลใดๆมาทำการผ่านขั้นตอนวิธีใดขั้นตอนวิธีหนึ่งแล้วได้ผลลัพธ์ออกมาต่อมาถ้าฐานข้อมูลดังกล่าวมีข้อมูลเพิ่มเข้ามาจำนวนหนึ่งถ้าต้องการผลลัพธ์ใหม่อีกครั้งต้องนำฐานข้อมูลเดิมและข้อมูลที่เพิ่มเข้ามาไปผ่านขั้นตอนวิธีดังกล่าวใหม่อีกครั้งขั้นตอนวิธีนี้เรียกว่าเป็นขั้นตอนวิธีแบบ One-Time Fashion ใน [4] ได้เสนอแนวทางการสร้างขั้นตอนวิธีที่เรียกว่า Incremental หรือขั้นตอนวิธีแปลงค่าข้อมูลแบบเพิ่มขึ้นโดยมีแนวคิดว่า ในขณะที่ผ่านขั้นตอนวิธีแบบ One-Time Fashion ในครั้งแรกนั้นจะเก็บค่าบางค่าไว้เพื่อช่วยในการทำงาน เมื่อข้อมูลเพิ่มเข้ามาจะใช้ข้อมูลที่เก็บไว้ดังกล่าวกับข้อมูลที่เพิ่มเข้ามาในขั้นตอนวิธีแปลงค่าข้อมูลแบบเพิ่มขึ้นเพื่อให้ลดการทำงานกับข้อมูลเดิมและขั้นตอนการทำงานให้น้อยที่สุดซึ่งผลลัพธ์ที่ได้ต้องเหมือนกับการทำขั้นตอนวิธี One-Time Fashion ใหม่อีกครั้ง

[7] ได้เสนอวิธีการหารูปแบบลำดับ (Sequential Patterns) เมื่อมีการเพิ่มข้อมูลเข้ามาในภายหลัง รูปที่ 2.6 ได้แสดงวิธีการหารูปแบบลำดับจากตารางฐานข้อมูลซึ่งเป็นข้อมูลการขายสินค้าให้ลูกค้าประจำโดยเริ่มจากข้อมูลที่ได้จากฐานข้อมูลในรูปแบบของตารางนำมาทำการปรับเปลี่ยนให้อยู่ในลักษณะของข้อมูลอนุกรมเวลาแล้วจึงนำไปหาค่ารูปแบบลำดับต่อไป

ในการตีความหมายของรูปแบบลำดับนั้น จากตัวอย่างรูปที่ 2.6 จะสามารถตีความหมายของรูปแบบลำดับ < (30) (40 70) > ได้ว่า มีลูกค้ามากกว่า 25% ที่ซื้อสินค้ารหัส 30 แล้วจะกลับมาซื้อสินค้ารหัส 40 พร้อมกับสินค้ารหัส 70 ซึ่ง 25% นั้นเป็นค่าสนับสนุนขั้นต่ำ (Minimum Support) ที่กำหนดขึ้นมาซึ่งอาจอยู่ในลักษณะเป็นจำนวนระเบียนก็ได้

ส่วนการค้นหานั้นจะเห็นว่ารูปแบบลำดับ < (30) (40 70) > ปรากฏในระเบียน ID ที่ 2 และ ID ที่ 4 ของตารางข้อมูลอนุกรมเวลาซึ่งมีทั้งหมด 5 ID แสดงว่ารูปแบบลำดับ < (30) (40 70) > มีค่าสนับสนุน (Support) เท่ากับ $\frac{2}{5}$ หรือ 40% ถึงแม้ว่าในระเบียน ID ที่ 2 จะมีสินค้ารหัส 60 ขั้นอยู่แต่ก็ถือว่านับระเบียนนี้ได้เพราะการหารูปแบบลำดับไม่สนใจการคั่นกลางของข้อมูลที่เวลาเดียวกัน และจะเห็นว่าในความเป็นจริงรูปแบบลำดับ < (30) >, < (40) >, < (70) >, < (30) (40) >, < (30) (70) > ก็มีค่าสนับสนุนมากกว่า 25% แต่ในการหารูปแบบลำดับนั้นจะนับเฉพาะรูปแบบลำดับที่ใหญ่ที่สุด ตัวอย่างเช่น < (30) (70) > เป็น ส่วนหนึ่งของ < (30) (40 70) > ฉะนั้นจึงนับแต่รูปแบบลำดับ < (30) (40 70) >

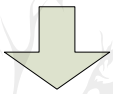
เมื่อมีการเพิ่มเข้ามาของข้อมูล [7] ได้แบ่งการเพิ่มเข้ามาของข้อมูลอนุกรมเวลาเป็น 2 แบบ คือ 1) แอพเพน (Append) หรือการเพิ่มข้อมูลการซื้อสินค้าของ Customer-ID เดิมที่มีอยู่แล้ว 2) อินเสิร์ท (Insert) หรือการเพิ่มเข้ามาของ Customer-ID กล่าวคือการมีลูกค้าประจำเพิ่มขึ้นมานั่นเองหรือสามารถมองได้อีกลักษณะว่าเป็นการแอพเพนกับเซทว่างก็ได้ ดังตัวอย่างตารางที่ 2.7

ในการหารูปแบบลำดับของข้อมูลอนุกรมเวลาเมื่อมีข้อมูลเพิ่มเข้ามา [7] ได้สร้างโครงสร้างข้อมูลแบบต้นไม้ (Tree) ขึ้นมาช่วยในการเก็บข้อมูลรูปแบบลำดับและเพื่อเพิ่มความเร็วในการค้นหารูปแบบลำดับ โครงสร้างข้อมูลแบบต้นไม้นี้ สร้างจากการหารูปแบบลำดับของข้อมูลอนุกรมเวลาในครั้งแรก โดยแบ่งรูปแบบลำดับออกเป็น 3 กลุ่มคือ 1) รูปแบบลำดับที่เกิดขึ้นบ่อยครั้งหรือเรียกว่าฟรีเควนท์ซีเควนซ์ (Frequent Sequence) ซึ่งมีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำ 2) รูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิหรือเรียกว่าเซมิฟรีเควนท์ซีเควนซ์ (Semi Frequent Sequence) ซึ่งมีค่าสนับสนุนเกือบถึงค่าสนับสนุนขั้นต่ำ โดยกำหนดว่าค่า

สนับสนุนต้องน้อยกว่าค่าสนับสนุนขั้นต่ำ แต่มากกว่าค่าสนับสนุนขั้นต่ำคูณกับค่ามิว (μ) ซึ่งค่ามิวนี้จะมีค่าอยู่ระหว่าง 0 ถึง 1 และเป็นค่าที่กำหนดขึ้นมาเองตามความเหมาะสมของงาน 3) รูปแบบลำดับที่เกิดขึ้นไม่บ่อยครั้งหรือเรียกว่าอินฟรีเควนท์ซีเควนซ์ (Infrequent Sequence) ซึ่งมีค่าสนับสนุนน้อยกว่าค่าสนับสนุนขั้นต่ำคูณกับค่ามิว

Customer ID	Transaction Time	Item Brought
1	June 25 '93	30
1	June 30 '93	90
2	June 10 '93	10, 20
2	June 15 '93	30
2	June 20 '93	40, 60, 70
3	June 25 '93	30, 50, 70
4	June 25 '93	30
4	June 30 '93	40, 70
4	July 25 '93	90
5	June 12 '93	90

ตารางข้อมูลการซื้อสินค้าของลูกค้า



Customer ID	Customer Sequence
1	< (30) (90) >
2	< (10 20) (30) (40 60 70) >
3	< (30 50 70) >
4	< (30) (40 70) (90) >
5	< (90) >

ตารางข้อมูลอนุกรมเวลา

Sequential Patterns with support > 25%
< (30) (90) >
< (30) (40 70) >

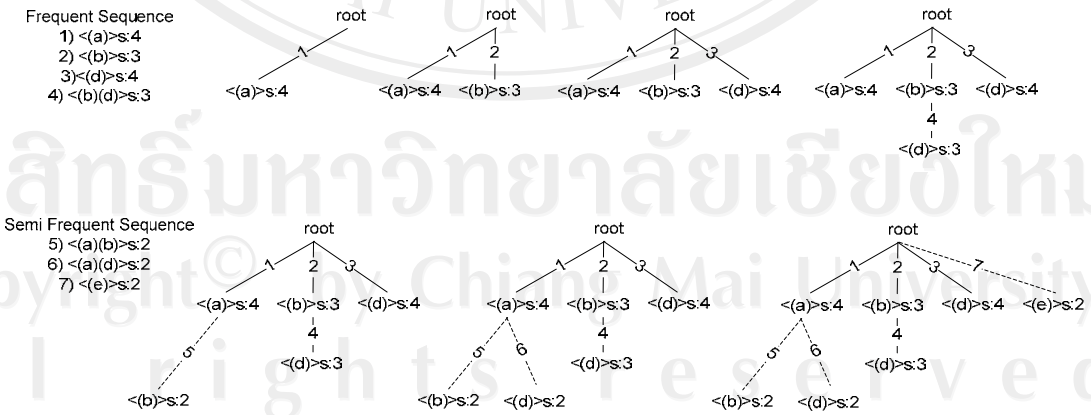
กฎความสัมพันธ์จากข้อมูลอนุกรมเวลา

รูปที่ 2.6 ขั้นตอนการหารูปแบบลำดับจากข้อมูลอนุกรมเวลา

ตารางที่ 2.7 ตัวอย่างการเติมข้อมูล

	Customer-ID	Original Part	Appended Part
	0	(a)(h)	(c)
	1	(eg)	(a)(bce)
	2	(a)(b)(d)	(ck)(l)
	3	(b)(df)(a)(b)	∅
	4	(a)(d)	∅
	5	(be)(d)	∅
New Insert	6	∅	(cd)(e)
	7	∅	(b)(d)

ในการสร้างโครงสร้างข้อมูลรูปต้นไม้นั้นจะนำรูปแบบลำดับในกลุ่ม 1) และ 2) มาใช้ในการสร้าง พิจารณาจากตารางที่ 11 ในส่วนของข้อมูลดั้งเดิมซึ่งเป็นส่วนที่ยังไม่มีการเพิ่มเข้ามาของข้อมูล สมมติกำหนดให้ค่าสนับสนุนขั้นต่ำเท่ากับ 3 ระเบียน ค่ามิวเท่ากับ 0.5 ซึ่งแสดงว่ารูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิต้องมี 2 ระเบียน เริ่มต้นโดยทำการหารูปแบบลำดับที่เกิดขึ้นบ่อยครั้งซึ่งในที่นี้ผลลัพธ์คือ < (a) > มีค่าสนับสนุนเท่ากับ 4, < (b) > มีค่าสนับสนุนเท่ากับ 3, < (d) > มีค่าสนับสนุนเท่ากับ 4, < (b)(d) > มีค่าสนับสนุนเท่ากับ 3 และรูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิคือ < (a)(b) >, < (a)(d) >, < (e) > ซึ่งทุกรูปแบบลำดับมีค่าสนับสนุนเท่ากับ 2 ต่อมาขั้นตอนวิธี InSpan จะนำรูปแบบลำดับที่เกิดขึ้นบ่อยครั้งและรูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิที่มาสร้างโครงสร้างข้อมูลต้นไม้ดังขั้นตอนในรูปที่ 2.7



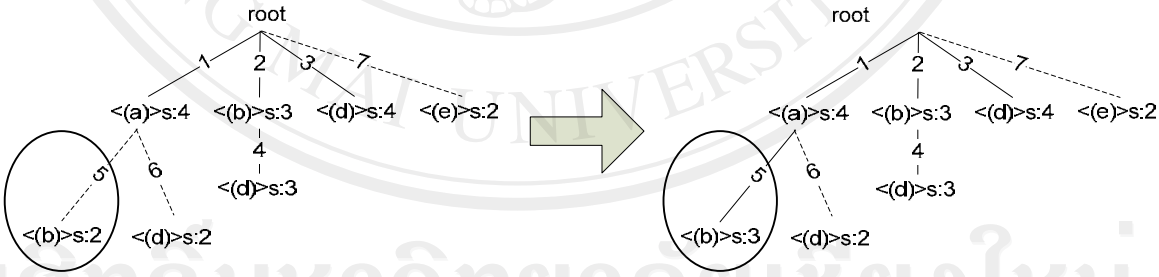
รูปที่ 2.7 ขั้นตอนการสร้างโครงสร้างรูปต้นไม้

เมื่อทำการสร้างโครงสร้างข้อมูลต้นไม้แล้ว จะเห็นว่าเมื่อมีการเพิ่มข้อมูล ตัวอย่างเช่นตารางที่ 2.7 จะต้องทำการปรับโครงสร้างข้อมูลรูปต้นไม้ใหม่อีกครั้ง ใน [7] ได้นำเสนอแนวคิดใน

การทำโครงสร้างต้นไม้ให้เป็นปัจจุบัน (Update) โดยมองว่าข้อมูลทุกข้อมูลใหม่ที่เพิ่มเข้ามาเป็นการแอพเพนและแยกกรณีการปรับโครงสร้างข้อมูลรูปต้นไม้เมื่อมีการเพิ่มเข้ามาของข้อมูลออกเป็น 6 กรณีคือ 1) ทำให้รูปแบบลำดับที่เกิดขึ้นบ่อยครั้งยังคงเป็นรูปแบบลำดับที่เกิดขึ้นบ่อยครั้ง 2) รูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิกลายเป็นรูปแบบลำดับที่เกิดขึ้นบ่อยครั้ง 3) ทำให้รูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิยังคงเป็นรูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิ 4) การแอพเพนทำให้เกิดสินค้าใหม่ (New item) 5) ทำให้รูปแบบลำดับที่เกิดขึ้นไม่บ่อยครั้งกลายเป็นรูปแบบลำดับที่เกิดขึ้นบ่อยครั้ง 6) ทำให้รูปแบบลำดับที่เกิดขึ้นไม่บ่อยครั้งกลายเป็นรูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิ

การปรับโครงสร้างข้อมูลรูปต้นไม้ให้เป็นปัจจุบันในแต่ละกรณี

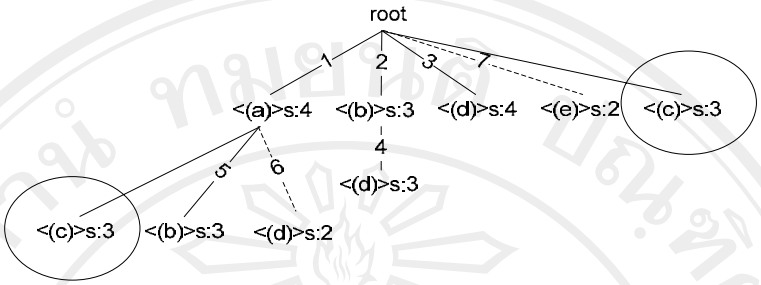
กรณีที่ 1) 2) และ 3) สามารถใช้วิธีเดียวกันได้ โดยการอ่านเฉพาะข้อมูลที่เพิ่มเข้ามาหาค่าสนับสนุนที่เพิ่มขึ้นแล้วทำการปรับโครงสร้างข้อมูลรูปต้นไม้ได้ ยกตัวอย่างเช่น จากตารางที่ 2.7 ระเบียนที่ 7 การแอพเพนทำให้เกิดกรณีที่ 2) รูปแบบลำดับ < (a) (b) > มีค่าสนับสนุนจาก 2 เป็น 3 ทำให้จากเดิมที่เป็นรูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิกลายเป็นรูปแบบลำดับที่เกิดขึ้นบ่อยครั้ง ในการปรับโครงสร้างข้อมูลรูปต้นไม้ให้ทำการเปลี่ยนจากเส้นประเป็นเส้นทึบแล้วปรับค่าสนับสนุนเป็น 3 ดังรูปที่ 2.8 ส่วนในกรณีที่ 1) และ 3) เพียงปรับค่าสนับสนุนเท่านั้น



รูปที่ 2.8 การเปลี่ยนโครงสร้างข้อมูลรูปต้นไม้ต้นไม้ในกรณีที่ 2)

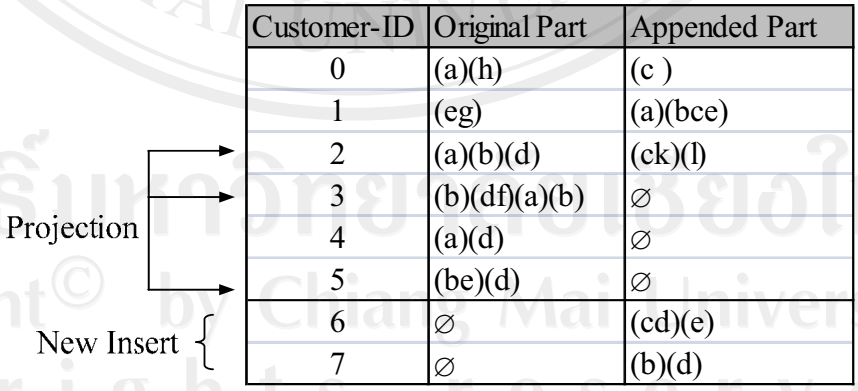
กรณีที่ 4) พิจารณาการแอพเพนในระเบียนที่ 0, 1 และ 2 จะเห็นว่าการแอพเพนได้เพิ่มสินค้า c ซึ่งไม่มีอยู่ในโครงสร้างข้อมูลรูปต้นไม้เมื่อพิจารณาหลังการแอพเพนจะได้รูปแบบลำดับใหม่อีกหลายลำดับที่เกิดจากเพิ่มเข้ามาของ c, k และ 1 แต่จะมีเพียงรูปแบบลำดับ < (c) > กับ < (a) (c) > ที่มีค่าสนับสนุนมากพอจะเพิ่มในโครงสร้างข้อมูลรูปต้นไม้ ซึ่งในกรณีนี้จะเป็นลำดับ

ที่เกิดขึ้นบ่อยครั้งในการปรับโครงสร้างจะทำการเพิ่ม < (c) > ต่อจาก root กับ < (c) > ต่อกับ < (c) > โดยใช้เส้นทึบดังผลลัพธ์จะได้ดังรูปที่ 2.9



รูปที่ 2.9 การเปลี่ยนโครงสร้างข้อมูลรูปต้นไม้ในกรณีที่ 4)

กรณีที่ 5) และ 6) พิจารณาการแอพเพนในระเบียนที่ 1 จะเห็นว่า < (b e) > ไม่มีอยู่ในโครงสร้างข้อมูลรูปต้นไม้แต่ปรากฏในตารางระเบียนที่ 5 ในการปรับโครงสร้างข้อมูลจำเป็นต้องอ่านข้อมูลจากตารางอีกครั้งเพื่อค้นหาว่ารูปแบบลำดับ < (b e) > จะมีค่าสนับสนุนเป็นเท่าไร เพียงพอที่จะเป็นรูปแบบลำดับที่เกิดขึ้นบ่อยครั้งหรือรูปแบบลำดับที่เกิดขึ้นบ่อยครั้งแบบเซมิหรือไม่ เพื่อนำไปเพิ่มในโครงสร้างต้นไม้ ในการค้นหานั้นถ้าใช้การค้นหาจากบนลงล่างจะทำให้การประมวลผลไม่มีประสิทธิภาพ ใน [7] จึงใช้เทคนิคการโปรเจคชัน (Projection) เข้ามาช่วยซึ่งการโปรเจคชันจะเข้าถึงข้อมูลไปที่ละค่าดังตัวอย่างรูปที่ 2.10เมื่อมอง < (b e) > ก็เข้าถึงทุกระเบียนที่มี b ก่อนจากนั้นจึงเข้าถึง < (b e) > โดยการเข้าถึงจะเป็นการเข้าถึงโดยตรง



Customer-ID	Original Part	Appended Part
0	(a)(h)	(c)
1	(eg)	(a)(bce)
2	(a)(b)(d)	(ck)(l)
3	(b)(df)(a)(b)	∅
4	(a)(d)	∅
5	(be)(d)	∅
6	∅	(cd)(e)
7	∅	(b)(d)

รูปที่ 2.10 การโปรเจคชันโดย b

ฉะนั้นโครงสร้างข้อมูลสุดท้ายหลังการทำงานของทั้ง 6 กรณีแล้วจะเป็นโครงสร้างข้อมูลต้นไม้ที่ให้รูปแบบลำดับครบตามที่ต้องการเหมือนการ อ่านข้อมูลใหม่ทั้งหมดเพื่อหารูปแบบลำดับอีกครั้ง