การใช้เทคนิค Association Rule Discovery เพื่อการจัดสรรกฎหมายในการพิจารณาคดีความ

กฤษณะ ไวยมัย และ ธีระวัฒน์ พงษ์ศิริปรีดา
อาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
และนิสิตปริญญาโทวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

ABSTRACT - Data Classification and Association Rule Discovery are two important data mining techniques. In this paper, we apply the two techniques to find appropriate laws and articles for lawsuits. We propose a data classification method using a classifier generated from association rule discovery technique. We utilize the classifier to help selecting laws and articles to be tried for a lawsuit. Our experiment shows that the classifier generated from a proposed method yields better accuracy than one generated from a general data classification method. In addition, our method increases efficiency in multi-group and multi-level classification.

KEY WORDS – data mining, knowledge discovery from large database, lawsuit

บทคัดย่อ – เทคนิค Data classification และเทคนิค Association rule discovery เป็นเทคนิคที่สำคัญของดาต้าไมน์นิง ในงานวิจัยฉบับนี้ เราได้นำเทคนิคทั้งสองมาประยุกต์ใช้ในการจัดสรรกฎหมายที่เหมาะสมกับคดีความ โดยนำเทคนิค Data classification มาสร้างตัวจำแนกข้อมูลจากกฎเกณฑ์ที่ได้จากเทคนิค Association rule discovery อีกทีหนึ่ง ซึ่งตัวจำแนกข้อมูลที่ได้นี้จะสามารถนำไปใช้ทำนายคดีความแต่ละคดีว่าควรใช้กฎหมายฉบับใดในการพิจารณา ผลการวิจัยที่ได้ แสดงให้เห็นว่าการสร้างตัวจำแนกตามวิธีที่เสนอนี้ได้ประสิทธิภาพดีกว่าการสร้างตัวจำแนกตามเทคนิค Data classification แบบปกติ นอกจากนี้ วิธีดังกล่าวยังช่วยแก้ปัญหาต่างๆ ที่เกิดขึ้นจากการใช้เทคนิค Data classification โดยทั่วไป

คำสำคัญ – ดาต้าไมน์นิง, การสืบค้นความรู้จากฐานข้อมูลขนาดใหญ่, การจัดสรรกฎหมาย

1. บทนำ

ในการพิจารณา หรือตัดสินคดีความใดๆ นักกฎหมายจำเป็นต้องอ้างอิงตัวบทกฎหมายเพื่อประกอบการพิจารณากฎหมายไทยที่ใช้อยู่ในปัจจุบันสามารถจำแนกออกได้เป็น พระราชบัญญัติ และพระราชกำหนด [1] ซึ่งมีจำนวนรวมกันประมาณ 260 ฉบับ ในแต่ละฉบับจะประกอบด้วยมาตราย่อยต่างๆ มากน้อยแตกต่างกันไป บางฉบับอาจมีจำนวนมาตราสูงถึงหลายพันมาตรา จำนวนกฎหมายที่มีปริมาณมากเช่นนี้ ทำให้ยากต่อการที่จะศึกษาได้อย่างครบถ้วน เมื่อนักกฎหมายจำเป็นต้องพิจารณาคดีความใดๆ จึงเกิดปัญหาในการเลือกกฎหมายฉบับที่เหมาะสม มาใช้ประกอบการพิจารณาคดีความ ในปัจจุบันนักกฎหมายโดยทั่วไปมักจะใช้ทักษะความชำนาญในสาขาวิชาชีพที่มีอยู่ ประกอบกับการสืบค้นคดีความเก่าๆ ขึ้นมาพิจารณา เพื่อดูว่าคดีที่เคยตัดสินไปแล้วนั้น อ้างถึงกฎหมายฉบับใดบ้าง เพื่อใช้เป็นแนวทางในการพิจารณาคดีที่กำลังสนใจอยู่ต่อไป

ที่ผ่านมาได้มีการใช้การค้นหาคำ หรือวลีแบบ Full-Text-Search [2] เข้ามาช่วย โดยการค้นหาคำหรือวลีจากคดีความเก่าๆ ที่เก็บไว้ เพื่อนำมาพิจารณาว่าสอดคล้องกับคดีปัจจุบันที่กำลังพิจารณาอยู่หรือไม่ ซึ่งผลลัพธ์ที่ได้ยังไม่เป็นที่น่าพอใจ เนื่องจากผู้ใช้ไม่สามารถตัดสินใจได้ว่าควรเลือกใช้ข้อความส่วนใดบ้างในการค้นหา เพราะถ้าระบุคำค้นมากอาจทำให้ค้นหาคดีความเก่าไม่พบ หรือถ้าระบุคำค้นน้อยเกินไป ก็จะทำให้ได้ผลลัพธ์มากเกินจนไม่มีความหมาย นอกจากนี้ผู้ใช้ยังต้องเสียเวลาในการอ่านคดีความเก่าๆที่ค้นมาได้ เพื่อพิจารณาว่าใจความของคดีเก่านั้น มีความหมายตรงตามที่ต้องการหรือไม่

งานวิจัยฉบับนี้ได้มุ่งเน้นที่จะศึกษาและคิดค้นเพื่อสร้างระบบที่สามารถทำนายได้ว่าแต่ละคดีมีความน่าจะเป็นที่จะใช้กฎหมายฉบับไหน และมาตราใดประกอบการพิจารณา เพื่อเป็นแนวทางในการศึกษา หรือทำงานกับคดีความนั้นๆ ต่อไป ระบบนี้นอกจากจะสามารถช่วยนักกฎหมายในการทำงานให้สะดวกรวดเร็วขึ้นแล้ว ยังสามารถเอื้อประโยชน์อย่างมากต่อประชาชนทั่วไป ที่ไม่มีความรู้ ความชำนาญทางด้านกฎหมาย ในการที่จะหากฎหมายที่เหมาะสมมาศึกษา เพื่อใช้ประกอบคดีความที่กำลังสนใจอยู่ โดยในการสร้างระบบดังกล่าว เรา

เลือกใช้เทคนิคการสืบค้นความรู้ที่เป็นประโยชน์และน่าสนใจจากฐานข้อมูลขนาดใหญ่ (Knowledge Discovery from very large Database: KDD) หรือที่เรียกกันว่า Data Mining [3,4,5,6,7]

Data Classification [8,9] และ Association Rule Discovery [10,11,12] เป็นเทคนิคที่สำคัญของ KDD เทคนิค Data Classification เป็นเทคนิคที่ใช้หากฎเกณฑ์ความสัมพันธ์ของข้อมูล เพื่อนำมาสร้างตัวจำแนกประเภทของข้อมูล ส่วนเทคนิค Association Rule Discovery เป็นเทคนิคที่ใช้หากฎเกณฑ์ความสัมพันธ์ทั้งหมดของข้อมูล ที่มีคุณสมบัติสอดคล้องตามค่า Minimum Support และ Minimum Confidence ข้อแตกต่างที่ชัดเจนของเทคนิคทั้งสอง คือ Data Classification เป็นเทคนิคที่กำหนดผลลัพธ์แน่นอน ซึ่งก็คือจำนวนประเภทของข้อมูลที่ต้องการจำแนก ส่วน Association Rule Discovery เป็นเทคนิคที่ไม่มีการกำหนดขอบเขตของผลลัพธ์ล่วงหน้า

ในงานวิจัย *Using Data Mining technique to select the laws and articles for lawsuits* [13] ได้มีการนำเทคนิค Data Classification มาใช้สร้างระบบจัดสรรกฎหมายที่เหมาะสมให้กับคดีความแบบอัตโนมัติ ผลการทดลองที่ได้ยังไม่เป็นที่น่าพอใจ ซึ่งสามารถสรุปได้ 3 ประเด็นคือ

ประเด็นที่ 1 การตัดคำในขั้นตอนการเตรียมข้อมูลยังขาดประสิทธิภาพ ทำให้ผลการทำนายของระบบมีความถูกต้องต่ำ

ประเด็นที่ 2 ระบบไม่สามารถทำนายกฎหมายให้กับคดีความได้ครั้งละมากกว่าหนึ่งกฎหมาย

ประเด็นที่ 3 ระบบไม่สามารถทำนายกฎหมายแบบมีลำดับชั้นในเวลาเดียวกันได้

จากงานวิจัย *Integrating Classification and Association Rule Mining* [14] ได้แสดงให้เห็นว่าเทคนิค Association Rule Discovery สามารถนำมาประยุกต์ร่วมกับเทคนิค Data Classification เพื่อใช้ในการสร้างตัวจำแนกข้อมูล โดยการใช้เทคนิค Association Rule Discovery หากฎเกณฑ์ความสัมพันธ์ของข้อมูลที่มีต่อประเภทของข้อมูลนั้นๆ กฎเกณฑ์ที่ได้จากขั้นตอนนี้เรียกว่า Class Association Rules (CARs) ซึ่งจะถูกนำไปใช้ในการสร้างตัวจำแนกข้อมูลต่อไป

เราได้ประยุกต์เทคนิคดังกล่าว มาใช้ในการสร้างตัวทำนายกฎหมายของระบบ ซึ่งผลการทดลองที่ได้ แสดงให้เห็นว่าการสร้างตัวทำนายด้วยวิธีดังกล่าว ให้เปอร์เซ็นต์ความถูกต้องสูงกว่าการสร้างตัวทำนายด้วยวิธี Data Classification แบบทั่วไป นอกจากนี้ วิธีดังกล่าวยังช่วยแก้ปัญหาทั้ง 3 ประการของระบบเดิมดังที่ได้กล่าวมาแล้ว

ในงานวิจัยฉบับนี้ เราได้อธิบายถึงเทคนิค และรายละเอียดการทดลองซึ่งแบ่งออกเป็น 3 ขั้นตอน คือ

1. ขั้นตอนการเตรียมข้อมูล
2. ขั้นตอนการหา Class Association Rules
3. ขั้นตอนการสร้างตัวจำแนกข้อมูลจาก Class Association Rules

ในตอนท้าย เราได้สรุปผลการทดลองโดยการแสดงกราฟเปรียบเทียบประสิทธิภาพของระบบใหม่ กับระบบเก่า พร้อมทั้งได้เสนอแนะแนวทางในการพัฒนาระบบต่อไป

2. ขั้นตอนการเตรียมข้อมูล (Pre-processing)

2.1 ข้อมูลคดีความ

ข้อมูลที่ใช้ทดสอบระบบ คือข้อมูลคดีความของศาลฎีกา ซึ่งเป็นศาลสูงสุด คดีความที่เข้าสู่ขั้นตอนการพิจารณาของศาลฎีกาเหล่านี้ เป็นคดีความที่ผ่านขั้นตอนการพิจารณาของศาลชั้นต้น และศาลอุทธรณ์มาแล้ว ซึ่งประกอบด้วยคดีอาญา และคดีแพ่ง ข้อมูลคดีความของศาลฎีกาที่นำมาทดสอบนี้ ประกอบด้วยรายละเอียดของคดีความ ได้แก่ ปีที่มีการวินิจฉัย ชื่อโจทย์ และจำเลย ชื่อกฎหมายที่ใช้ประกอบการพิจารณาตัดสินคดี และตัวเนื้อคดีความ ในตัวคดีความจะเป็นส่วนการบรรยายรูปคดี ที่ผ่านขั้นตอนการพิจารณาจากศาลชั้นต้น และศาลอุทธรณ์มาแล้ว ข้อมูลทั้งหมดสามารถแบ่งได้เป็น 30,000 คดี ตั้งแต่ปี พ.ศ. 2489 ถึงปี พ.ศ. 2541 ซึ่งโดยเฉลี่ยแล้ว แต่ละคดีความจะใช้กฎหมาย 1-2 ฉบับ ฉบับละ 3-4 มาตรา ในการพิจารณาจากข้อมูลทั้งหมด เราสามารถแบ่งกลุ่มกฎหมายที่ใช้พิจารณาคดีความออกได้เป็น 20 กลุ่ม และกลุ่มมาตราได้ 2200 กลุ่ม

2.2 การตัดคำ

คดีความแต่ละคดีจะถูกตัดคำด้วยพจนานุกรมภาษาไทย เพื่อแบ่งคดีความออกเป็นคำสั้นๆ สำหรับใช้ประมวลผลในขั้นตอนต่อไป พจนานุกรมภาษาไทยที่ใช้ในระบบนี้ เราสร้างขึ้นมาจากการหาวลีจากคดีความทั้งหมด โดยการใช้เทคนิค Suffix array [15]

เทคนิค Suffix array เป็นเทคนิคที่ใช้กันอย่างแพร่หลายเพื่อพิจารณาความสัมพันธ์ของชุดข้อมูล เราใช้เทคนิคนี้ในการหา กลุ่มของตัวอักษรที่ยาวที่สุดที่ปรากฏซ้ำๆอยู่ในชุดข้อมูล ซึ่งก็คือวลีที่เราต้องการนั้นเอง

ให้ A แทน String ใดๆ (ซึ่งถ้ามองอีกรูปหนึ่ง A ก็คือ array of characters นั้นเอง) เราจะได้ S แทน Suffix array ของ A โดยที่ S คือ Array of pointer ที่ชี้ไปที่ตำแหน่งในหน่วยความจำของสมาชิกใน A แต่ละตัว ฉะนั้นสมาชิกของ S แต่ละตัวก็คือ sub string ที่เริ่มต้นด้วยตัวอักษรตำแหน่งต่างๆ ของ A

ยกตัวอย่างเช่น A = banana
เราจะได้ S[0] = banana
S[1] = anana
S[2] = nana
S[3] = ana
S[4] = na
S[5] = a

จาก S ที่ได้ เรานำสมาชิกทั้งหมดของ S มาจัดเรียงลำดับจากน้อยไปมาก ซึ่งจะได้ผลดังต่อไปนี้

S[5] = a
S[3] = ana
S[1] = anana
S[0] = banana
S[4] = na
S[2] = nana

จาก S ที่ได้ผ่านการเรียงลำดับแล้ว จะทำให้เราสามารถหากลุ่มของตัวอักษรที่ยาวที่สุดที่ซ้ำกันในชุดของข้อมูลที่ติดกัน ซึ่งก็คือ ana ที่ปรากฏอยู่ที่ S[3] และ S[1] นั่นเอง

รูปที่ 1 แสดงอัลกอริทึมของโปรแกรมการหาวลีจากชุด string ที่ผ่านการเรียงข้อมูลแล้ว

```
1  FOR  m = 1  TO  Count(S) - FNUM  DO
2      sl = 999;
3      FOR  i = m  TO  m + FNUM  DO
4          FOR  c = 1  TO  Length(S_i)  DO
5              IF  S_i [c] < > S_i+1 [c]  THEN
6                  IF  c < sl  THEN  sl = c
7                  break;
8              END
9          END
10     END
11     IF  sl > LNUM  THEN  ReturnPhase ( S 1..sl )
12 END
```

รูปที่ 1. แสดงอัลกอริทึมของโปรแกรมหาวลี

S คือ จำนวนชุดตัวอักษรย่อยที่ผ่านการจัดเรียงลำดับแล้ว

LNUM คือ จำนวนตัวอักษรน้อยที่สุดของชุดตัวอักษร ที่จะถูกเลือกเป็นวลี

FNUM คือ จำนวนความถี่น้อยที่สุดในการเกิดชุดตัวอักษรซ้ำๆกัน ที่จะถูกเลือกเป็นวลี

sl คือ ตัวแปรสำหรับนับจำนวนตัวอักษรที่เกิดซ้ำกันในชุดข้อมูลย่อยทั้ง 2 ชุด

โปรแกรมจะวนรอบเป็นจำนวนครั้ง เท่ากับ จำนวนชุดตัวอักษรย่อย Count (S) ลบด้วยค่า FNUM ในแต่ละรอบ โปรแกรมจะเริ่มด้วยการให้ค่าเริ่มต้นตัวแปร sl มีค่าสูงสุดซึ่งคือ 999 (บรรทัดที่ 2) ถัดมาจะวนรอบเพื่อเปรียบเทียบชุดตัวอักษรเป็นจำนวนครั้ง เท่ากับค่า FNUM (บรรทัดที่ 3) ในรอบเปรียบเทียบชุดตัวอักษรนี้ โปรแกรมจะเปรียบเทียบตัวอักษรทีละตัวเป็นจำนวนครั้งเท่ากับจำนวนตัวอักษรในชุดตัวอักษรนั้น (บรรทัดที่ 4) หรือเมื่อพบตัวอักษรที่ต่างกันของ 2 ชุดตัวอักษร (บรรทัดที่ 5 ถึง 8) ทุกครั้งที่พบตัวอักษรที่ต่างกัน โปรแกรมจะตรวจสอบค่า sl กับ ตำแหน่งตัวอักษรที่เริ่มต่างกันนี้ (บรรทัดที่ 6) ถ้าหากว่าตำแหน่งตัวอักษรนี้น้อยกว่าค่า sl ก็จะ กำหนดค่า sl ให้เท่ากับตำแหน่งตัวอักษรนั้น ด้วยวิธีนี้เมื่อโปรแกรมทำงานผ่านไปครบจำนวน FNUM รอบ จะได้ค่า sl เป็นค่าของจำนวนตัวอักษรของชุดตัวอักษรที่เกิดซ้ำกันในชุดตัวอักษรทั้ง FNUM ชุด ซึ่งถ้าหากว่าค่า sl นี้ไม่น้อยกว่าค่า LNUM ที่กำหนดไว้โปรแกรมก็จะเลือกให้ตัวอักษรตัวที่ 1 ถึง sl เป็นวลี (บรรทัดที่ 11)

เรานำเทคนิคดังกล่าว มาใช้กับคดีความที่มีอยู่ โดยแบ่งกลุ่มคดีความเป็นกลุ่มๆ 70 กลุ่ม ตามกฎหมายที่ใช้พิจารณาระดับมาตรา แต่ละกลุ่มมี 100 คดีความ หลังจากนั้น เราจะโหลดคดีความครั้งละกลุ่มเข้าสู่หน่วยความจำเพื่อสร้างเป็น Array of characters (A) ในขั้นตอนนี้ เราจะได้ A คือ ชุดของตัวอักษรในคดีความเรียงต่อเนื่องกัน ซึ่งมีขนาดประมาณ 200 Kbytes

เราให้ S เป็น Array of pointer ที่ชี้ไปที่ตำแหน่งสมาชิกแต่ละตัวของ A เมื่อเราจัดเรียง S ด้วยเทคนิค quick sort ก็จะได้ผลลัพธ์คือ sub string ย่อยๆ จำนวน 200 K strings ที่เรียงลำดับจากน้อยไปมาก หลังจากนั้นเราสามารถหากลุ่มของตัวอักษรยาวที่สุดที่ซ้ำๆกัน ซึ่งก็คือวลีต่างๆ ที่เราต้องการนั้นเอง

จากวิธีการดังกล่าว ขั้นตอนที่ใช้เวลานานที่สุดคือขั้นตอนการเปรียบเทียบชุดตัวอักษรเพื่อหาวลี ซึ่งเราได้ปรับปรุงวิธีการให้เร็วขึ้น โดยการตัดคำชุดของตัวอักษร A ด้วยพจนานุกรมคำเดี่ยวเสียก่อน หลังจากนั้นให้ S แต่ละตัว ชี้ไปที่ตำแหน่งหน่วยความจำของตัวอักษรใน A ที่เป็นตัวเริ่มต้นคำใหม่เท่านั้น ด้วยวิธีการนี้จะทำให้ S มีขนาดลดลง 70% ซึ่งทำให้ขั้นตอนการหาวลีจากชุดตัวอักษรมีความเร็วขึ้นมาก

เมื่อเราทำขั้นตอนดังกล่าวกับข้อมูลครบทั้ง 70 ชุด เราจะได้ผลลัพธ์คือวลีจำนวน 23,722 วลี ซึ่งจะถูกนำไปสร้างเป็นพจนานุกรมเพื่อใช้ตัดคำต่อไป

2.3 การแปลงข้อมูล

พจนานุกรมที่ได้จากขั้นตอน 2.2 จะถูกจัดเก็บลงฐานข้อมูล โดยแต่ละวลีจะมีค่าตัวเลขกำกับอยู่เพื่อใช้ในการแปลงวลีให้อยู่ในรูปตัวเลข เพื่อความสะดวกในการทำงานขั้นตอนต่อไป สำหรับข้อมูลคดีความที่มีอยู่เราได้แปลงเข้าสู่ฐานข้อมูลเช่นกัน โดยเนื้อความของ แต่ละคดี จะถูกตัดคำด้วยพจนานุกรมดังกล่าว ซึ่งจะได้ผลลัพธ์เป็นตัวเลขแทนวลีที่ปรากฏอยู่ในคดีความนั้นๆ ในแต่ละคดีความมีรหัสกฎหมายที่ใช้ในการพิจารณาระบุอยู่ด้วย ซึ่งแบ่งเป็น 2 ระดับคือระดับกฎหมาย และระดับมาตรา รูปแบบของฐานข้อมูลแสดงได้ดังรูปที่ 2 ซึ่งประกอบไปด้วย รหัสกฎหมายและ มาตรา (LAW_CHAPTER) รหัสคดีความ (CASE) และวลีที่ปรากฏอยู่ในแต่ละคดีความ (WORDID)

LAW_CHAPTER	CASE	WORDID
01_249	2489_0002	17566
01_249	2489_0002	14690
01_249	2489_0002	14690
01_249	2489_0002	9830
01_249	2489_0002	20880
01_249	2489_0002	5189
01_249	2489_0002	9231
01_249	2489_0002	21113
01_249	2489_0002	682
01_249	2489_0002	5138
01_249	2489_0002	11008
01_249	2489_0002	5189
01_249	2489_0009	15651
01_249	2489_0009	21122
01_249	2489_0009	5189
01_249	2489_0009	1532
01_249	2489_0009	21549
01_249	2489_0009	1913
01_249	2489_0009	284
01_249	2489_0009	23144
01_249	2489_0009	7889
01_249	2489_0009	22295
01_249	2489_0009	2846

รูปที่ 2. แสดงโครงสร้างข้อมูลที่ผ่านการเตรียม

3. การหา Class Association Rules (CARs)

ด้วยเทคนิคของ Association Rule Discovery เราสามารถหากฎเกณฑ์ ที่บอกความสัมพันธ์ของข้อมูล โดยกฎเกณฑ์ที่ได้จะต้องมีคุณสมบัติสอดคล้องตาม ค่า MinSupt และ MinConfd ที่กำหนดโดยผู้ใช้ กฎเกณฑ์ที่ได้นี้เรียกว่า Association Rules ซึ่งจะมีรูปแบบดังต่อไปนี้

{item1, item2} → item3 Supt% Confd%

- *Supt* คือ เปอร์เซ็นต์ของจำนวนข้อมูลที่มีสมาชิกสอดคล้องตามกฎ ต่อจำนวนข้อมูลทั้งหมด
- *Confd* คือ เปอร์เซ็นต์ของจำนวนข้อมูลที่สอดคล้องตามกฎ ต่อจำนวนข้อมูลทั้งหมดที่มีสมาชิกตามกฎฝั่งซ้ายมือ
- *MinSupt* คือ ค่า Supt ต่ำสุดที่ทำให้เกิดกฎเกณฑ์
- *MinConfd* คือ ค่า Confd ต่ำสุดที่ทำให้เกิดกฎเกณฑ์

จากตัวอย่างกฎข้างบน มีความหมายว่า เมื่อมี item1 และ item2 ปรากฏอยู่ในข้อมูล จะมี item3 ปรากฏอยู่ในข้อมูลชุดนั้นด้วยเสมอ ตามเปอร์เซ็นต์ความเชื่อมั่น (Confd) และตามเปอร์เซ็นต์ปริมาณข้อมูลที่สอดคล้องตามกฎ (Supt)

Class Association Rules (CARs) คือกฎเกณฑ์ที่ได้จากกระบวนการ Association Rule Discovery เช่นกัน แต่เป็นกฎเกณฑ์ที่บอกความสัมพันธ์ของข้อมูลที่มีต่อประเภทข้อมูลนั้นๆ (Class) กฎเกณฑ์ที่ได้นี้ จะมีลักษณะดังต่อไปนี้

{item1, item2} → Class Supt% Confd%

เราได้นำเทคนิคดังกล่าวมาใช้กับงานกฎหมาย โดยอาศัยข้อมูลที่ได้ผ่านขั้นตอนการเตรียมข้อมูลมาแล้วส่วนหนึ่งคือ 70% เพื่อสอนให้ระบบเรียนรู้การจำแนกประเภทข้อมูล เป็นการเรียนรู้เพื่อหากฎเกณฑ์ความสัมพันธ์ของวลีในคดีความ ที่มีต่อกฎหมายที่ใช้พิจารณาคดีความนั้นๆ ในกรณีนี้ประเภทของข้อมูลก็คือกฎหมายฉบับที่ใช้พิจารณาคดีความนั้นเอง กฎเกณฑ์ที่ได้นี้จะอยู่ในรูปต่อไปนี้

วลี → กฎหมาย Supt% Confd%

จากรูปแบบ CARs ที่แสดงข้างบน มีความหมายว่า ถ้าปรากฏวลีหนึ่งในคดีความ มีความน่าจะเป็นที่คดีความนั้นจะใช้กฎหมาย ที่ระบุอยู่ทางฝั่งขวามือของกฎเป็นตัวพิจารณาคดี ตามค่าความเชื่อมั่น Confd% และมีความถี่ของการเกิดเหตุการณ์เช่นนี้ Supt%

ซ้ายมือของกฎก็คือวลีต่างๆ ที่หาได้จากคดีความ เรากำหนดให้ฝั่งซ้ายมือของกฎเกณฑ์มีค่าเป็นวลี 1 วลี ไม่ใช่เซตของวลี เนื่องจากว่าจำนวนวลีในระบบมีมากถึง 23,722 วลี ทำให้ไม่สามารถหากฎเกณฑ์ของเซตวลีได้ และเนื่องจากว่าวลีที่ใช้ในระบบนี้คือวลีที่มีความยาวพอสมควร (ผ่านการตัดคำแบบวลีในขั้นตอนการเตรียมข้อมูล) ซึ่งในตัววลีแต่ละตัวจะประกอบไปด้วยคำสั้นๆ รวมกันอยู่แล้ว ทำให้ไม่จำเป็นต้องหากฎเกณฑ์ของเซตวลีอีกต่อไป

ฝั่งขวามือของกฎคือ กฎหมาย ซึ่งอาจหมายถึงกฎหมายฉบับที่ใช้ หรือมาตราที่ใช้ประกอบการพิจารณาตัดสินคดีความนั้นๆ

สำหรับค่า MinSupt และค่า MinConfd เรากำหนดให้มีค่า 0 เปอร์เซ็นต์ ในระหว่างทำการทดลอง เนื่องจากว่าข้อมูลในระบบนี้ที่ทำการทดลอง มีการกระจายตัวสูง ทำให้ค่า Supt และค่า Confd ของ CARs ที่ได้มีค่าต่ำมาก อย่างไรก็ตาม CARs ต่างๆที่หาได้จากขั้นตอนนี้ จะถูกคัดเลือกเพื่อนำไปใช้สร้างตัวจำแนกข้อมูลอีกครั้ง ในขั้นตอนต่อไป ฉะนั้นค่า Supt และค่า Confd จึงไม่มีความหมายต่อการหา CARs ในขั้นตอนนี้

ยกตัวอย่างข้อมูลที่นำมาสร้าง CARs ดังแสดงในตารางที่ 1

ตารางที่ 1. แสดงตัวอย่างข้อมูลที่นำมาสร้าง CARs

กฎหมาย และมาตรา	วลี
กฎหมายลักษณะอาญา มาตรา 1	ทำร้ายร่างกาย
กฎหมายลักษณะอาญา มาตรา 2	ทำร้ายร่างกาย
กฎหมายวิธีพิจารณาความอาญา มาตรา 3	ทำร้ายร่างกาย
กฎหมายแพ่งและพาณิชย์ มาตรา 4	ไม่ชำระเงินตามสัญญา
กฎหมายแพ่งและพาณิชย์ มาตรา 4	ไม่ชำระเงินตามสัญญา

เนื่องจากประเภทของข้อมูล ที่ต้องการจำแนก มีอยู่ 2 ประเภท คือ รหัสกฎหมาย และมาตราของกฎหมาย ทำให้การหากฎเกณฑ์ต้องแบ่งออกเป็น 2 ชุด ซึ่งจะมีลักษณะดังต่อไปนี้

ตารางที่ 2. แสดง CARs ระดับกฎหมาย

Rules	Supt	Confd
ทำร้ายร่างกาย → กฎหมายลักษณะอาญา	40.00%	66.67%
ทำร้ายร่างกาย → กฎหมายวิธีพิจารณาความอาญา	20.00%	33.33%
ไม่ชำระเงินตามสัญญา → กฎหมายแพ่งและพาณิชย์	40.00%	100.00%

จากอัลกอริทึมรูปที่ 3 โปรแกรมจะเริ่มต้นด้วยการกำหนดค่าเริ่มต้นให้กับตัวแปร f มีค่าเป็น 0 (บรรทัดที่ 1) หลังจากนั้นจะเริ่มวนรอบทำงานกับข้อมูลของวลีทีละตัว เป็นจำนวน Count(Word) รอบ ซึ่งเท่ากับจำนวนวลีที่มีอยู่ในพจนานุกรม โดยในแต่ละรอบจะเริ่มต้นด้วยการดึงข้อมูลทั้งหมดของวลีนั้นขึ้นมาก่อน ด้วยฟังก์ชัน GetDataOfWord (บรรทัดที่ 3) ตัวแปรที่ผ่านเข้าสู่ฟังก์ชันคือค่าตัวแปร w ซึ่งก็คือรหัสของวลีในพจนานุกรมนั่นเอง ข้อมูลที่ดึงออกมานี้คือข้อมูลที่ได้จากขั้นตอนการเตรียมข้อมูล ซึ่งประกอบด้วย รหัสวลี, รหัสกฎหมาย และมาตรา (ในกรณีนี้เราจะไม่นำรหัสคดีความมาพิจารณาด้วย) ข้อมูลที่ได้ออกมาจากฟังก์ชันจะถูกจัดเรียงลำดับตามรหัสกฎหมาย และมาตรา (Class)

ตารางที่ 3. แสดง CARs ระดับมาตรา

Rules	Supt	Confd
ทำร้ายร่างกาย → กฎหมายลักษณะอาญา มาตรา 1	20.00%	33.33%
ทำร้ายร่างกาย → กฎหมายลักษณะอาญา มาตรา 2	20.00%	33.33%
ทำร้ายร่างกาย → กฎหมายวิธีพิจารณาความอาญา มาตรา 3	20.00%	33.33%
ไม่ชำระเงินตามสัญญา → กฎหมายแพ่งและพาณิชย์ มาตรา 4	40.00%	100.00%

สำหรับอัลกอริทึมในการหา CARs สามารถแสดงได้ดังรูปที่ 3 โดยการหา CARs ระดับกฎหมาย และ CARs ระดับมาตราไม่มีความแตกต่างกัน

```
1   f = 0
2   FOR  w = 1  TO  Count(Word) DO
3       GetDataOfWord(w)
4       FOR  i = 1  TO  Count(Data)  DO
5           IF  (i > 1) and (Class_i <> Class_i-1) THEN
6               supt = f / Count(All)*100
7               conf = f / Count(Data) * 100
8               ReturnRule( W → Class , supt, conf)
9               f = 0
10          ELSE
11              f = f+1
12          END
13      END
14  END
```

รูปที่ 3. แสดงอัลกอริทึมของโปรแกรมสร้าง CARs

หลังจากนั้นโปรแกรมจะวนรอบเป็นจำนวน Count(Word) ครั้ง ซึ่งก็คือจำนวนรายการข้อมูลที่ได้จากฟังก์ชัน GetDataOfWord เพื่อหา CARs ของวลีนั้นๆออกมา (บรรทัดที่ 4 ถึง 13) สำหรับการหา CARs ระดับกฎหมายนั้น ให้พิจารณาค่าตัวแปร Class ที่ปรากฏอยู่ในบรรทัด ที่ 5 และบรรทัดที่ 8 เป็นรหัสกฎหมาย โปรแกรมจะวนรอบนับจำนวนรายการแล้วเก็บไว้ที่ตัวแปร f เรื่อยๆ จนกระทั่งเมื่อ Class หรือรหัสกฎหมายของรายการข้อมูลเปลี่ยนไปจากรายการข้อมูลชุดก่อนหน้า (บรรทัดที่ 5) ซึ่งหมายความว่ารายการข้อมูลของรหัสกฎหมายนั้นๆ ได้หมดไปแล้ว และกำลังเริ่มเข้าสู่รายการของรหัสกฎหมายตัวใหม่เนื่องจากว่าข้อมูลมีการเรียงลำดับไว้แล้วตามรหัสกฎหมาย ถึงขั้นตอนนี้โปรแกรมจะสร้างกฎ

ขึ้นมา 1 กฎ (บรรทัดที่ 8) และกำหนดค่าตัวแปร f ให้กลับเป็นค่า 0 อีกครั้งหนึ่ง สำหรับกฎที่โปรแกรมสร้างขึ้นนี้ จะมีค่าฝั่งซ้ายมือเป็น w ซึ่งก็คือรหัสวลีนั่นเอง ส่วนทางฝั่งขวามือของกฎจะมีค่า Class ซึ่งก็คือรหัสกฎหมายนั่นเอง ส่วนค่า Supt สามารถหาได้จาก จำนวนรายการที่นับไว้ที่ตัวแปร f หารด้วย จำนวนรายการข้อมูลทั้งหมดของระบบ คูณด้วย 100 (บรรทัดที่ 6) และค่า Confd สามารถหาได้จาก จำนวนรายการที่นับไว้ที่ตัวแปร f หารด้วย จำนวนรายการของวลีนั้นๆ คูณด้วย 100 (บรรทัดที่ 7)

สำหรับการหา CARs ระดับมาตรา ก็ทำตามอัลกอริทึมที่แสดงไว้เช่นกัน แต่ตัวแปร Class ที่ปรากฏอยู่ในบรรทัด ที่ 5 และ บรรทัดที่ 8 ให้พิจารณาเป็นรหัสมาตราของกฎหมายแทน

4. การสร้างตัวจำแนกจาก Class Association Rules

ขั้นตอนนี้เป็นขั้นตอนการสร้างตัวจำแนก โดยการนำ CARs ของแต่ละวลีมาจัดเรียงตามค่าความสำคัญ (Precedence) จากมากไปน้อย ซึ่งค่าความสำคัญของกฎจะพิจารณาจากค่า Confd และค่า Supt ซึ่งมีรายละเอียดดังนี้

ถ้าค่า Confd ของกฎที่ 1 มากกว่าค่า Confd ของกฎที่ 2 ให้กฎที่ 1 มีค่าความสำคัญมากกว่ากฎที่ 2

ถ้าค่า Confd ของกฎที่ 1 เท่ากับค่า Confd ของกฎที่ 2 ให้พิจารณาค่า Supt ถ้าค่า Supt ของกฎที่ 1 มากกว่าค่า Supt ของกฎที่ 2 ให้กฎที่ 1 มีค่าความสำคัญมากกว่ากฎที่ 2

ถ้าค่า Confd และค่า Supt ของกฎที่ 1 และกฎที่ 2 มีค่าเท่ากัน ให้กฎที่มาก่อนมีค่าความสำคัญมากกว่า

เนื่องจากเราได้จัดเก็บ CARs ไว้ในฐานข้อมูล ทำให้การเรียงลำดับค่าความสำคัญของกฎสามารถทำได้ด้วยภาษา SQL คือ "*SELECT Rules FROM CARs ORDER BY Confd DESC, Supt DESC*"
กฎเกณฑ์ที่ได้ผ่านการจัดเรียง ค่าความสำคัญแล้ว ก็คือตัวจำแนกข้อมูลนั้นเอง ซึ่งใช้สัญลักษณ์ว่า CARsp (Class Association Rules sorted with Precedence)

เมื่อมีคดีความใหม่ ที่ต้องการทำนายว่าใช้กฎหมายฉบับไหนในการพิจารณาเข้าสู่ระบบ ระบบจะทำงาน โดยแบ่งเป็น 2 ขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 ตัดคำคดีความใหม่ที่เข้ามา ด้วยพจนานุกรมวลี เช่นเดียวกับการตัดคำในขั้นตอนการเตรียมข้อมูล ผลลัพธ์ที่ได้จะถูกแปลงเป็นตัวเลขซึ่งแทนรหัสของวลีต่างๆในคดีความนั้น

ขั้นตอนที่ 2 เลือกกฎเกณฑ์ของวลีแต่ละวลีในคดีความ จาก CARsp ออกมา กฎเกณฑ์แต่ละชุดที่ได้นี้จะถูกจัดเรียงลำดับตามค่าความสำคัญ (Precedence) อีกครั้งหนึ่ง เนื่องจากเราได้จัดเก็บ CARs ไว้ในฐานข้อมูล ทำให้การเรียงลำดับค่าความสำคัญของกฎสามารถทำได้ด้วยภาษา SQL เช่นเดิม คือ "*SELECT Rules FROM CARs WHERE Word in (w1,w2,w3,...) ORDER BY Confd DESC, Supt DESC*"

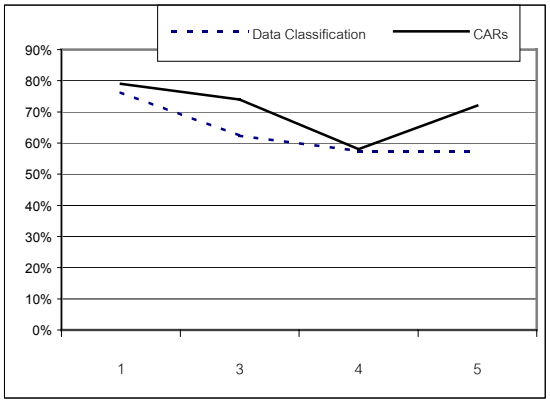
กฎเกณฑ์ที่มีค่าความสำคัญสูงที่สุด (รายการแรกของผลการทำงานตามคำสั่งภาษา SQL) ก็คือผลการทำนายประเภทข้อมูลที่ได้ ซึ่งจะทำให้เรารู้ว่าคดีความนั้นๆ ใช้กฎหมายฉบับไหนในการพิจารณา

5. ผลการทดลอง

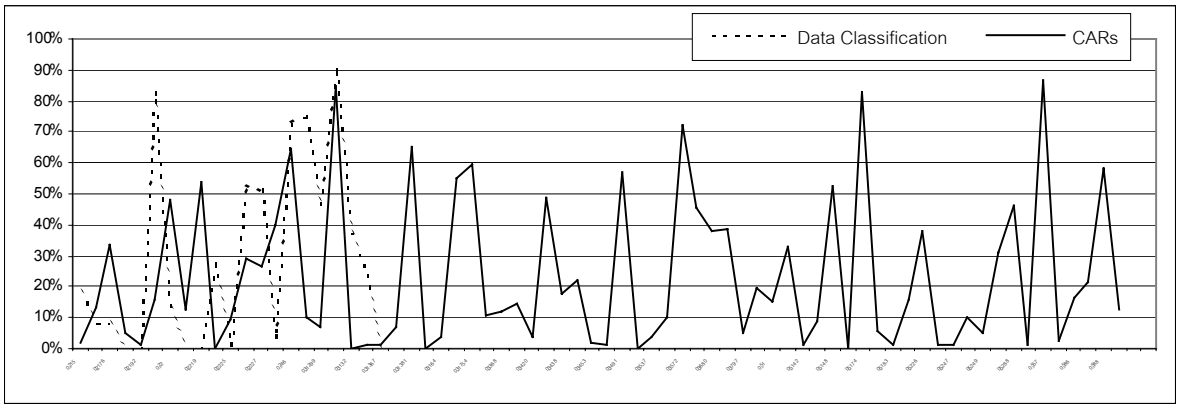
จากข้อมูลที่ผ่านขั้นตอนการเตรียมข้อมูลแล้ว เรานำข้อมูลที่เหลือ 30% มาทดสอบกับระบบใหม่ที่ได้ เพื่อเปรียบเทียบเปอร์เซ็นต์ความถูกต้องกับผลลัพธ์ที่ได้จากระบบเดิมที่มีการสร้างตัวจำแนกด้วยเทคนิค Data Classification แบบทั่วไป ซึ่งก็คือเทคนิค Decision Tree [9] โดยเราแบ่งการทดสอบออกเป็น 2 ชุด คือชุดทดสอบระดับกฎหมาย ซึ่งประกอบด้วยประเภทข้อมูลที่ต้องการจำแนก 4 ประเภท และชุดทดสอบระดับมาตรา ซึ่งประกอบด้วยประเภทข้อมูลที่ต้องการจำแนก 70 ประเภท ผลการเปรียบเทียบแสดงได้ดังกราฟรูปที่ 4.1 และ 4.2 โดยแกนตั้งของกราฟแสดงเปอร์เซ็นต์ความถูกต้องของผลการจำแนกข้อมูล และแกนนอนแสดงรหัสกฎหมาย และมาตราของกฎหมายตามลำดับ

จากกราฟ 4.1 และ 4.2 แสดงให้เห็นว่าการสร้างตัวจำแนก ด้วยเทคนิคใหม่ที่เสนอได้ผลลัพธ์ในการทำนายดีกว่าตัวจำแนกที่สร้างขึ้นด้วยเทคนิค Data Classification แบบทั่วไป โดยการทดสอบในระดับกฎหมาย พบว่าระบบใหม่ที่เสนอ ให้เปอร์เซ็นต์ความถูกต้องสูงกว่าในกฎหมายทุกฉบับ ส่วนในการทดสอบระดับมาตรานั้น ระบบใหม่ที่เสนอได้เปอร์เซ็นต์ความถูกต้องใกล้เคียงกับระบบเดิมเมื่อเทียบมาตราต่อมาตรา แต่ระบบใหม่สามารถจำแนกประเภทข้อมูลได้จำนวนกลุ่มมากกว่า คือสามารถทำนายผลลัพธ์ได้โดยที่เปอร์เซ็นต์ความถูกต้องไม่เป็น 0 เกือบทุกมาตรา ต่างจากระบบเดิมที่สามารถทำนายได้เพียง 20 มาตราจากทั้งสิ้น 70 มาตรา

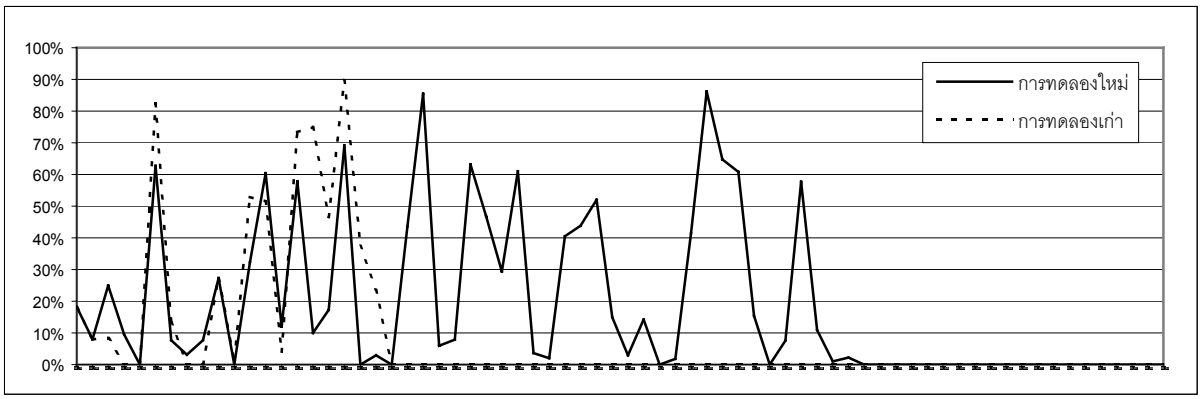
จากกราฟรูปที่ 4.2 เมื่อเปรียบเทียบเปอร์เซ็นต์ความถูกต้องของทั้ง 2 ระบบโดยพิจารณามาตราต่อมาตรา จะพบว่าทั้ง 2 ระบบมีเปอร์เซ็นต์ความถูกต้องสอดคล้องกัน โดยในมาตราที่เปอร์เซ็นต์ความถูกต้องต่ำ ก็จะได้เปอร์เซ็นต์ความถูกต้องต่ำทั้ง 2 ระบบ และในมาตราที่เปอร์เซ็นต์ความถูกต้องสูง ก็จะสูงตามกันทั้ง 2 ระบบเช่นกัน แสดงให้เห็นว่าการทำงานของระบบทั้ง 2 นั้นให้ผลลัพธ์ที่สอดคล้องตรงกัน ในมาตราที่ได้มาตราเปอร์เซ็นต์ความถูกต้องสูง มีสาเหตุจากการที่มาตรานั้นๆ มีวลีที่เด่นชัดแตกต่างจากคำในมาตราอื่นๆ ส่วนในมาตราที่ได้เปอร์เซ็นต์ความถูกต้องต่ำ ก็เนื่องมาจากนั้นๆ ไม่มีวลีเด่น ที่สามารถแยกความแตกต่างจากวลีในมาตราอื่นๆได้เลย



รูปที่ 4.1 แสดงกราฟเปรียบเทียบเปอร์เซ็นต์ความถูกต้องของตัวจำแนกที่สร้างขึ้นด้วยเทคนิค Data Classification ทั่วไป กับตัวจำแนกที่สร้างจาก CARs ระดับกฎหมาย



รูปที่ 4.2 แสดงกราฟเปรียบเทียบเปอร์เซ็นต์ความถูกต้องของตัวจำแนกที่สร้างขึ้นด้วยเทคนิค Data Classification ทั่วไป กับตัวจำแนกที่สร้างจาก CARs ระดับมาตรา

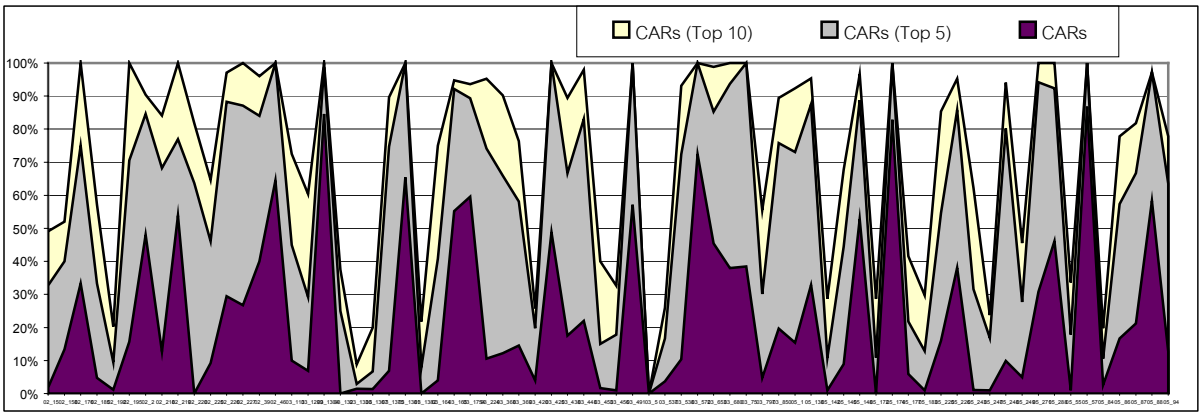


รูปที่ 5. แสดงกราฟเปรียบเทียบเปอร์เซ็นต์ความถูกต้องของตัวจำแนกข้อมูลที่สร้างขึ้นด้วยการตัดคำแบบเดิม กับการตัดคำแบบวลี

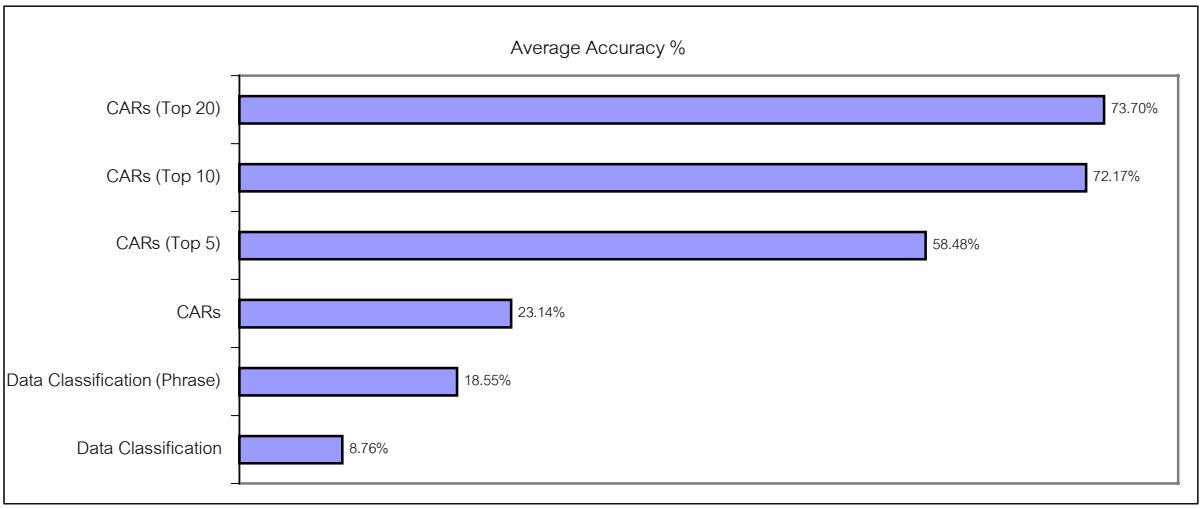
สาเหตุที่ทำให้ระบบใหม่ที่เสนอ มีเปอร์เซ็นต์ความถูกต้องของการทำนายสูงขึ้น เกิดจากปัจจัย 2 ประการ ปัจจัยแรกคือการใช้เทคนิค Association Rule Discovery มาช่วยในการสร้างตัวจำแนก และอีกปัจจัยคือการเปลี่ยนรูปแบบการตัดคำเป็นการตัดวลี เพื่อสนับสนุนสมมติฐานดังกล่าว เราได้ทำการทดลองกับระบบเดิมที่มีการสร้างตัวจำแนกข้อมูลด้วยเทคนิค Data Classification แบบทั่วไปอีกครั้ง แต่ปรับเปลี่ยนฟังก์ชันในการตัดคำ เป็นการตัดวลีแทน ผลลัพธ์ที่ได้ เป็นไปตามกราฟรูปที่ 5 โดยแกนตั้งของกราฟแสดงเปอร์เซ็นต์ความถูกต้องของผลการจำแนกข้อมูล และแกนนอนแสดงกฎหมายมาตราต่างๆ

จากกราฟรูปที่ 5 แสดงให้เห็นว่าการทดลองใหม่ คือการตัดคำแบบวลี ให้เปอร์เซ็นต์ความถูกต้องใกล้เคียงกับการทดลองเก่า เมื่อเปรียบเทียบมาตรา ต่อมาตรา แต่พบว่าในการทดลองใหม่ ตัวจำแนกสามารถจำแนกข้อมูลได้มากประเภทกว่า

เนื่องจากว่ารูปแบบของข้อมูลทางกฎหมาย ที่ต้องการหากฎหมายที่เหมาะสมสำหรับการพิจารณาคดีความ โดยปกติแล้วในแต่ละคดีความจะใช้กฎหมายมากกว่า 1 ฉบับในการพิจารณา ซึ่งไม่สามารถทำได้ด้วยเทคนิค Data Classification แบบทั่วไปดังที่ได้กล่าวมาแล้ว เมื่อเราปรับเปลี่ยนขั้นตอนการสร้างตัวจำแนก มาเป็นการสร้างตัวจำแนกจาก CARs ทำให้สามารถแก้ปัญหาดังกล่าว คือแทนที่จะเลือกกฎเกณฑ์จาก CARsp 1 กฎที่มีค่าความสำคัญสูงสุด ก็เปลี่ยนมาเป็นเลือกกฎเกณฑ์ ครั้งละ 5 หรือ 10 กฎแรกที่มีค่าความสำคัญสูงสุดแทน วิธีนี้ทำให้ระบบสามารถทำนายผลการจำแนกได้ครั้งละมากกว่า 1 ประเภท ซึ่งหมายความว่าระบบได้เสนอกฎหมายมาตราต่างๆ ครั้งละ 5 หรือ 10 มาตรา ที่คิดว่าเหมาะสมสำหรับคดีความนั้นๆ เพื่อให้ผู้ใช้นำไปศึกษาอันทำให้ก่อประโยชน์ต่อไป การทำนายผลเช่นนี้ จะทำให้ผลการทำนายมีเปอร์เซ็นต์ความถูกต้องสูงขึ้น ซึ่งสามารถดูค่าเปรียบเทียบได้จากกราฟ รูปที่ 6



รูปที่ 6. แสดงกราฟเปรียบเทียบเปอร์เซ็นต์ความถูกต้องของตัวจำแนก เมื่อทำนายผลครั้งละ 1, 5 และ 10 มาตรา



รูปที่ 7. แสดงกราฟเปรียบเทียบเปอร์เซ็นต์ความถูกต้องของการทดลองแบบต่างๆ

อีกปัญหาหนึ่งที่ได้กล่าวไว้แล้วในตอนต้น คือเรื่องการแบ่งกลุ่มข้อมูลแบบมีลำดับชั้น ซึ่งไม่สามารถทำได้ในระบบเดิม ทำให้ต้องแบ่งตัวจำแนกข้อมูลเป็น 2 ชุด โดยมีการใช้งานแยกจากกันโดยเด็ดขาด ผู้ใช้ต้องกำหนดว่าจะเลือกใช้ตัวจำแนกชุดไหน สำหรับระบบใหม่ที่เสนอนี้ ถึงแม้ว่าตัวจำแนกข้อมูลจะยังคงต้องแบ่งเป็น 2 ชุดเช่นเดิม แต่ด้วยรูปแบบของการจำแนกที่ใช้การเลือกกฎเกณฑ์ที่มีค่าความสำคัญสูงสุด ขึ้นมาทำนายประเภทข้อมูล ทำให้ตัวจำแนกใหม่นี้ สามารถเลือกกฎเกณฑ์จากตัวจำแนกทั้ง 2 ชุดได้ในขณะเดียวกัน โดยสามารถกำหนดเงื่อนไขได้ว่า ถ้าหากค่าความสำคัญของกฎเกณฑ์ระดับมาตรา มีค่าน้อยเกินไป ก็สามารถเลือกทำนายผลเฉพาะระดับกฎหมายได้โดยอัตโนมัติ

เมื่อเปรียบเปอร์เซ็นต์ความถูกต้องโดยเฉลี่ย ของตัวจำแนกมาตราแบบต่างๆ ให้ผลออกมาตาม กราฟรูปที่ 7 เมื่อแกนนอนคือเปอร์เซ็นต์ความ

ถูกต้องโดยเฉลี่ยของการจำแนก และแกนตั้งคือตัวจำแนกแบบต่างๆ เริ่มต้นจากตัวจำแนกระบบเดิม ที่ถูกสร้างด้วยเทคนิค Data Classification แบบทั่วไป ถัดมาคือตัวจำแนก ที่ได้เปลี่ยนฟังก์ชันการตัดคำเป็นการตัดวลี ถัดมาคือตัวจำแนกที่สร้างขึ้นจาก CARs และที่เหลือคือตัวจำแนกที่สร้างขึ้นจาก CARs เช่นกันแต่มีการทำนายผลครั้งละ 5, 10 และ 20 มาตราตามลำดับ

6. สรุป

จากกราฟรูปที่ 7 แสดงให้เห็นอย่างชัดเจนว่า งานวิจัยนี้สามารถปรับปรุงเทคนิคในการสร้างตัวจำแนก ให้มีประสิทธิภาพ ทางด้านความถูกต้องในการจำแนกเพิ่มมากขึ้น นอกจากนี้ระบบใหม่ยังช่วยแก้ปัญหาต่างๆ ที่เกิดขึ้นจากระบบเดิม ซึ่งประกอบด้วย

1. ปัญหาเรื่องข้อจำกัดทางด้านจำนวนประเภทข้อมูลที่ระบบสามารถจำแนกได้ ซึ่งก็คือจำนวนมาตราที่ระบบสามารถทำนายได้นั่นเอง ในระบบเดิม ตัวจำแนกสามารถทำนายผลได้เพียง 20 มาตราเท่านั้น ซึ่งจากผลการทดลองระบบใหม่ ได้แสดงให้เห็นแล้วว่าระบบใหม่ที่เสนอ สามารถทำนายผลได้ครบทุกมาตรา

2. ปัญหาเรื่องการทำนายผลให้กับคดีความครั้งละมากกว่า 1 มาตรา ซึ่งระบบเดิมไม่สามารถทำได้ จากการทดลองแสดงให้เห็นแล้วว่าระบบใหม่ที่มีการสร้างตัวจำแนกจาก CARs สามารถทำนายผลลักษณะเช่นนี้ได้ โดยสามารถระบุจำนวนมาตราที่ต้องการให้ระบบทำนายได้ นอกจากนี้ด้วยวิธีการดังกล่าว ยังเป็นการเพิ่มเปอร์เซ็นต์ความถูกต้องเฉลี่ยของการทำนายด้วย

3. ปัญหาเรื่องการทำนายผลแบบมีระดับชั้น คือระดับกฎหมายและระดับมาตรา ระบบใหม่ที่เสนอนี้ ถึงแม้ว่าตัวกฎเกณฑ์ หรือ CARs จะถูกจัดเก็บแยกกันเป็น 2 ชุด เช่นเดียวกับระบบเดิม แต่ในขั้นตอนการทำนายผล ระบบสามารถทำนายผลระดับกฎหมาย และระดับมาตราได้ในครั้งเดียว โดยระบบสามารถตรวจสอบได้ว่าถ้าเปอร์เซ็นต์ความถูกต้อง ของผลการทำนายระดับมาตรา มีแนวโน้มที่ต่ำ ก็จะยกเลิกการทำนายระดับมาตรานั้นแล้วทำนายเฉพาะระดับกฎหมาย

ระบบที่เสนอในงานวิจัยนี้ มุ่งเน้นที่จะแก้ไขปัญหาเฉพาะทาง โดยมุ่งเน้นที่จะพัฒนาระบบที่สามารถทำนายกฎหมายที่ใช้สำหรับพิจารณาคดีความโดยอัตโนมัติ ซึ่งทำให้การออกแบบ และเทคนิคที่เสนอมีความเฉพาะเจาะจงสำหรับรูปแบบข้อมูล และวิธีการในการทำงานเช่นนี้ ฉะนั้นการนำเทคนิคที่เสนอในงานวิจัยนี้ไปประยุกต์ใช้กับงานลักษณะอื่นๆ ก็จำเป็นต้องปรับปรุงเทคนิคและวิธีการหลายจุด เพื่อให้ระบบสามารถตอบสนองต่อความต้องการของระบบนั้นๆ ได้

เปอร์เซ็นต์ความถูกต้องที่ได้จากระบบที่เสนอในงานวิจัยฉบับนี้ยังไม่สูงนัก ซึ่งมีแนวโน้มว่าจะสามารถปรับปรุงให้ระบบมีความสามารถเพิ่มมากขึ้นได้ ไม่ว่าจะเป็นการเสนอเทคนิคในการสร้างตัวจำแนกด้วยวิธีอื่นๆ หรือการปรับปรุงขั้นตอนการเตรียมข้อมูลให้มีประสิทธิภาพเพิ่มมากขึ้น

เอกสารอ้างอิง

[1] คณาจารย์คณะนิติศาสตร์มหาวิทยาลัยกรุงเทพ. ความรู้เบื้องต้นเกี่ยวกับกฎหมาย. แผนกตำราและคำสอน มหาวิทยาลัยกรุงเทพ. หน้า 5, 2540.

[2] Witten, I.H., Moffat, A. and Bell, T.C. Managing Gigabytes: Compressing and indexing Documents and Images. Morgan Kaufmann Publishers, page 4, 1999.

[3] K. Waiyamai and L. Lakhal 2000. Knowledge Discovery Very Large Database using Frequent Concept Lattices. Springer-Verlag Lecture Notes in Artificial Intelligent, pages 437-445, June 1810.

[4] M. S. Chen, J. Han, and P. S. Yu. Data Mining: An overview from a database perspective. IEEE Trans. Knowledge and Data Engineering, 8:866-883, 1996.

[5] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. Communications of ACM, 39:58-64, 1996.

[6] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advance in knowledge Discovery and Data Mining. AAAi/MIT Press, 1996.

[7] M. Berry and G. Linoff. Data Mining Techniques: For Marketing Sales and Customer Support, John Wiley & Sons, 1997.

[8] P.K.Chan and S.J.Stolfo. Learning arbiter and combiner trees from partitioned data for scaling machine learning. In Proc. 1st.Int.Conf. Knowledge Discovery and Data Mining (KDD'95), pages 39-44, Montreal, Canada, August 1995.

[9] J.Gehrke, R.Ramakrishnan, and V.Ganti. Rainforest: A framework for fast decision tree construction of large datasets. In Proc. 1998 Int. Conf. Very Large Datacase, pages 416-427, New York, NY, August 1998.

[10] K. Waiyamai 1998. A novel Approach for Association Rule Discovery. The National Computer Science and Engineering Conferrence (NCSEC'98). Bangkok, Thailand, 19-21 October.

[11] R. Agrawal, T. Imielinski, and A. Sawami. Mining Association Rules between sets of items in large database. SIGMOD'93, pages 207-216, Washington,D.C.

[12] R. Feldman & I. Dagan, "Knowledge Discovery in Textual Database." In Proceedings of the International Conference on Knowledge Discovery and Data Mining, 1995.

[13] T. Pongsiripreeda and K. Waiyamai. Using Data Mining technique to select the laws and articles for lawsuits (in thai language). The National Computer Science and Engineering Conferrence (NCSEC'2000). Bangkok, Thailand, November.

[14] B.Liu and W.Hsu and Y.Ma. Integrating Classification and Association Rule Mining. In Proc. 1998 Int. Conf. KDD'98, 1998.

[15] J.L. Bentley. Programming Pearls. Addision-Wesley, pages 164-167, May 1989.

กฤษณะ ไวยมัย จบปริญญาตรีและปริญญาโททางวิทยาศาสตร์คอมพิวเตอร์ จาก University of Picardie ประเทศฝรั่งเศส จบปริญญาเอกทางด้านวิทยาการคอมพิวเตอร์จาก University of Clermont ประเทศฝรั่งเศส งานวิจัยหลักประกอบด้วยสามส่วนสำคัญ ส่วนแรกคือ ระบบสืบค้นความรู้บนฐานข้อมูลขนาดใหญ่ (Knowledge Discovery from very large Databases: Data Mining) ส่วนที่สองคือ ระบบสารสนเทศ (Information System) และ ส่วนที่สามคือ ระบบฐานข้อมูล (Database Management System) ปัจจุบัน ดร.กฤษณะ ไวยมัย ดำรงตำแหน่งอาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

ธีระวัฒน์ พงษ์ศิริปรีดา จบปริญญาตรีทางวิทยาศาสตร์คอมพิวเตอร์ จากมหาวิทยาลัยธรรมศาสตร์ จบปริญญาโททางด้านวิศวกรรมคอมพิวเตอร์จากมหาวิทยาลัยเกษตรศาสตร์ งานวิจัยหลักคือระบบสืบค้นความรู้บนฐานข้อมูลขนาดใหญ่ (Knowledge Discovery from very large Databases: Data Mining) ปัจจุบันเป็นพนักงานบริษัท เอเทรียม เทคโนโลยี จำกัด ในตำแหน่ง System Analyst