รหัสโครงการ SUT7-705-46-12-37

รายงานการวิจัย

การศึกษาผลของการลดขนาดของข้อมูลในกระบวนการทำเหมืองข้อมูล

(The effect of data reduction in the process of data mining)

ผู้วิจัย

หัวหน้าโครงการ

ผู้ช่วยศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ

สาขาวิชาวิศวกรรมคอมพิวเตอร์

สำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ได้รับทุนอุดหนุนการวิจัยจากมหาวิทยาลัยเทคโนโลยีสุรนารี ปีงบประมาณ พ.ศ. 2546

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

มิถุนายน 2548

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณมหาวิทยาลัยเทคโนโลยีสุรนารี ที่ได้จัดสรรงบประมาณให้ในปีงบประมาณ 2546 เพื่อสนับสนุนการวิจัยนี้ งานวิจัยนี้สำเร็จได้อย่างดีด้วยการมีส่วนร่วมจากอาสาสมัครจำนวนมาก ได้แก่ นักศึกษาวิศวกรรมคอมพิวเตอร์รุ่นที่ 3, 4 และ 5 ที่มีโอกาสได้เรียนรายวิชาเลือก Knowledge Discovery and Data Mining โดยมีส่วนร่วมในการสุ่มข้อมูลรวมถึงการทดสอบอัลกอริทึมการทำเหมืองข้อมูลเพื่อยืนยันผลลัพธ์กับรายงานฉบับนี้ การสุ่มข้อมูลด้วยเทคนิค progressive sampling ได้รับความร่วมมือทดสอบข้อมูลจากนายจักนรินทร์ คงเจริญ และนางสาวทิพยา ทิพยโสต นักศึกษาปริญญาโทสาขาวิศวกรรมคอมพิวเตอร์ รุ่นที่ 1

บทคัดย่อภาษาไทย

การทำเหมืองข้อมูล หรือการค้นหาความรู้ เป็นกระบวนการคัดแยกข้อสนเทศที่มีประโยชน์และยังไม่ถูกค้นพบมาก่อนออกจากเซตหรือกลุ่มข้อมูลขนาดใหญ่ แต่เนื่องจากกระบวนการนี้ใช้เวลาในการประมวลผลมากโดยเฉพาะเมื่อข้อมูลมีขนาดใหญ่ การใช้กลุ่มตัวอย่างแทนที่จะใช้ข้อมูลทั้งหมดจะช่วยให้การประมวลผลรวดเร็วขึ้น แต่ทั้งนี้ผลลัพธ์ที่ได้จะต้องมีคุณภาพคงเดิม งานวิจัยนี้จึงมีจุดมุ่งหมายที่จะศึกษาพฤติกรรมของอัลกอริทึมค้นหาความรู้ เมื่อกลุ่มข้อมูลมีขนาดลดลงตามลำดับจากการสุ่มข้อมูล ทั้งนี้เพื่อค้นหาขนาดของกลุ่มข้อมูลที่ให้ผลลัพธ์ใกล้เคียงที่สุดกับผลลัพธ์ที่ได้จากข้อมูลประชากรทั้งหมด อัลกอริทึมที่ใช้ในการสังเคราะห์ความรู้ได้แก่ อัลกอริทึมเบย์อย่างง่าย และอัลกอริทึมสร้างต้นไม้ตัดสินใจ ซึ่งจัดอยู่ในอัลกอริทึมประเภทค้นหากฎที่สามารถจำแนกข้อมูลและสามารถอธิบายข้อมูลแต่ละประเภทหรือแต่ละคลาสได้ เทคนิคการสุ่มข้อมูลที่ใช้ในการศึกษานี้ได้แก่ การสุ่มอย่างง่าย การสุ่มแบบเป็นระบบ การสุ่มแบบแบ่งชั้น การสุ่มแบบก้าวหน้าเชิงเลขคณิต และการสุ่มแบบก้าวหน้าเชิงเรขาคณิต ข้อมูลที่ใช้ในการศึกษาวิจัยนี้มีจำนวน 5 ชุด แต่ละชุดแยกเป็นข้อมูลฝึกและข้อมูลทดสอบ โดยข้อมูลเหล่านี้เป็นข้อมูลมาตรฐานจากแหล่งข้อมูลมหาวิทยาลัยแคลิฟอร์เนียเมืองเออร์ไวน์ การทดสอบคุณภาพของกลุ่มตัวอย่างจะใช้วิธีทดสอบความแม่นตรงของโมเดลซึ่งเป็นผลลัพธ์ที่ได้จากอัลกอริทึมค้นหาความรู้

บทคัดย่อภาษาอังกฤษ

Data mining or knowledge discovery is the process of extracting useful and previously unknown information from the very large data set. However, extracting knowledge from a large data set is computationally inefficient. Using a sample from the original data can speed up the data mining process, but this is only acceptable if it does not reduce the quality of the induced information. We thus investigate the behavior of learning algorithms on decreasing sample sizes to decide which sample is sufficiently similar to the original data. We observe the accuracy of the induced classification rules extracted from training samples of various sizes and use these results to determine when a sample is sufficiently small, yet maintain the acceptable accuracy rate. We evaluate four sampling methods: simple random, systematic random, stratified random, arithmetic progressive, and geometric progressive. The five data sets to be sampled are taken from the UCI repository and the learning algorithms to induce knowledge from each sample are naive Bayes and decision tree induction. The performance of each sampling scheme is evaluated on the basis of the induced-model accuracy tested on the supplied test data.

สารบัญ

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1
บทนำ

การทำเหมืองข้อมูลเป็นวิธีการวิเคราะห์ข้อมูลอย่างฉลาด (intelligent data analysis) โดยเน้นที่การวิเคราะห์ข้อมูลปริมาณมาก และการวิเคราะห์ต้องทำได้โดยอัตโนมัติจึงจะเรียกได้ว่าเป็นการวิเคราะห์อย่างฉลาด แต่ปัญหาของการวิเคราะห์ข้อมูลปริมาณมากคือต้องใช้เวลานานมากในการประมวลผล หรือในกรณีเลวร้ายกว่านั้นข้อมูลมีปริมาณมากเกินกว่าที่จะอ่านมาเก็บไว้ในหน่วยความจำของคอมพิวเตอร์ ทำให้การวิเคราะห์ข้อมูลไม่สามารถทำได้ วิธีการแก้ไขคือต้องลดขนาดของข้อมูล งานวิจัยนี้จึงมุ่งศึกษาในประเด็นเทคนิคการลดขนาดข้อมูลว่าควรจะลดขนาดด้วยเทคนิคใด และลดลงด้วยสัดส่วนเท่าใดจึงจะยังคงให้ผลการวิเคราะห์ที่เชื่อถือได้

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

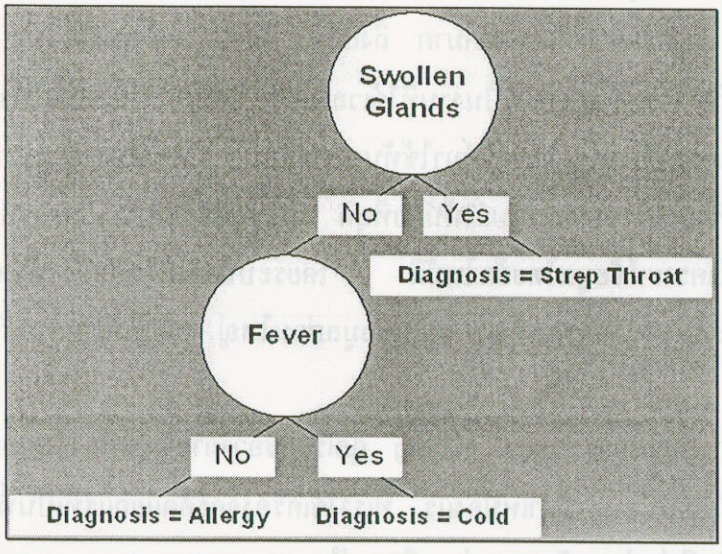
การทำเหมืองข้อมูล (data mining) หรือในบางครั้งเรียกว่าการค้นหาความรู้ (knowledge discovery) คือ กระบวนการค้นหาแนวโน้ม รูปแบบร่วม ความสัมพันธ์ หรือความรู้ใหม่อื่นๆ จากข้อมูลจำนวนมาก data mining จัดได้ว่าเป็นเทคโนโลยีใหม่ที่ช่วยให้การวิเคราะห์ข้อมูลทำได้โดยอัตโนมัติ และมีประสิทธิภาพสูงขึ้นกว่าที่เคยเป็นมา จึงได้รับความสนใจนำไปใช้อย่างแพร่หลายในทุกวงการ โดยเฉพาะในกรณีที่ข้อมูลมีขนาดใหญ่มาก เช่นข้อมูลสำมะโนประชากร ข้อมูลที่ได้รับจากดาวเทียมสำรวจสภาพอากาศและพื้นผิวโลก ปัญหาของข้อมูลขนาดใหญ่เหล่านี้คือ การวิเคราะห์ข้อมูลด้วยแรงงานของนักวิเคราะห์ข้อมูลเป็นสิ่งที่แทบจะเป็นไปไม่ได้ เพราะเป็นงานที่ใช้แรงงาน ทรัพยากรและเวลามาก ถึงแม้จะมีโปรแกรมทางสถิติ เช่น SPSS เป็นเครื่องมือช่วย แต่การวิเคราะห์ข้อมูลก็ยังเป็นงานที่ใช้เวลามาก แนวทางที่จะช่วยให้งานวิเคราะห์ข้อมูลทำได้รวดเร็วขึ้นและได้ผลการวิเคราะห์มาใช้ทันเวลาก็คือ ทำให้กระบวนการวิเคราะห์ข้อมูลเป็นอัตโนมัติมากขึ้น ใช้แรงงานมนุษย์ให้น้อยที่สุด แนวคิดนี้จึงทำให้เกิดการทำเหมืองข้อมูลซึ่งเป็นกระบวนการวิเคราะห์ข้อมูลโดยอัตโนมัติ โดยระบบคอมพิวเตอร์จะทำหน้าที่ค้นหาแนวโน้ม, ค้นหาลักษณะที่น่าสนใจต่างๆ ที่ปรากฏในข้อมูลส่วนใหญ่ และค้นหาความสัมพันธ์ภายในรายการข้อมูล

กระบวนการ data mining ถูกเรียกชื่อว่าการทำเหมืองข้อมูลเพราะการทำงานของโปรแกรมเปรียบเสมือนการทำเหมืองแร่ ที่เราใช้เครื่องจักรคัดแยกแร่ที่เป็นที่ต้องการออกจากกองหิน กรวด ดินที่ปะปนมากับสายแร่ เพียงแต่ในกระบวนการ data mining สิ่งที่เราได้จากกองข้อมูลมหาศาล คือ ความรู้ (knowledge) ที่ซ่อนอยู่ในกองข้อมูล ความรู้นี้จะช่วยให้เราเข้าใจ

ลักษณะของข้อมูล และเข้าใจปัจจัยที่ทำให้เกิดลักษณะบางอย่างขึ้นในข้อมูลบางกลุ่ม ซึ่งจะช่วยให้เราสามารถทำนายแนวโน้มของข้อมูลใหม่ที่จะเกิดขึ้นในอนาคตได้ รวมถึงเข้าใจความสัมพันธ์ที่เชื่อมโยงข้อมูลแต่ละกลุ่มย่อยเข้าด้วยกัน ตัวอย่างเช่น จากบันทึกประวัติการตรวจรักษาคนไข้ในอดีตจำนวน 10 คน (รูปที่ 1.1) ที่มีกลุ่มอาการพื้นฐานคล้ายกันซึ่งแพทย์ได้วินิจฉัยแล้วว่าคนไข้แต่ละรายป่วยด้วยโรคหวัด (cold), ภูมิแพ้ (allergy) หรือเป็นการติดเชื้อ (strep throat) ข้อมูลในอดีตเหล่านี้จะทำหน้าที่เป็นข้อมูลฝึกเพื่อช่วยให้โปรแกรม data mining สามารถสร้างโมเดล (รูปที่ 1.2) ที่แสดงรูปแบบกลุ่มอาการคนไข้ที่ป่วยด้วยไข้หวัด ภูมิแพ้ และการติดเชื้อ (การทำ data mining แบบนี้เรียกว่า การทำ classification โดยมีคลาสของข้อมูล 3 คลาส คือ cold, allergy, strep throat ผลลัพธ์จะเป็นโมเดลที่อธิบายลักษณะของข้อมูลแต่ละคลาส และสามารถใช้โมเดลทำนายการวินิจฉัยโรคให้กับคนไข้ได้)

Patient ID#	Sore Throat	Fever	Swollen Glands	Congestion	Headache	Diagnosis
1	Yes	Yes	Yes	Yes	Yes	Strep throat
2	No	No	No	Yes	Yes	Allergy
3	Yes	Yes	No	Yes	No	Cold
4	Yes	No	Yes	No	No	Strep throat
5	No	Yes	No	Yes	No	Cold
6	No	No	No	Yes	No	Allergy
7	No	No	Yes	No	No	Strep throat
8	Yes	No	No	Yes	Yes	Allergy
9	No	Yes	No	Yes	Yes	Cold
10	Yes	Yes	No	Yes	Yes	Cold

รูปที่ 1.1 ข้อมูลคนไข้ที่ได้รับการวินิจฉัยโรคแล้วรวม 10 คน



รูปที่ 1.2 โมเดลของอาการคนไข้แต่ละโรคแสดงในลักษณะของต้นไม้ตัดสินใจ

โดยทั่วไปกระบวนการ data mining จะประกอบด้วย 4 ขั้นตอนใหญ่ๆคือ

ขั้นตอนที่ 1 **เตรียมข้อมูล** (data preparation) : ถ้าข้อมูลไม่อยู่ในรูปแบบที่ถูกต้องหรือเหมาะสม จะต้องมีการปรับข้อมูลให้อยู่ในรูปแบบที่โปรแกรม data mining จะเรียกใช้งานได้

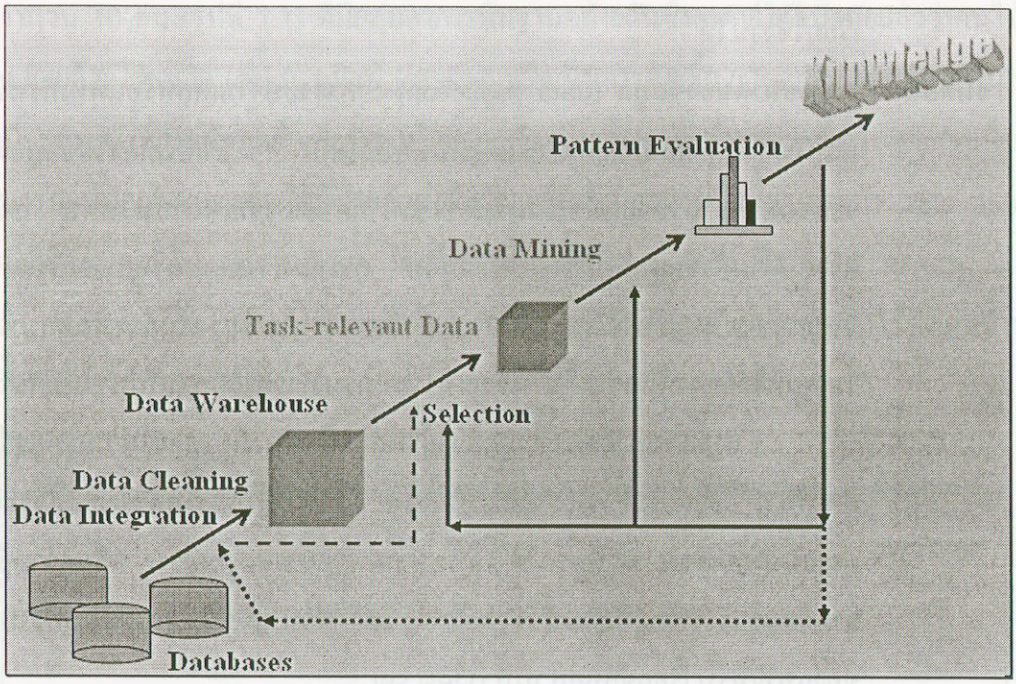
ขั้นตอนที่ 2 **ลดขนาดของข้อมูล** (data reduction) : การจะหาโมเดลหรือแพทเทิร์นที่ข้อมูลส่วนใหญ่แสดงลักษณะเหล่านั้นออกมาเหมือนกัน จำเป็นต้องใช้ข้อมูลตัวอย่างจำนวนมาก ถ้าข้อมูลน้อยเกินไปอาจจะหาลักษณะร่วมเหล่านั้นไม่พบ แต่ในทางตรงกันข้ามถ้าข้อมูลมีปริมาณมากเกินไป การค้นหาโมเดลหรือแพทเทิร์นจากกลุ่มข้อมูลขนาดใหญ่ต้องใช้เวลามาก ซึ่งถ้าลดจำนวนข้อมูลลงด้วยสัดส่วนที่ถูกต้องโมเดลที่ได้ยังคงเป็นเช่นเดิมในขณะที่โปรแกรมใช้เวลาในการค้นหาโมเดลสั้นลง

การลดขนาดของข้อมูลทำได้ในสองลักษณะคือ ลดจำนวนเรคคอร์ด และลดจำนวน attribute ของแต่ละเรคคอร์ด ข้อมูลที่ผ่านการลดขนาดแล้วจะถูกแบ่งออกเป็นสองส่วน ส่วนแรกใช้ในกระบวนการค้นหาแพทเทิร์น หรือความสัมพันธ์จากข้อมูล เรียกข้อมูลส่วนนี้ว่า training set ส่วนที่สองใช้ตรวจสอบความถูกต้องของแพทเทิร์น เรียกข้อมูลส่วนนี้ว่า test set

ขั้นตอนที่ 3 **ค้นหาโมเดล** (หรือความสัมพันธ์) จากข้อมูล (data modeling/discovery) : กระบวนการค้นหาโมเดลหรือความสัมพันธ์จะเริ่มจากข้อมูลเริ่มต้นจำนวนไม่มากนัก จากนั้นนำผลที่ได้จากกระบวนการค้นหา (learning process/method) ไปยืนยันกับข้อมูลทดสอบ ถ้าผลที่ได้ยังไม่น่าพอใจอาจจะต้องปรับค่าพารามิเตอร์บางตัวของ learning method และเริ่มกระบวนการค้นหาใหม่กับข้อมูลจำนวนมากขึ้นจนกว่าผลที่ได้มีความถูกต้องอยู่ในระดับที่ยอมรับได้ จึงจะจบกระบวนการค้นหา

ขั้นตอนที่ 4 **ตรวจสอบและวิเคราะห์ผล** (solution analyses) : โมเดลหรือความสัมพันธ์ที่หามาได้ในขั้นตอนที่ 3 จะต้องถูกนำมาทดสอบอัตราความผิดพลาดและวิเคราะห์ความซับซ้อนของรูปแบบโมเดล ถ้าอัตราความผิดพลาดยังสูงเกินไป อาจจะต้องย้อนกลับไปที่ขั้นตอนที่ 3 อีกครั้ง เพื่อปรับปรุงโมเดลให้ถูกต้องยิ่งขึ้น ในทำนองเดียวกันถ้าโมเดลที่หามาได้มีรูปแบบที่ซับซ้อนเกินไปจนยากต่อการทำความเข้าใจ อาจจะต้องย้อนกระบวนการกลับไปที่ขั้นตอนที่ 3 เพื่อให้หาโมเดลใหม่ที่มีความถูกต้องเท่าเดิมแต่มีรูปแบบที่ซับซ้อนน้อยลง

ขั้นตอนทั้งสี่อาจจะไม่ได้ทำสิ้นสุดในคราวเดียว ถ้าผลลัพธ์ที่ได้ยังไม่ถึงเกณฑ์ที่น่าพอใจสามารถมีการย้อนกลับได้ในบางขั้นตอน หรือย้อนกลับไปตั้งต้นตั้งแต่ขั้นตอนแรกใหม่ก็ได้ วัฏจักรของการทำ data mining แสดงเป็นแผนภาพได้ดังรูปที่ 1.3



รูปที่ 1.3 วัฏจักรของกระบวนการทำเหมืองข้อมูล

จากทั้งสี่ขั้นตอนของกระบวนการทำเหมืองข้อมูลที่กล่าวมาข้างต้น ขั้นตอนของการค้นหาโมเดล (ขั้นตอนที่ 3) เป็นหัวใจสำคัญของกระบวนการ แต่ทั้งนี้โมเดลหรือความรู้ที่ค้นหามาได้จะมีคุณภาพและเป็นประโยชน์เพียงใดนั้นปัจจัยสำคัญที่สุดคือข้อมูล คุณภาพและจำนวนของข้อมูลจะมีผลโดยตรงต่อผลลัพธ์ที่จะได้จากการทำ data mining

คุณภาพของข้อมูลสะท้อนมาจากลักษณะต่างๆ (attribute or feature) ที่ประกอบกันขึ้นเป็นข้อมูลหนึ่งรายการ ถ้าลักษณะที่รวบรวมมาเป็นลักษณะหลักที่จะสามารถตัดสินพฤติกรรมของข้อมูล ผลลัพธ์ของการทำ data mining ก็จะถูกต้อง นอกจากนี้จำนวนของข้อมูลจะเป็นอีกปัจจัยสำคัญที่ช่วยยืนยันว่าผลลัพธ์ที่ได้นั้นมีความเที่ยงตรงสูงเพียงใด ในทางอุดมคติจำนวนข้อมูลยิ่งมากเท่าไรจะช่วยให้ผลลัพธ์มีความถูกต้องน่าเชื่อถือมากขึ้นเท่านั้น แต่ในทางปฏิบัติแล้วเนื่องจากทรัพยากรของระบบคอมพิวเตอร์ (ขนาดของหน่วยความจำและความเร็วของหน่วยประมวลผล) มีจำกัด โดยเฉพาะในสภาพการณ์ของประเทศไทยที่ซุปเปอร์คอมพิวเตอร์เป็นทรัพยากรที่หาได้ยากหรือแทบจะเป็นไปไม่ได้ที่จะจัดหาทรัพยากรในระดับนี้ การพยายามลดจำนวนข้อมูลให้สามารถประมวลผลได้ในเครื่องคอมพิวเตอร์ส่วนบุคคลจึงเป็นแนวทางแก้ปัญหาที่เหมาะสมกว่า

ดังนั้นโครงการวิจัยนี้จึงมุ่งเน้นที่การศึกษาผลของการลดขนาดของข้อมูลเพื่อใช้ในงานทำเหมืองข้อมูลว่าเราสามารถลดจำนวนข้อมูลลงได้มากที่สุดถึงจุดใด โดยที่คุณภาพของผลลัพธ์หรือโมเดลที่ได้จากกระบวนการ data mining ยังคงเดิม หรืออยู่ในระดับของความถูกต้องที่ยังยอมรับได้ ผลที่ได้รับตอบแทนจากการลดขนาดของข้อมูลคือ สามารถทำ data mining ได้ในเวลาที่รวดเร็วขึ้น หรือในกรณีที่ข้อมูลเริ่มแรกมีขนาดใหญ่จนไม่สามารถประมวลผลได้ในคอมพิวเตอร์ส่วนบุคคล การลดขนาดของข้อมูลจะทำให้การทำ data mining เป็นสิ่งที่เป็นไปได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อศึกษาผลของการลดขนาดของข้อมูลว่าให้คุณภาพของโมเดลที่สังเคราะห์ขึ้นจากข้อมูลขนาดต่างๆ กันแตกต่างกันหรือไม่ เพียงใด รวมถึงสังเกตเวลาที่ใช้ในการสังเคราะห์โมเดลจากข้อมูลขนาดต่างๆ กัน การศึกษาคุณภาพและเวลาที่สังเคราะห์โมเดลนี้จะเปรียบเทียบกับข้อมูลเริ่มต้นที่ยังไม่ได้มีการลดขนาด ผลของการทดสอบเปรียบเทียบจะนำไปสู่เกณฑ์ที่จะช่วยในการพิจารณาว่า การทำ data mining แต่ละครั้งนั้นควรจะใช้ข้อมูลขนาดเท่าไรจึงจะได้ผลลัพธ์ที่มีคุณภาพดีภายในเวลาที่เหมาะสม

1.3 ขอบเขตของการวิจัย

โครงการวิจัยที่เสนอนี้มุ่งไปที่การศึกษาการสุ่มด้วยวิธีการต่างๆ เพื่อลดขนาดข้อมูล เทคนิคในการสุ่มจะใช้ 5 เทคนิค ได้แก่ simple random sampling, systematic random sampling, stratified random sampling, arithmetic progressive sampling และ geometric progressive sampling การศึกษาจะวิเคราะห์พฤติกรรมของอัลกอริทึมสังเคราะห์ความรู้เมื่อข้อมูลมีขนาดลดลง เกณฑ์ในการสังเกตพฤติกรรมจะใช้ (1) โมเดลหรือความรู้ที่สังเคราะห์ได้โดยจำกัดขอบเขตของงานวิเคราะห์ไปที่การจำแนกประเภท (classification) ซึ่งเป็นงานวิเคราะห์และทำนายข้อมูลที่ใช้มากที่สุดในการทำ data mining (2) ความแม่นตรง (accuracy) ในการใช้โมเดลวิเคราะห์กับข้อมูลทดสอบ และ (3) เวลาที่ใช้ในการสังเคราะห์โมเดล เมื่อข้อมูลมีขนาดแตกต่างกัน

เทคนิคในการทดสอบความแม่นยำของโมเดล จะใช้เทคนิค Holdout นั่นคือจะแบ่งข้อมูลเริ่มต้นออกเป็นสองส่วน ส่วนแรกจะกันไว้เป็นข้อมูลทดสอบ (test data) ส่วนที่เหลือจะเป็นข้อมูลที่อัลกอริทึมใช้เพื่อสังเคราะห์ความรู้ เรียกว่า ข้อมูลฝึก (train data) การสุ่มจะกระทำกับข้อมูลฝึก และโมเดลที่สังเคราะห์ได้จะถูกทดสอบด้วยข้อมูลทดสอบ

1.4 ประโยชน์ที่ได้รับจากการวิจัย

ผลของการศึกษานี้จะเป็นประโยชน์อย่างยิ่งสำหรับการเตรียมข้อมูลเพื่อใช้ในงานทำเหมืองข้อมูล โดยเฉพาะในกรณีที่ข้อมูลมีขนาดใหญ่มากในระดับจิกะไบต์ขึ้นไปจนถึงระดับเทอราไบต์ โปรแกรม data mining ในปัจจุบันไม่สามารถรองรับข้อมูลในขนาดนี้ได้ และข้อมูลที่จำเป็นต้องได้รับการวิเคราะห์อัตโนมัติ มักจะเป็นข้อมูลที่มีขนาดใหญ่เกินกว่าจะวิเคราะห์ด้วยแรงงานผู้เชี่ยวชาญ เช่น ข้อมูลสำมะโนประชากร ข้อมูลพันธุกรรมมนุษย์ การลดขนาดของข้อมูลจึงเป็นแนวทางเดียวที่จะช่วยให้การวิเคราะห์อัตโนมัติด้วยโปรแกรม data mining เป็นไปได้ การศึกษาวิจัยนี้จึงเป็นประโยชน์กับทุกวงการที่จะพิจารณานำเทคนิค data mining ไปใช้

นอกจากนี้องค์ความรู้ที่ได้จากการศึกษานี้ สามารถนำไปสู่การพัฒนาโปรแกรมที่จะช่วยในการเตรียมและลดขนาดของข้อมูลเพื่อเข้าสู่ขั้นตอน data mining ได้ต่อไปในอนาคต ซึ่งในปัจจุบันโปรแกรม data mining ที่มีความสามารถสูงและรองรับข้อมูลขนาดใหญ่ได้จะต้องซื้อในราคาที่แพง การศึกษาเพื่อนำไปสู่การพัฒนาโปรแกรม data mining ขีดความสามารถสูง จึงจะช่วยให้ประเทศชาติประหยัดค่าใช้จ่ายในด้านซอฟต์แวร์ได้มาก

หน่วยงานที่สามารถใช้ประโยชน์จากผลการวิจัยนี้

- หน่วยงานการศึกษาและวิจัย ในสาขาที่ต้องวิเคราะห์ข้อมูลจำนวนมาก
- หน่วยงานภาคธุรกิจ เช่น ธนาคาร ธุรกิจประกันภัย
- หน่วยงานภาครัฐ เช่น สำนักงานสถิติแห่งชาติ
- หน่วยงานภาคอุตสาหกรรม ที่ต้องมีการวิเคราะห์และวางแผนการผลิต

บทที่ 2

เทคนิคการลดขนาดของข้อมูล

การลดขนาดของข้อมูลทำได้ในสองมิติคือลดแอททริบิวต์และลดเรคคอร์ด การลดแอททริบิวต์เป็นการลดลักษณะที่ประกอบกันเป็นข้อมูลหนึ่งรายการ ในขณะที่การลดเรคคอร์ดเป็นการลดจำนวนรายการข้อมูล งานวิจัยนี้มุ่งความสนใจที่การลดเรคคอร์ดด้วยเทคนิคหลักคือการสุ่มข้อมูลตัวอย่างเพื่อใช้เป็นตัวแทนข้อมูลทั้งหมด เนื้อหาในบทนี้เป็นการอธิบายวิธีการต่างๆที่ใช้ในการสุ่มข้อมูล

2.1 การลดขนาดของข้อมูลในงานเหมืองข้อมูล

เทคนิคการลดขนาดของข้อมูลมักจะกระทำในสองแนวทางคือ ลดลักษณะหรือแอททริบิวต์ (attribute) ที่อธิบายข้อมูลแต่ละรายการให้เหลือเฉพาะลักษณะหลักที่สำคัญ งานวิจัยในแนวทางนี้จะอยู่ในกลุ่มที่เรียกว่า feature subset selection รายละเอียดปรากฏในหนังสือและงานวิจัยจำนวนมาก (Neter et al., 1996; Dash & Liu, 1997; Liu & Motoda, 1998; Kohavi & John, 1997; Dash et al.,1997)

แนวทางที่สองในการลดขนาดของข้อมูล ซึ่งเป็นแนวทางหลักของโครงการวิจัยนี้ ใช้วิธีการลดจำนวนรายการข้อมูลหรือเรคคอร์ด ให้เหลือเฉพาะข้อมูลที่สามารถเป็นตัวแทนของข้อมูลกลุ่มใหญ่ได้ เทคนิคที่นิยมใช้ลดจำนวนข้อมูลคือ การสุ่ม (sampling) การจัดกลุ่ม(clustering) และการใช้ฮิสโตแกรม(histogram) เทคนิคที่นิยมใช้มากที่สุดคือการสุ่ม โดยเฉพาะการสุ่มอย่างง่าย (simple random sampling)

John และ Langley (1996) ได้ศึกษา static sampling และ dynamic sampling โดยใช้อัลกอริทึม naïve Bayes ในการสังเคราะห์โมเดลจากข้อมูล 11 กลุ่ม แต่ไม่มีการปรับขนาดของข้อมูลในแต่ละกลุ่ม ในขณะที่ Josien และคณะ (2001) ใช้ข้อมูลเพียงกลุ่มเดียวแต่ปรับขนาดของข้อมูลให้มี 5 ขนาด และวิเคราะห์ข้อมูลด้วยอัลกอริทึม fuzzy clustering นักวิจัย Kivinen และ Mannila (1993) ได้วิเคราะห์ในเชิงทฤษฎีถึงจำนวนข้อมูลที่สามารถเป็นตัวแทนของข้อมูลกลุ่มใหญ่ได้แต่ยังขาดการยืนยันด้วยผลการทดลองจริง Barbara และคณะ (1997) รามถึง Devore และ Peck (1997) ได้นำเทคนิคฮิสโตแกรมมาใช้วิเคราะห์การกระจายของข้อมูลเพื่อสุ่มข้อมูลจากแต่ละกลุ่ม

โครงการวิจัยนี้ใช้แนวทางการสุ่มข้อมูลเป็นหลายขนาดโดยมีข้อมูลเริ่มต้นเป็นพื้นฐานในการเปรียบเทียบผล และขยายขอบเขตของการทดลองเพิ่มจากที่ John และ Langley (1996) และ Josien พร้อมคณะ (2001) ได้ทำ โดยใช้อัลกอริทึมมากกว่าหนึ่งอัลกอริทึม และเป็นอัลกอริทึมที่มีเทคนิคพื้นฐานแตกต่างกัน ทั้งนี้เพื่อยืนยันผลว่าขนาดของข้อมูลที่เหมาะสมเป็นตัวแทนของข้อมูลกลุ่มใหญ่จะให้ผลการสังเคราะห์โมเดลที่สอดคล้องกันในแต่ละอัลกอริทึม

2.2 ความหมายของการสุ่มข้อมูล

การสุ่มข้อมูล คือ ขั้นตอนในการเลือกตัวแทนของข้อมูลที่เป็นสมาชิกของประชากรอย่างมีระบบ เมื่อเลือกตัวอย่างข้อมูลที่ถูกต้องและถูกวิธี เช่น การสัมภาษณ์ หรือ การสังเกต ทำให้แน่ใจได้ว่าข้อมูลที่เป็นตัวอย่างนั้นจะมีประโยชน์ สามารถเป็นตัวแทนของประชากรทั้งระบบได้ และทำให้เกิดประโยชน์หลายประการ เช่น สามารถควบคุมค่าใช้จ่าย ลดเวลาในการเก็บข้อมูล เพิ่มประสิทธิภาพในการวิเคราะห์และลดความเอนเอียงในการเลือกได้

การสุ่มตัวอย่างจากประชากรเรียกว่า random selection ในขณะที่ random assignment เป็นการนำกลุ่มตัวอย่างที่สุ่มได้มาสุ่มเข้ากลุ่มที่แตกต่างกัน เช่น สุ่มเข้ากลุ่มทดลองและกลุ่มควบคุม มีความเป็นไปได้ที่ในการวิจัยเรื่องหนึ่งอาจจะมีการสุ่มทั้ง 2 ประเภท ก็คือ สุ่มกลุ่มตัวอย่าง 100 คน จากประชากรทั้งหมด 1,000 คน (random selection) แล้วนำกลุ่มตัวอย่างที่สุ่มได้ 100 คน มาสุ่มเพื่อจำแนกออกเป็นกลุ่มควบคุม 50 คน และกลุ่มทดลอง 50 คน (random assignment)

มีความเป็นไปได้ที่ในการวิจัยเรื่องหนึ่งอาจจะใช้การสุ่มประเภทใดประเภทหนึ่ง เช่น อาจจะไม่สุ่ม 100 คนจาก 1,000 แต่อาจจะใช้รายชื่อประชากรทั้ง 1,000 คน แล้วเลือกเอาเฉพาะ 100 คนแรก แล้วจึงค่อยใช้การสุ่มจำแนกออกเป็นกลุ่มควบคุม 50 คน และกลุ่มทดลอง 50 คน

นอกจากนี้ยังมีความเป็นไปได้ที่ในการวิจัยเรื่องหนึ่งอาจจะไม่มีการสุ่มทั้ง 2 ประเภท เช่น เจาะจงทดลองกับเด็กนักเรียนชั้นประถมศึกษาปีที่ 6 โรงเรียนวัดหนองปรู ซึ่งมีอยู่ 4 ห้อง เลือกมา 2 ห้องเป็นกลุ่มทดลองโดยใช้หลักสูตรใหม่ ส่วนอีก 2 ห้องที่เหลือใช้หลักสูตรเก่าเป็นกลุ่มควบคุม เป็นต้น

random selection มีความสำคัญที่จะช่วยให้งานวิจัยมีความเที่ยงตรงภายนอก (external validity) นั่นคือสามารถสรุปอ้างอิงผลการวิจัยไปยังประชากรได้ ส่วน random assignment มีความสำคัญในการออกแบบการวิจัยเชิงทดลองเพราะจะช่วยให้เกิดความเที่ยงตรงภายใน (internal validity)

ตัวอย่างร้านขายคุกกี้

ร้านขายคุกกี้แห่งหนึ่ง มีคุกกี้อยู่ 3 ชนิดคือ ช็อกโกแลตชิป เนยถั่ว และมะพร้าว โดยราคาขายจะเป็นไปตามน้ำหนักของคุกกี้แต่ละชิ้น มีลูกค้าคนหนึ่งเลือกคุกกี้ทั้งสามชนิดมา 15 ชิ้น และไปชำระเงิน พนักงานชั่งคุกกี้ทั้ง 15 ชิ้น น้ำหนักของคุกกี้ทั้ง 15 ชิ้นแสดงได้ดังตารางที่ 2.1

ตารางที่ 2.1 ข้อมูลน้ำหนักของประชากรคุกกี้ 15 ชิ้น

ช็อกโกแลตชิป		เนยถั่ว		มะพร้าว	
ชิ้นที่	น้ำหนัก (กรัม)	ชิ้นที่	น้ำหนัก (กรัม)	ชิ้นที่	น้ำหนัก (กรัม)
1	2.5	7	5	12	5.5
2	3	8	2.5	13	4
3	2	9	4	14	6
4	3.5	10	4.5	15	3.5
5	4	11	3		
6	2.5				
น้ำหนักรวม	17.5		19		19
น้ำหนักเฉลี่ย	2.9		3.8		4.75
น้ำหนักรวมทั้งหมด	55.5				
น้ำหนักเฉลี่ยทั้งหมด	3.7				

2.3 ค่าทางสถิติที่เกี่ยวข้องกับการสุ่มข้อมูล

ค่ามัชฌิมเลขคณิต หรือ ค่าเฉลี่ย (mean)

ในตารางที่ 2.1 แสดงจำนวนของประชากรคุกกี้ 15 ชิ้น พร้อมทั้งน้ำหนักของคุกกี้แต่ละชิ้น จำแนกตามชนิด น้ำหนักของคุกกี้รวมทุกชนิด น้ำหนักเฉลี่ยของคุกกี้แต่ละชนิด และน้ำหนักเฉลี่ยของคุกกี้ทั้งหมด โดยน้ำหนักเฉลี่ยของคุกกี้ช็อกโกแลตชิปเป็น 2.9 กรัม คุกกี้เนยถั่วมีน้ำหนักเฉลี่ย 3.8 กรัม คุกกี้มะพร้าวมีน้ำหนักเฉลี่ย 4.75 กรัม และน้ำหนักของคุกกี้โดยเฉลี่ยทั้งหมดคือ 3.7 กรัม (55.5/15 = 3.7 กรัมต่อชิ้น)

จากข้อมูลในตารางพบว่า (1) คุกกี้มะพร้าวมีน้ำหนักเฉลี่ยสูงที่สุด นั่นคือ 4.75 กรัม และคุกกี้ช็อกโกแลตชิปมีน้ำหนักเฉลี่ยต่ำสุด คือ 2.9 กรัม, (2) คุกกี้ที่หนักที่สุดคือคุกกี้ชิ้นที่ 14 หนัก 6 กรัม และชิ้นที่เบาที่สุดคือชิ้นที่ 3 หนัก 2 กรัม, และ (3) คุกกี้เนยถั่วเป็นคุกกี้ที่มีน้ำหนักเฉลี่ยใกล้เคียงกับน้ำหนักเฉลี่ยของคุกกี้ทั้งหมด

ความแปรปรวน (variance)

ความแปรปรวนเป็นค่าที่ใช้วัดความแตกต่างของน้ำหนักคุกกี้แต่ละชิ้นกับน้ำหนักเฉลี่ยของคุกกี้ทั้งหมด ยกกำลังสอง แล้วหารด้วยจำนวนของคุกกี้ ดังนั้นในตารางที่ 2.1 เราจะคำนวณความแปรปรวนได้ ดังนี้

$$\text{Variance} = \frac{(2.5-3.7)^2+(3-3.7)^2+\ldots+(6-3.7)^2+(3.5-3.7)^2}{15}$$

$$= 1.29$$

การสุ่มคุกกี้

สมมติว่าคุกกี้ทั้งหมด 15 ชิ้นเป็นประชากรคุกกี้ แล้วให้เลือกมา 5 ชิ้น โดยปราศจากอคติส่วนบุคคล จากนั้นหาน้ำหนักเฉลี่ยของคุกกี้ทั้ง 5 ชิ้น การเลือกคุกกี้อาจจะกระทำโดยการเขียนหมายเลข 1 ถึง 15 ลงบนกระดาษ แล้วสุ่มจับกระดาษขึ้นมา 5 ใบ แล้วหยิบคุกกี้ตามลำดับหมายเลขที่สุ่มจับได้ ข้อมูลในตารางที่ 2.2 เป็นน้ำหนักของคุกกี้ที่เลือกได้

ตารางที่ 2.2 น้ำหนักของตัวอย่างคุกกี้ 5 ชิ้น

ชิ้นที่	น้ำหนัก (กรัม)
2	3
6	2.5
7	5
11	3
14	6
น้ำหนักรวม	19.5
น้ำหนักเฉลี่ย	3.9

จะเห็นว่าน้ำหนักของคุกกี้ทั้ง 5 ชิ้น (ชิ้นที่ 2, 6, 7, 11 และ 14) มีน้ำหนักโดยเฉลี่ยเป็น 3.9 กรัม น้ำหนักนี้สูงกว่าน้ำหนักเฉลี่ยรวมของคุกกี้ 15 ชิ้นที่มีค่าเฉลี่ยเป็น 3.7 กรัม โดยธรรมชาติแล้วถ้าเลือกคุกกี้ชุดใหม่ขึ้นมาอีก 5 ชิ้น น้ำหนักเฉลี่ยของคุกกี้ชุดใหม่ย่อมแตกต่างจากชุดเดิม

ความคลาดเคลื่อนในการสุ่ม

เรามีคุกกี้ 15 ชิ้นที่เป็นประชากรของคุกกี้ แล้วเลือกคุกกี้มา 5 ชิ้นเป็นกลุ่มตัวอย่างโดยดึงมาจากประชากรคุกกี้ 15 ชิ้น สมมติว่าเราเลือกคุกกี้ 5 ชิ้นจากประชากรทั้งหมด 15 ชิ้น ทำการเลือกเช่นนี้หลายๆ ครั้ง เราก็จะได้คุกกี้ที่แตกต่างกัน (เช่น เลือกคุกกี้ชิ้นที่ 1, 2, 3, 4, 5 หรือชิ้นที่ 1,

2, 3, 4, 6 หรือชิ้นที่ 1 3, 5, 7, 9 หรือชิ้นที่ 11, 12, 13, 14, 15 ฯลฯ) กลุ่มตัวอย่างคุกกี้ที่เลือกได้ในแต่ละครั้งจะมีน้ำหนักเฉลี่ยที่แตกต่างกัน ทั้งอาจสูงกว่าและต่ำกว่าน้ำหนักเฉลี่ยของประชากรคุกกี้

ตารางที่ 2.3 แสดงกลุ่มตัวอย่างคุกกี้ 2 กลุ่ม คือกลุ่ม L และกลุ่ม H น้ำหนักเฉลี่ยของคุกกี้กลุ่ม L คือ 2.5 กรัม ขณะที่น้ำหนักเฉลี่ยของคุกกี้กลุ่ม H คือ 5 กรัม จากตารางที่ 2.1 น้ำหนักเฉลี่ยของประชากรคุกกี้ทั้ง 15 ชิ้นคือ 3.7 กรัม แน่นอนว่าถ้าเราเลือกกลุ่มตัวอย่างได้เป็นกลุ่ม L แล้วน้ำหนักเฉลี่ยของกลุ่มตัวอย่างที่ได้จะต่ำกว่าค่าเฉลี่ยของประชากรคุกกี้ (3.7 - 2.5 = 1.2 กรัม) และถ้าเราเลือกกลุ่มตัวอย่างได้เป็นกลุ่ม H แล้วน้ำหนักเฉลี่ยของกลุ่มตัวอย่างที่ได้จะสูงกว่าค่าเฉลี่ยของประชากรคุกกี้ (3.7 - 5.0 = -1.3 กรัม) จะเห็นว่ากลุ่มตัวอย่างทั้งสอง มีน้ำหนักเฉลี่ยไม่เท่ากับน้ำหนักของประชากร ซึ่งผลต่างของน้ำหนักที่ได้จะเรียกว่า "ความคลาดเคลื่อนในการสุ่ม"

ตารางที่ 2.3 น้ำหนักคุกกี้กลุ่ม L และกลุ่ม H

กลุ่ม L		กลุ่ม H	
ชิ้นที่	น้ำหนัก (กรัม)	ชิ้นที่	น้ำหนัก (กรัม)
1	2.5	5	4
2	3	7	5
3	2	10	4.5
6	2.5	12	5.5
8	2.5	14	6
น้ำหนักรวม	12.5		25
น้ำหนักเฉลี่ย	2.5		5

ในตารางที่ 2.2 จะเห็นว่ากลุ่มตัวอย่างที่เราสุ่มมาในครั้งแรกจะมีค่าเฉลี่ยเข้าใกล้กับค่าเฉลี่ยของประชากร (3.9 และ 3.7 ตามลำดับ) ถ้าเราสุ่มคุกกี้ 5 ชิ้นมาหลาย ๆ ครั้ง จะมีโอกาสมากขึ้นที่จะได้น้ำหนักของกลุ่มตัวอย่างใกล้เคียงกับน้ำหนักของประชากร ส่วนน้ำหนักที่มีค่าห่างไกลจากน้ำหนักของประชากรเช่นกลุ่ม L และกลุ่ม H ในตารางที่ 2.3 จะมีโอกาสน้อยกว่า

2.4 วิธีการสุ่มข้อมูล

การสุ่มข้อมูลตัวอย่างมีขั้นตอนที่ใช้ในการออกแบบเพื่อให้ได้ตัวอย่างที่ดี ดังนี้

- เลือกคุณลักษณะของข้อมูลที่จะเป็นตัวอย่าง
- กำหนดกลุ่มประชากรที่จะสุ่มตัวอย่าง
- เลือกวิธีการสุ่มตัวอย่าง
- กำหนดขนาดของตัวอย่าง

การเลือกคุณลักษณะของข้อมูลที่จะเป็นตัวอย่าง

จะต้องมีการกำหนดตัวแปร คุณสมบัติ และความสัมพันธ์ของข้อมูลที่ต้องการ รวมทั้งวัตถุประสงค์ในการเลือกข้อมูล รวมถึงวิธีที่ใช้ และต้องมีแผนการที่แน่ชัดในการกระทำกับข้อมูลที่ได้จากการสุ่ม เพื่อไม่ให้เสียเวลากับข้อมูลที่ไม่ถูกต้องตามวัตถุประสงค์

การกำหนดกลุ่มประชากรที่จะเป็นตัวอย่าง

ต้องมีการกำหนดกลุ่มประชากร โดยเฉพาะถ้าเป็นข้อมูลเชิงคุณภาพ ต้องสุ่มในขนาดที่เพียงพอ

การเลือกวิธีการสุ่มตัวอย่าง

วิธีการสุ่มสามารถจำแนกได้เป็นสองประเภทหลัก คือ การสุ่มแบบอาศัยความน่าจะเป็น และการสุ่มแบบไม่อาศัยความน่าจะเป็น

❒ ***การสุ่มแบบอาศัยความน่าจะเป็น*** *(probability sampling)*

เป็นการสุ่มตัวอย่างจากประชากร โดยมีเงื่อนไขดังต่อไปนี้

1. รู้จำนวนประชากรทั้งหมด
2. ประชากรทั้งหมดมีโอกาสที่จะถูกสุ่มมาเป็นกลุ่มตัวอย่างเท่าเทียมกัน
3. ใช้วิธีการสุ่มที่เหมาะสม เพื่อให้ทุกตัวอย่างมีโอกาสถูกสุ่มเท่าเทียมกัน
4. ใช้วิธีประมาณค่าพารามิเตอร์ที่เหมาะสม

การสุ่มแบบอาศัยความน่าจะเป็น เป็นวิธีที่นิยมใช้กันมาก เพราะมีความน่าเชื่อถือ การสุ่มแบบนี้มีหลายวิธี คือ

(1) การสุ่มตัวอย่างด้วยวิธีอย่างง่าย *(simple random sampling)*

การสุ่มตัวอย่างโดยวิธีนี้ต้องมีรายการหรือรายชื่อของประชากรทั้งหมด เพื่อให้แน่ใจว่าตัวอย่างมีโอกาสถูกเลือกเท่ากัน ประชากรจะต้องกำหนดเฉพาะลงไปว่าเป็นกลุ่มใด เช่น ประชากรเป็นเด็กนักเรียนระดับมัธยมศึกษาปีที่ 1 ในโรงเรียนสังกัดกรมสามัญศึกษา เขตการศึกษา 5 เป็นต้น

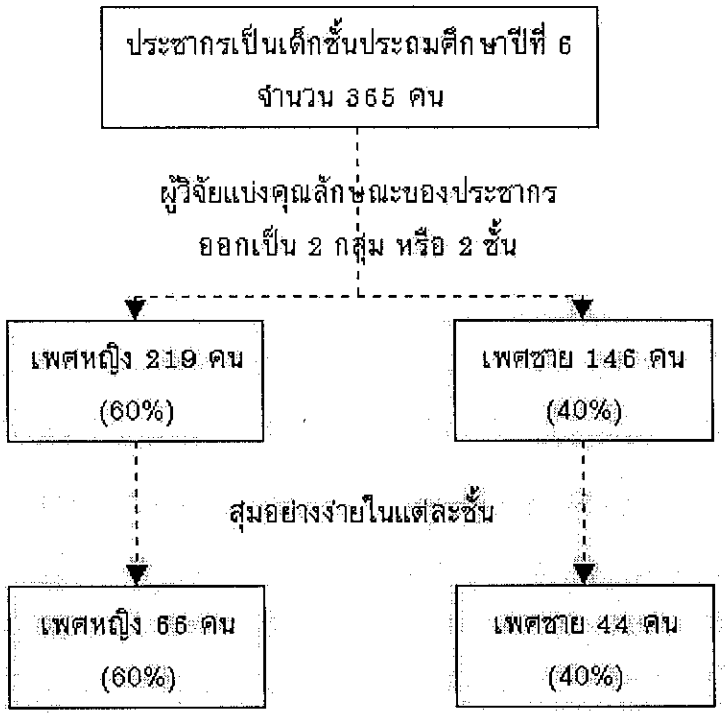
การสุ่มแบบนี้จะต้องกำหนดเลขลำดับให้กับประชากรแต่ละหน่วยเช่น ต้องการกลุ่มตัวอย่าง 100 คนจากประชากร 2,000 คน จะต้องมีรายชื่อของประชากรทั้ง 2,000 คน แล้วให้เลขลำดับแก่ประชากรแต่ละคน ตั้งแต่ 0001 ถึง 2000 จากนั้นอาจจะใช้ตารางเลขสุ่มในการสุ่มตัวอย่างดังแสดงในตารางที่ 2.4

การสุ่มนั้นจะต้องเลือกสดมภ์ใด สดมภ์หนึ่งขึ้นมา แล้วอ่านตัวเลขในแถวแรกจำนวน 4 หลักโดยไม่ต้องพิจารณาหลักที่ 5 (ที่ต้องกำหนดเป็นเพียง 4 หลักเพราะประชากรมี 2,000 คน คนที่ 1 มีเลขลำดับ 4 หลักคือ 0001 จนถึงคนสุดท้ายมีเลขลำดับ 4 หลักคือ 2000) ตัวเลขสุ่มตัวแรกตามตารางที่ 2.4 และเลขสี่หลักแรกคือ 0117 ดังนั้นกลุ่มตัวอย่างคนแรกก็คือคนที่มีเลขลำดับที่ 0117 อ่านแถวต่อไปได้เลข 9123 แต่เลขลำดับที่ 9123 ไม่มี จึงต้องข้ามไปอ่านเลขในแถวถัดไปคือ 0864 ดังนั้นกลุ่มตัวอย่างคนที่สองก็คือคนที่มีเลขลำดับที่ 0864 และกลุ่มตัวอย่างคนที่สามก็คือคนที่มีเลขลำดับที่ 0593 แถวถัดมาได้เลข 6662 ซึ่งก็ต้องข้ามไปอีกเช่นกัน ดังนั้นกลุ่มตัวอย่างคนที่สี่ก็คือคนที่มีเลขลำดับที่ 0519 อ่านไปเรื่อย ๆ จนกระทั่งได้กลุ่มตัวอย่างครบ 100 คน ตามที่ต้องการ

ตารางที่ 2.4 ส่วนหนึ่งของตารางเลขสุ่ม

01172	22345	22216	03276	06228	56545
91233	97915	23398	10923	93412	98767
08640	01626	41114	25128	60234	65908
05939	02233	08067	45455	01156	23787
66627	10659	87980	89903	90987	19890
05196	00457	03690	03770	50009	04666
06304	78632	09800	51037	02435	14567
01172	22345	22216	03276	06228	56545

จุดเด่นของการสุ่มแบบนี้ก็คือมีความสะดวกและใช้ได้ง่าย แต่มีข้อเสียคือ ถ้ากลุ่มตัวอย่างที่ต้องการมีจำนวนมาก การใช้วิธีนี้ก็จะเสียเวลามาก เนื่องจากผู้วิจัยต้องรู้จักประชากรทุกคน คือต้องรู้ว่า ประชากรลำดับที่ 0117 เป็นใคร ยิ่งกว่านั้นผู้วิจัยบางคนจะไม่ใช้การสุ่มอย่างง่าย ถ้าแน่ใจว่ากลุ่มประชากรสามารถจำแนกออกเป็นกลุ่มย่อยที่มีสัดส่วนแน่นอน ผู้วิจัยอาจจะใช้การสุ่มแบบแบ่งชั้นแทน



รูปที่ 2.1 การสุ่มแบบแบ่งชั้น

การสุ่มแบบแบ่งชั้นมีประโยชน์ช่วยให้ผู้วิจัยมีความมั่นใจว่าคุณลักษณะหรือตัวแปรที่สนใจศึกษาที่อยู่ในประชากรนั้น ก็มีอยู่ในกลุ่มตัวอย่างในสัดส่วนที่เท่ากัน

*(4) **การสุ่มแบบแบ่งกลุ่ม** (cluster random sampling)*

เป็นวิธีที่ผู้วิจัยใช้ในการแบ่งประชากรออกเป็นกลุ่มตามเขตพื้นที่ (จึงอาจเรียกการสุ่มแบบนี้ได้ว่า การสุ่มตามพื้นที่ หรือ area sampling) ซึ่งในแต่ละเขตพื้นที่จะมีประชากรที่มีคุณลักษณะที่ต้องการกระจายกันอยู่อย่างเท่าเทียมกัน แล้วสุ่มกลุ่มมาจำนวนหนึ่งด้วยวิธีการสุ่มที่เหมาะสม เช่น บริษัทผลิตอาหารกึ่งสำเร็จรูปต้องการสำรวจความต้องการบริโภคอาหารกึ่งสำเร็จรูปของประชากรในกรุงเทพมหานคร ถ้าจะใช้วิธีการสุ่มอย่างง่าย บริษัทจะต้องมีรายชื่อของประชากรในกรุงเทพมหานครทั้งหมด ซึ่งเป็นเรื่องที่ยุ่งยากมาก หรือถ้าจะใช้การสุ่มแบบแบ่งชั้นโดยจำแนกประชากรตามระดับรายได้ แน่นอนว่ากลุ่มตัวอย่างที่ได้จะกระจัดกระจายไปทั่วกรุงเทพมหานคร ทำให้ต้องเสียค่าใช้จ่ายสูงในการเก็บข้อมูลภาคสนาม ดังนั้นการสุ่มแบบแบ่งกลุ่มจึงเป็นวิธีที่เหมาะสมที่สุด โดยการใช้วิธีการแบ่งกรุงเทพมหานครออกเป็นเขตจำนวน n เขต แล้วสุ่มเขต m เขตมาเป็นกลุ่มตัวอย่าง (m < n) เรียกว่า การสุ่มแบบแบ่งกลุ่มขั้นเดียว (single-stage cluster

sampling) เมื่อได้เขตพื้นที่ที่เป็นกลุ่มตัวอย่างมาแล้ว จำนวนกลุ่มตัวอย่างอาจจะยังมีเป็นจำนวนมากอยู่ บริษัทที่วิจัยอาจจะสุ่มอีกครั้ง จากเขตที่สุ่มได้มา m เขตประกอบไปด้วยตำบล q ตำบล แล้วสุ่มตำบลมา p ตำบลเป็นกลุ่มตัวอย่าง (p < q) เรียกว่า การสุ่มแบบแบ่งกลุ่มสองขั้นตอน (two-stage cluster sampling) ถ้าหากจำนวนกลุ่มตัวอย่างที่ได้ยังมีจำนวนมากอยู่ อาจสุ่มโดยใช้หมู่บ้านหรือชุมชนเป็นหน่วยในการสุ่ม เรียกว่า การสุ่มแบบแบ่งกลุ่มหลายขั้นตอน (multi-stage cluster sampling)

(5) การสุ่มแบบหลายขั้นตอน (multi-stage sampling)

มีวิธีการสุ่ม 4 แบบที่อธิบายไว้แล้ว คือ การสุ่มอย่างง่าย การสุ่มอย่างมีระบบ การสุ่มแบบแบ่งชั้น และการสุ่มแบบแบ่งกลุ่ม ในการทำวิจัยจริง ๆ เราอาจจะใช้วิธีการสุ่มที่ซับซ้อนมากขึ้น โดยอาจจะพิจารณาใช้วิธีการสุ่มหลายแบบร่วมกันเพื่อให้ได้ประโยชน์สูงสุดและเพื่อให้ได้กลุ่มตัวอย่างที่ผู้วิจัยต้องการอย่างแท้จริง เรียกว่าการสุ่มแบบหลายขั้นตอน

ตัวอย่างเช่น ประชากรคือนักเรียนชั้นมัธยมศึกษาตอนปลาย ในเขตการศึกษา 10 ใช้การสุ่มแบบแบ่งกลุ่ม แบ่งออกเป็น 7 จังหวัด สุ่มได้มา 3 จังหวัด ในทั้ง 3 จังหวัดมี 45 โรงเรียน ใช้การสุ่มแบบแบ่งชั้นโดยแบ่งตามขนาดของโรงเรียน คือโรงเรียนขนาดใหญ่ 10 โรงเรียน ขนาดกลาง 12 โรงเรียน และขนาดเล็ก 23 โรงเรียน สุ่มได้โรงเรียนขนาดใหญ่ 2 โรงเรียน ขนาดกลาง 3 โรงเรียน และขนาดเล็ก 5 โรงเรียน ใน 10 โรงเรียนมีนักเรียนทั้งหมด 10,000 คน ใช้การสุ่มอย่างง่ายมา 20% จากจำนวนนักเรียนทั้งหมด ได้นักเรียนมาเป็นกลุ่มตัวอย่าง 2,000 คน

❒ ***การสุ่มแบบไม่อาศัยความน่าจะเป็น (non-probability sampling)***

เป็นการสุ่มตัวอย่างที่บางครั้งอาจไม่ทราบจำนวนประชากรที่แท้จริง ทำให้ไม่สามารถใช้การสุ่มแบบอาศัยความน่าจะเป็นได้ และการสุ่มแต่ละครั้งนั้นทุกๆ หน่วยของประชากรมีโอกาสถูกสุ่มมาเป็นกลุ่มตัวอย่างไม่เท่าเทียมกัน การสุ่มแบบนี้มีหลายวิธี คือ

(1) การเลือกตัวอย่างตามความสะดวก (convenience หรือ accidental sampling)

เป็นการเลือกแบบไม่มีกฎเกณฑ์ อาศัยความสะดวกของผู้วิจัยเป็นหลัก กลุ่มตัวอย่างจะเป็นใครก็ได้ที่ให้ความร่วมมือกับผู้วิจัยในการให้ข้อมูลบางอย่าง เช่น

- สอบถามความคิดเห็นในการให้บริการอาหารกลางวันของมหาวิทยาลัยแห่งหนึ่ง ผู้วิจัยอาจจะไปยืนตรงประตูทางเข้าโรงอาหาร แล้วคอยสัมภาษณ์นักศึกษา 50 คนที่เดินเข้ามารับประทานอาหารในช่วงเช้าของวันหนึ่ง

- ครูคนหนึ่งปรับปรุงแผนการสอนใหม่ และต้องการจะทดลองแผนการสอนใหม่ว่าจะให้ผลแตกต่างจากแผนการสอนเดิมหรือไม่ โดยใช้กลุ่มตัวอย่าง เป็นนักเรียนที่เขาต้องรับผิดชอบสอน

(2) การเลือกตัวอย่างแบบเจาะจง (purposive หรือ judgmental sampling)

เป็นการเลือกกลุ่มตัวอย่างที่จะเป็นใครก็ได้ที่มีลักษณะตรงตามความต้องการของผู้วิจัย โดยอาจจะกำหนดเป็นคุณลักษณะเฉพาะเจาะจงลงไป เช่น

- เป็นเพศหญิงที่ทำงานในธนาคารอายุระหว่าง 30 ถึง 40 ปี
- เป็นนักเรียนระดับมัธยมศึกษาปีที่ 3 ที่เรียนอยู่ในโรงเรียนสังกัดกรมสามัญศึกษา ที่ได้เกรดเฉลี่ย 3.50 ขึ้นไป และมีความสามารถพิเศษทางดนตรี

(3) การเลือกตัวอย่างแบบโควต้า (quota sampling)

เป็นการเลือกตัวอย่างโดยกำหนดคุณลักษณะและสัดส่วนที่ต้องการไว้ล่วงหน้า โดยที่คุณลักษณะอาจเป็น เพศ อายุ เชื้อชาติ ระดับการศึกษา หรืออื่นๆ เช่น

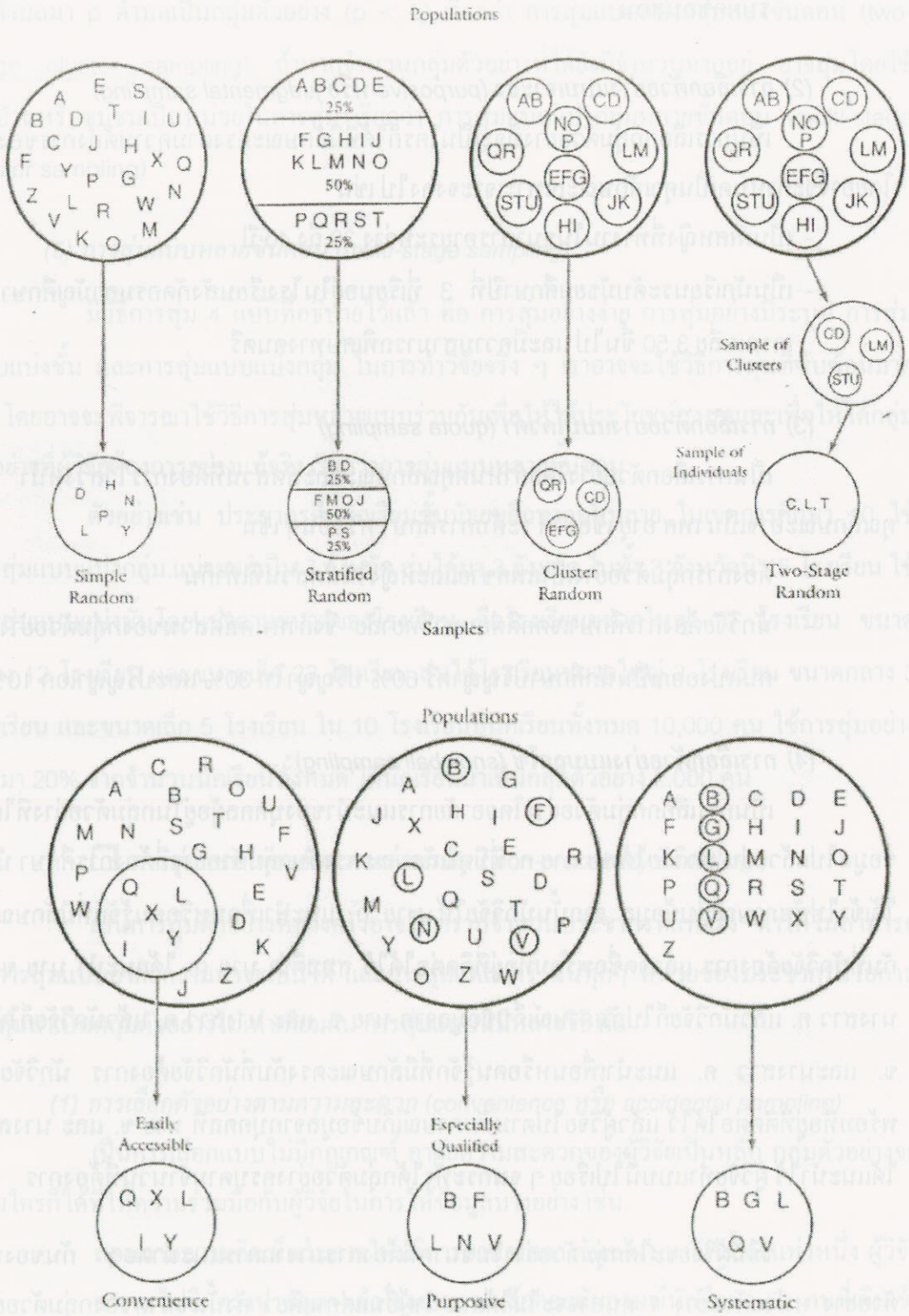
- ต้องการกลุ่มตัวอย่างเป็นเพศชายและหญิงในสัดส่วนที่เท่ากัน
- นักวิจัยต้องการศึกษาเจตคติต่อมหาวิทยาลัย จึงกำหนดสัดส่วนของกลุ่มตัวอย่าง 400 คนแบ่งออกเป็นนักศึกษาปริญญาตรี 60% ปริญญาโท 30% และปริญญาเอก 10%

(4) การเลือกตัวอย่างแบบลูกโซ่ (snowball sampling)

เป็นการเลือกกลุ่มตัวอย่างโดยอาศัยการแนะนำของบุคคลผู้อยู่ในกลุ่มตัวอย่างที่ได้เก็บข้อมูลไปแล้ว เช่น นักวิจัยได้พบ นาย ก. ที่มีคุณลักษณะตรงกับกลุ่มตัวอย่างที่ต้องการศึกษา นักวิจัยได้เข้าไปสัมภาษณ์เก็บข้อมูล จากนั้นนักวิจัยให้ นาย ก. แนะนำเพื่อนหรือคนรู้จักที่มีลักษณะตรงกับที่นักวิจัยต้องการ แล้วจดชื่อพร้อมที่อยู่ที่ติดต่อได้ไว้ สมมติว่า นาย ก. ได้แนะนำ นาย ข. และนางสาว ค. แล้วนักวิจัยก็ไปสัมภาษณ์เก็บข้อมูลจาก นาย ข. และ นางสาว ค. แล้วนักวิจัยก็ให้ นาย ข. และนางสาว ค. แนะนำเพื่อนหรือคนรู้จักที่มีลักษณะตรงกับที่นักวิจัยต้องการ นักวิจัยจดชื่อพร้อมที่อยู่ที่ติดต่อได้ไว้ แล้วผู้วิจัยไปตามสัมภาษณ์เก็บข้อมูลจากบุคคลที่ นาย ข. และ นางสาว ค. ได้แนะนำไว้ ผู้วิจัยทำแบบนี้ไปเรื่อย ๆ จนกระทั่งได้กลุ่มตัวอย่างครบตามจำนวนที่ต้องการ

วิธีนี้ผู้วิจัยจะได้กลุ่มตัวอย่างในขนาดที่ต้องการมาจากการแนะนำต่อๆ กันของหน่วยตัวอย่าง หน่วยตัวอย่าง 1 คนอาจจะไม่ได้แนะนำผู้อื่นแค่คนเดียว ดังนั้นขนาดของกลุ่มตัวอย่างจะเพิ่มขึ้นทุกครั้งที่ได้ไปสัมภาษณ์เก็บข้อมูล เหมือนกับก้อนหิมะที่ยิ่งกลิ้งไปลูกหิมะก็จะยิ่งใหญ่ขึ้น ดังนั้นวิธีนี้จึงได้ชื่อว่า snowball sampling

วิธีการสุ่มแบบต่างๆ ที่กล่าวมาข้างต้น ทั้งที่อาศัยความน่าจะเป็นและไม่อาศัยความน่าจะเป็น สามารถแสดงสรุปเป็นแผนภาพได้ดังรูปที่ 2.2



รูปที่ 2.2 วิธีการสุ่มแบบอาศัยความน่าจะเป็นและไม่อาศัยความน่าจะเป็น

การกำหนดขนาดตัวอย่าง

ขนาดของกลุ่มตัวอย่างจะเล็กหรือใหญ่ขึ้นอยู่กับปัจจัยหลายประการ เช่น ค่าใช้จ่าย ระยะเวลาที่ใช้ในการสุ่มตัวอย่าง หรือ คุณลักษณะของข้อมูล ปัจจัยเหล่านี้จะมีผลต่อการกำหนดขนาดของกลุ่มตัวอย่าง ซึ่งมีขั้นตอนการคำนวณโดยละเอียด ดังต่อไปนี้

- ***การกำหนดขนาดตัวอย่างเมื่อสุ่มข้อมูลตามคุณลักษณะ*** *(sampling data on attributes)*

การสุ่มตัวอย่างตามคุณลักษณะของข้อมูล เช่น ข้อมูลเป็นสัดส่วนของคนในองค์กรที่มีแนวความคิดเดียวกัน หรือ เปอร์เซ็นต์ของแบบฟอร์มในการป้อนข้อมูลที่มีความผิดพลาด โดยแบ่งขั้นตอนในการกำหนดขนาดตัวอย่าง 7 ขั้นตอนดังนี้

(1) กำหนดคุณลักษณะของข้อมูลที่จะทำการสุ่ม

(2) สืบค้นข้อมูลจากฐานข้อมูลหรือรายงาน

(3) ตรวจสอบคุณลักษณะ ซึ่งจะได้สัดส่วนของคุณลักษณะจากฐานข้อมูลทั้งหมด เป็นค่าตัวแปร *p* (proportion)

(4) กำหนดช่วงการยอมรับความผิดพลาด เป็นตัวแปร *i* (acceptable interval)

(5) เลือกระดับความน่าเชื่อถือ (confidence level) เช่น 99% หรือ 95% แล้วหาค่าสัมประสิทธิ์ความน่าเชื่อถือ จากตาราง z (ตารางที่ 2.5)

(6) คำนวณหาค่าเบี่ยงเบนมาตรฐาน จากสูตร $\sigma_p = i / z$

(7) สามารถกำหนดขนาดของตัวอย่างที่จะสุ่มได้จากสูตร $N = (p(1-p) / \sigma_p^2) + 1$

ตารางที่ 2.5 ค่าสัมประสิทธิ์ความน่าเชื่อถือ (z-value)

Confidence Level	Confidence Coefficient (z-value)
99%	2.58
98%	2.33
97%	2.17
96%	2.05
95%	1.96
90%	1.65
80%	1.28
50%	0.67

ตัวอย่าง บริษัทผลิตชั้นวางของทำด้วยโลหะแห่งหนึ่ง ต้องการสุ่มตัวอย่างเพื่อหาสัดส่วนของใบสั่งสินค้าที่บันทึกข้อมูลผิดพลาด จึงได้ทำขั้นตอนดังนี้

1. หาใบสั่งสินค้าที่มีการบันทึกข้อมูลผิด โดยมีข้อมูลลูกค้า ปริมาณของที่สั่ง และหมายเลขสินค้าที่บันทึกผิดพลาด
2. โดยใช้ใบสั่งซื้อเมื่อ 6 เดือนที่ผ่านมา
3. ทำการตรวจสอบจากใบสั่งซื้อดังกล่าว พบสัดส่วนข้อมูลที่ผิดพลาดประมาณ 5% ดังนั้น p = 0.05
4. กำหนดช่วงการยอมรับให้เกิดความผิดพลาดได้ที่ค่า 0.02 ซึ่งกำหนดให้เป็นตัวแปร i
5. เลือกระดับความน่าเชื่อถือ 95% ซึ่งหาสัมประสิทธิ์ความน่าเชื่อถือได้ค่า z = 1.96
6. คำนวณหาค่าเบี่ยงเบนมาตรฐานจากสูตร $\sigma_p = i/z$

$$\text{ดังนั้น } \sigma_p = 0.02/1.96$$

$$= 0.0102$$

7. สามารถหาขนาดตัวอย่างที่จะสุ่มได้จากสูตร $N = (p(1-p)/ \sigma_p^2) + 1$

$$N = (0.05 \times 0.95) / (0.0102 \times 0.0102) + 1$$

$$= 458 \text{ รายการ}$$

จากตัวอย่างได้จำนวนตัวอย่างที่ต้องทำการสุ่ม 458 รายการ ในการสุ่มตัวอย่างยิ่งกำหนดระดับความความน่าเชื่อถือมากขึ้นเท่าไร หรือมีช่วงการยอมรับความผิดพลาดแคบลงเท่าไร ยิ่งต้องการขนาดของตัวอย่างที่จะสุ่มมากขึ้นเท่านั้น เช่น ถ้าเพิ่มระดับความน่าเชื่อถือเป็น 99% จะได้ค่าสัมประสิทธิ์ความน่าเชื่อถือ z = 2.58 เมื่อแทนค่าในสูตรจะได้ขนาดตัวอย่าง 782 รายการแต่ถ้าเรายังคงระดับความน่าเชื่อถือเป็น 95% แต่ลดช่วงการยอมรับเป็น 0.01 เราจะต้องใช้จำนวนตัวอย่างมากขึ้นเป็น 1,827 รายการ

- ***การกำหนดขนาดตัวอย่างเมื่อสุ่มข้อมูลตามตัวแปร*** *(sampling on variables)*

ในบางครั้งนักวิเคราะห์อาจต้องการที่จะเก็บข้อมูลตามค่าใดค่าหนึ่งที่แน่นอน เช่น ยอดขายสินค้า จำนวนรายการสินค้าที่ถูกส่งกลับ หรือ จำนวนของความผิดพลาดที่เกิดจากการป้อนข้อมูล ข้อมูลประเภทนี้เปรียบเสมือนเป็นตัวแปร ขั้นตอนในการกำหนดขนาดตัวอย่างคล้ายกับที่ผ่านมา คือ

(1) กำหนดคุณลักษณะของข้อมูลที่จะทำการสุ่ม

(2) สืบค้นข้อมูลจากฐานข้อมูลหรือรายงาน

(3) ตรวจสอบข้อมูลเพื่อหาค่าเฉลี่ยเพื่อกำหนดช่วงการยอมรับและส่วนเบี่ยงเบนมาตรฐาน ให้เป็นค่าตัวแปร S

(4) กำหนดช่วงการยอมรับความผิดพลาด เป็นตัวแปร i (acceptable interval)

(5) เลือกระดับความน่าเชื่อถือ (confidence level) เช่น 99% หรือ 95% แล้วหาค่าสัมประสิทธิ์ความน่าเชื่อถือจากตาราง z

(6) คำนวณหาค่าเบี่ยงเบนมาตรฐาน จากสูตร $\sigma_x = i/z$

(7) สามารถกำหนดขนาดของตัวอย่างที่จะสุ่มได้จากสูตร $N = (S / \sigma_p)^2 + 1$

ตัวอย่าง บริษัทผลิตชั้นวางของทำด้วยโลหะแห่งหนึ่ง ต้องการสุ่มตัวอย่างเพื่อหาค่าเฉลี่ยของจำนวนเงินที่สั่งซื้อสินค้า จึงได้ทำขั้นตอนดังนี้

1. หาใบสั่งสินค้าที่มีการบันทึกข้อมูลลูกค้า ปริมาณของที่สั่ง จำนวนเงินที่สั่ง
2. โดยใช้ใบสั่งซื้อเมื่อ 6 เดือนที่ผ่านมา
3. ทำการตรวจสอบจากใบสั่งซื้อดังกล่าว พบจำนวนเงินที่สั่งซื้อโดยเฉลี่ย 1,500 บาท ด้วยส่วนเบี่ยงเบนมาตรฐานประมาณ 100 บาท
4. กำหนดช่วงการยอมรับให้เกิดความผิดพลาดได้เป็น 5.00 ให้เป็นตัวแปร i
5. เลือกระดับความน่าเชื่อถือ 96% ซึ่งหาสัมประสิทธิ์ความน่าเชื่อถือได้ค่า z = 2.05
6. คำนวณหาค่าเบี่ยงเบนมาตรฐานจากสูตร ได้ค่า $\sigma_x = i/z$

แทนค่าในสูตร จะได้ค่า $\sigma_x = 5.00/2.05$

$= 2.44$

7. สามารถหาขนาดตัวอย่างที่จะสุ่มได้จากสูตร $N = (S / \sigma_p)^2 + 1$

$N = (100 / 2.44)^2 + 1$

$= 1,681$ รายการ

จากตัวอย่างได้จำนวนตัวอย่างที่ต้องการสุ่ม 1,681 รายการ การสุ่มตัวอย่างยิ่งกำหนดระดับความน่าเชื่อถือมากขึ้นเท่าไร หรือมีช่วงการยอมรับความผิดพลาดแคบลงเท่าไร ยิ่งต้องการขนาดของตัวอย่างที่จะสุ่มมากขึ้นเท่านั้น เช่น ถ้าเพิ่มระดับความน่าเชื่อถือเป็น 99% จะได้ค่าสัมประสิทธิ์ความน่าเชื่อถือ z = 2.58 เมื่อแทนค่าในสูตรจะได้ขนาดตัวอย่าง 2,658 รายการ แต่ถ้าเรายังคงระดับความน่าเชื่อถือเป็น 96% แต่ลดช่วงการยอมรับเป็น 1.00 เราจะต้องใช้จำนวนตัวอย่างมากขึ้นเป็น 41,992 รายการ

บทที่ 3
วิธีดำเนินการวิจัย

โครงการวิจัยนี้มีจุดมุ่งหมายที่จะศึกษาผลของการลดขนาดของข้อมูลด้วยการสุ่มแบบพื้นฐานที่อาศัยความน่าจะเป็น ได้แก่ การสุ่มอย่างง่าย (simple random sampling), การสุ่มแบบเป็นระบบ (systematic random sampling), การสุ่มแบบแบ่งชั้น (stratified random sampling) นอกจากนี้ยังได้ศึกษาเปรียบเทียบวิธีการสุ่มแบบก้าวหน้าอีกสองรูปแบบ คือ การสุ่มแบบก้าวหน้าเชิงเลขคณิต (arithmetic progressive sampling) และ การสุ่มแบบก้าวหน้าเชิงเรขาคณิต (geometric progressive sampling) การสุ่มข้อมูลทั้งห้าเทคนิคนี้ใช้ในการเตรียมข้อมูลเพื่อทำเหมืองข้อมูลประเภทสังเคราะห์โมเดลจำแนกข้อมูล (classification model) การศึกษาวิจัยนี้มีวัตถุประสงค์เพื่อค้นหารูปแบบการสุ่มข้อมูลที่เหมาะสมที่สุด รายละเอียดเนื้อหาในบทนี้ประกอบด้วยคำอธิบายระเบียบวิธีวิจัย ลักษณะของข้อมูลที่ใช้และวิธีการสุ่มข้อมูล และสุดท้ายเป็นการอธิบายวิธีการทดสอบเพื่อเปรียบเทียบรูปแบบการสุ่มข้อมูล

3.1 ระเบียบวิธีวิจัย

การศึกษาวิจัยประกอบด้วย 7 ขั้นตอน ดังนี้

(1) ศึกษาและรวบรวมสรุปงานวิจัยที่เกี่ยวข้อง

รวบรวมเอกสารเกี่ยวกับเทคนิคการสุ่มข้อมูล (sampling technique) และอัลกอริทึมในการสังเคราะห์ความรู้จากข้อมูล (learning algorithm) เพื่อใช้ประกอบการเตรียมข้อมูลและคัดเลือกอัลกอริทึมที่จะใช้ในการวิจัย โดยจำกัดขอบเขตเฉพาะอัลกอริทึมจำแนกประเภทข้อมูล (classification algorithm)

(2) ติดตั้งและศึกษาระบบ WEKA

ในการศึกษาวิจัยนี้เลือกใช้ระบบ WEKA (Waikato Environment for Knowledge Analysis) ที่เป็นซอฟต์แวร์เปิดเผยซอร์สโค้ด พัฒนาและเผยแพร่โดยทีมนักวิจัยมหาวิทยาลัย Waikato ประเทศนิวซีแลนด์ (http://www.cs.waikato.ac.nz/~ml/weka) ซอฟต์แวร์นี้ประกอบด้วยชุดโปรแกรมที่ใช้ในงาน machine learning และการทำเหมืองข้อมูล

(3) รวบรวมข้อมูล

ข้อมูลที่ใช้ในการศึกษาวิจัยนี้จะต้องมีขนาดใหญ่เพียงพอที่จะเอื้อให้สามารถลดจำนวนข้อมูลลงเป็นสัดส่วนต่างๆได้ โดยข้อมูลจะต้องสามารถถูกอ่านขึ้นมาประมวลผลในระบบคอมพิวเตอร์ได้ ข้อมูลที่ใช้ในการทดลองนี้คัดเลือกมาจากชุดข้อมูลมาตรฐาน UCI repository (http://www.ics.uci.edu/~mlearn/MLRepository) คัดเลือกมาใช้ในการทดลองจำนวนรวม 5 ชุดข้อมูล

(4) แปลงรูปแบบข้อมูล

ข้อมูลที่คัดเลือกไว้จะถูกแปลงเป็นรูปแบบ arff (attribute-relation file format) ซึ่งเป็นรูปแบบการใช้งานของระบบ WEKA ข้อมูลแต่ละชุดจะถูกแยกออกเป็นสองส่วนเพื่อทำหน้าที่เป็นข้อมูลฝึก (training data) และข้อมูลทดสอบ (test data) ข้อมูลฝึกจะมีจำนวนประมาณสองในสาม และข้อมูลทดสอบจะมีจำนวนประมาณหนึ่งในสาม

(5) สุ่มข้อมูล

ข้อมูลฝึกแต่ละชุดจะถูกสุ่มออกเป็นสัดส่วนต่างๆกัน เช่น 1%, 2%,3%,... ด้วยวิธีการสุ่ม 5 รูปแบบคือ การสุ่มอย่างง่าย (simple random sampling), การสุ่มแบบเป็นระบบ (systematic random sampling), การสุ่มแบบแบ่งชั้น (stratified random sampling), การสุ่มแบบก้าวหน้าเชิงเลขคณิต (arithmetic progressive sampling) และ การสุ่มแบบก้าวหน้าเชิงเรขาคณิต (geometric progressive sampling)

(6) ทดสอบการสังเคราะห์ความรู้จากข้อมูล

ใช้ข้อมูลฝึกที่ถูกสุ่มเป็นสัดส่วนต่างๆกัน ทดสอบด้วยโปรแกรมสังเคราะห์ความรู้ (learning algorithm) โดยในการทดลองนี้เลือกใช้โปรแกรมเบย์อย่างง่าย (naive Bayes) ที่มีพื้นฐานการทำงานเป็นทฤษฎีของเบย์ซึ่งใช้หลักการของความน่าจะเป็นแบบมีเงื่อนไข และโปรแกรมสังเคราะห์ความรู้จากต้นไม้ตัดสินใจ (decision tree induction) ที่มีพื้นฐานการทำงานจากทฤษฎีสารสนเทศและโครงสร้างข้อมูลแบบต้นไม้ โมเดลหรือความรู้เพื่อการจำแนกข้อมูลที่ได้เป็นผลลัพธ์จากการรันโปรแกรมทั้งสองจะถูกทดสอบความแม่นตรงของโมเดลด้วยข้อมูลทดสอบที่ได้สร้างเตรียมไว้ นอกจากความแม่นตรงแล้วเวลาที่ใช้ในการประมวลผลเพื่อสร้างโมเดลจะถูกบันทึกและนำมาวิเคราะห์เพื่อเปรียบเทียบระหว่างข้อมูลสุ่มแต่ละขนาด

(7) วิเคราะห์ผลการทดสอบและรายงานผล

วิเคราะห์ผลการทดสอบและเสนอแนะรูปแบบการลดข้อมูลด้วยการสุ่มข้อมูลที่เหมาะสมในงานทำเหมืองข้อมูล เพื่อให้ได้ผลการสังเคราะห์ความรู้ที่มีค่าความแม่นตรงอยู่ในระดับที่ยอมรับได้ ขั้นตอนสุดท้ายจะเป็นการเตรียมรายงานฉบับสมบูรณ์

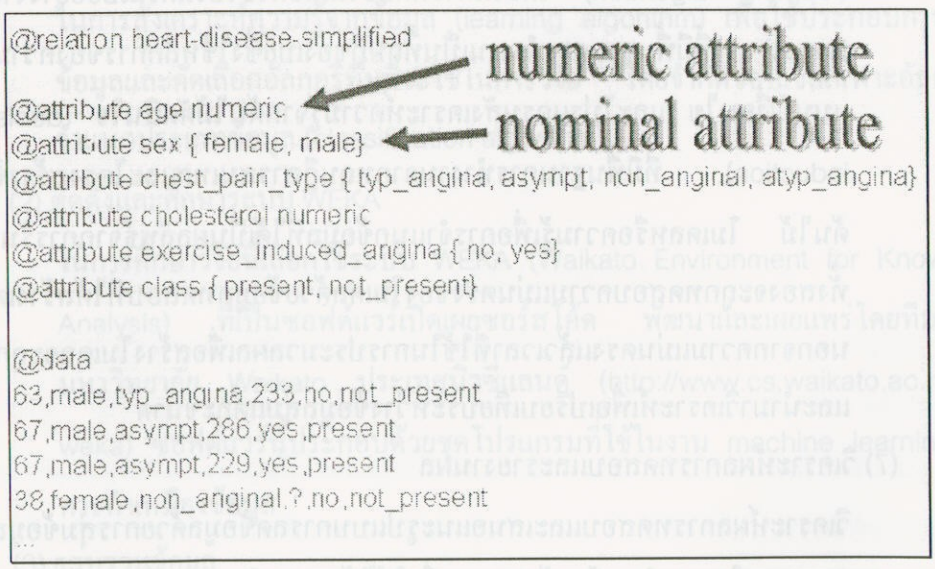
3.2 แหล่งที่มาของข้อมูลและการเตรียมข้อมูล

ข้อมูลที่ใช้ในงานวิจัยนี้เป็นชุดข้อมูลมาตรฐานได้มาจากแหล่งข้อมูลของมหาวิทยาลัยแห่งรัฐแคลิฟอร์เนีย เมืองเออร์ไวน์ (University of California at Irvine, UCI) โดยข้อมูลเหล่านี้เป็นข้อมูลที่นักวิจัยในสาขา machine learning และ data mining นิยมใช้ในการทดสอบประสิทธิภาพของอัลกอริทึมสังเคราะห์ความรู้ ข้อมูลที่คัดเลือกเพื่อนำมาใช้ในโครงการวิจัยนี้ประกอบด้วยข้อมูล Adult, Letter , Mushroom , Shuttle และ Satellite Image รายชื่อชุดข้อมูล จำนวนแอททริบิวต์(ฟิลด์) และคลาส สรุปได้ดังตารางที่ 3.1

ตารางที่ 3.1 ชุดข้อมูล ข้อมูลทดสอบและรายละเอียดของแอททริบิวต์โดยสรุป

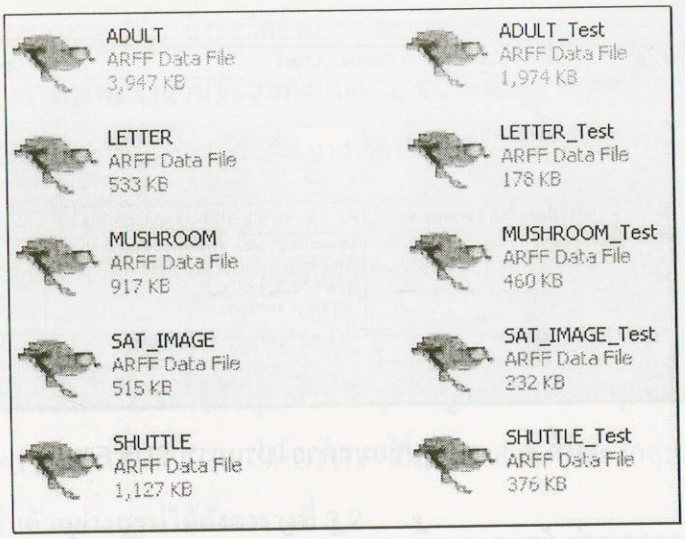
ชื่อข้อมูล	จำนวนข้อมูล	จำนวนข้อมูลทดสอบ	จำนวนแอททริบิวต์	จำนวนคลาส
1.Adult	32,561	16,281	14 (numeric = 6)	2
2.Letter	15,000	5,000	16 (all numeric)	26
3.Mushroom	5,416	2,708	22 (all nominal)	2
4.Shuttle	43,500	14,500	9 (all numeric)	7
5.Satellite Image	4,435	2,000	36 (all numeric)	6

ข้อมูลทั้งห้าชุดที่คัดเลือกมาจากแหล่งข้อมูล UCI จะต้องถูกแปลงให้อยู่ในรูปแบบ arff (attribute-relation file format) เพื่อให้โปรแกรม WEKA สามารถทำงานกับข้อมูลได้ รูปแบบไฟล์ arff ประกอบด้วยส่วนคำอธิบายข้อมูล และส่วนรายละเอียดข้อมูล ดังตัวอย่างในรูปที่ 3.1



รูปที่ 3.1 รูปแบบไฟล์ arff

ข้อมูล Adult, Letter , Mushroom , Shuttle และ Satellite Image ในรูปแบบ arff จะถูกแบ่งสองในสามเป็นข้อมูลฝึกและหนึ่งในสามเป็นข้อมูลทดสอบ ทำให้มีชุดข้อมูลรวมทั้งหมดในขั้นตอนนี้เป็น 10 ชุดข้อมูล ดังรูปที่ 3.2



รูปที่ 3.2 ข้อมูลฝึกและข้อมูลทดสอบในรูปแบบไฟล์ arff

ข้อมูลทดสอบ (ADULT_Test, LETTER_Test, MUSHROOM_Test, SAT_IMAGE_Test, SHUTTLE_Test) จะถูกกันไว้เพื่อใช้ในขั้นตอนทดสอบผลลัพธ์ ในส่วนของข้อมูลฝึก (ADULT, LETTER, MUSHROOM, SAT_IMAGE, SHUTTLE) แต่ละข้อมูลจะถูกนำมาสุ่มด้วย 3 เทคนิคพื้นฐาน คือ simple random sampling, systematic random sampling และ stratified random sampling ขนาดของการสุ่มจะเป็น 1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%, 15%, 20% โปรแกรมที่ใช้ช่วยในการสุ่มคือโปรแกรม SUT Filter (รูปที่ 3.3) พัฒนาขึ้นโดยทีมงานของหน่วยปฏิบัติการวิจัยวิศวกรรมข้อมูลและการค้นหาความรู้ มหาวิทยาลัยเทคโนโลยีสุรนารี ดังนั้นแต่ละชุดข้อมูลจะมีจำนวนข้อมูลสุ่มดังนี้

ADULT → simple random → 1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%, 15%, 20%

ADULT → systematic random → 1%, 2%, 3%, 4%, 5%, 6%,7%,8%,9%,10%,15%, 20%

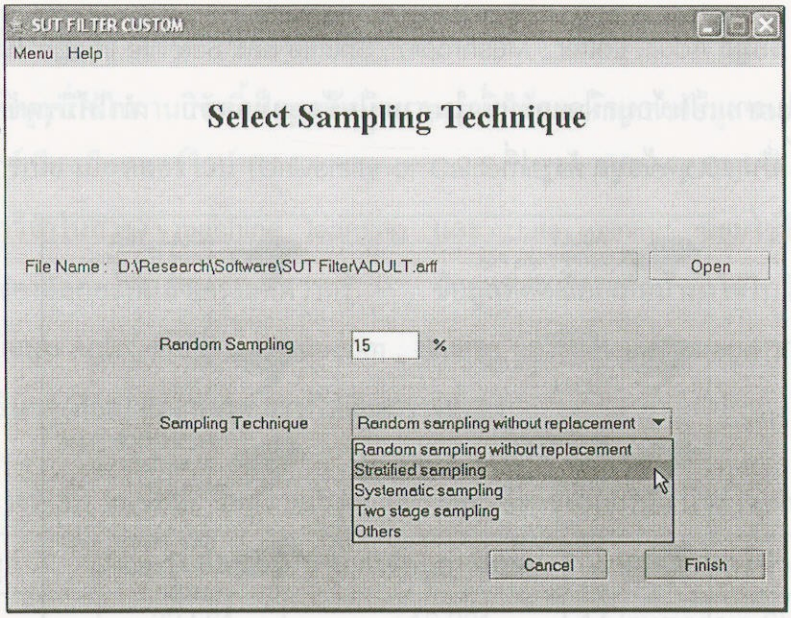
ADULT → stratified random → 1%, 2%, 3%, 4%, 5%, 6%,7%,8%,9%, 10%, 15%, 20%

... ...

SHUTTLE → simple random → 1%, 2%, 3%, 4%, 5%, 6%,7%,8%,9%, 10%, 15%, 20%

SHUTTLE → systematic random → 1%,2%,3%, 4%, 5%,6%,7%,8%,9%,10%,15%,20%

SHUTTLE → stratified random → 1%, 2%, 3%,4%,5%,6%,7%,8%,9%, 10%, 15%, 20%



รูปที่ 3.3 การสุ่มข้อมูลด้วยโปรแกรม SUT Filter

นอกจากการสุ่มข้อมูลแบบพื้นฐานที่อาศัยความน่าจะเป็นแล้ว งานวิจัยนี้ยังทดสอบการสุ่มข้อมูลแบบก้าวหน้า (progressive sampling) ซึ่งใช้วิธีการเพิ่มปริมาณข้อมูลในกลุ่มตัวอย่างเป็นลำดับขั้นขึ้นไป ลักษณะการเพิ่มขนาดของกลุ่มตัวอย่างเป็นได้สองลักษณะ คือเพิ่มแบบเลขคณิต (arithmetic progressive sampling) ที่มีอัตราการเพิ่มคงที่ และเพิ่มแบบเรขาคณิต (geometric progressive sampling) ที่อัตราการเพิ่มเป็นแบบก้าวกระโดด การสุ่มแบบก้าวหน้าได้ถูกนำมาใช้ในการทำเหมืองข้อมูลแบบเพิ่มพูน (incremental data mining) ซึ่งมีขั้นตอนการทำงานแสดงได้ดังอัลกอริทึมต่อไปนี้

Input: Schedule $S = \{n_1, n_2, ..., n_k\}$ of sample sizes when $n_j > n_i$ for $j > i$,
and n_k is all training data.
Output: An accurate model M that is trained from a sample n_i.

1. Initialize accuracy Acc = 10, i = 1
2. M ← model induced from sample n_i
3. While $Acc_M > Acc$ // Acc_M is accuracy of model M
   3.1) $i \leftarrow i + 1$
   3.2) $Acc \leftarrow Acc_M$
   3.3) M ← model induced from sample n_i
   3.4) end while
4. Return M

ในงานวิจัยนี้ทำการสุ่มตัวอย่างทั้งสองลักษณะ คือ สุ่มแบบก้าวหน้าเชิงเลขคณิต และ สุ่มแบบก้าวหน้าเชิงเรขาคณิต

การสุ่มแบบก้าวหน้าเชิงเลขคณิตนั้นจะมีการเพิ่มของข้อมูล ดังสมการต่อไปนี้

$$S_a = n_1 + (i * n_\delta)$$

เมื่อ $i \geq 0$ และ n_δ = จำนวนข้อมูลเริ่มต้นซึ่งจะมีค่าคงที่ ในการทดลองจะให้ $n_1 = 100$, i มีค่าเป็น 0, 1, 2, ... และ $n_\delta = 200$ ซึ่งจะได้จำนวนข้อมูลตัวอย่างดังนี้ 100, 300, 500, 700, ..., 3300, 3500, all data ทำให้มีจำนวนข้อมูลทั้งหมด 18 ขนาดหรือ 18 ชุดข้อมูลย่อย

ในส่วนของการสุ่มแบบก้าวหน้าเชิงเรขาคณิตนั้นจะมีการเพิ่มของข้อมูล ดังสมการต่อไปนี้

$$S_g = a^i * n_1$$

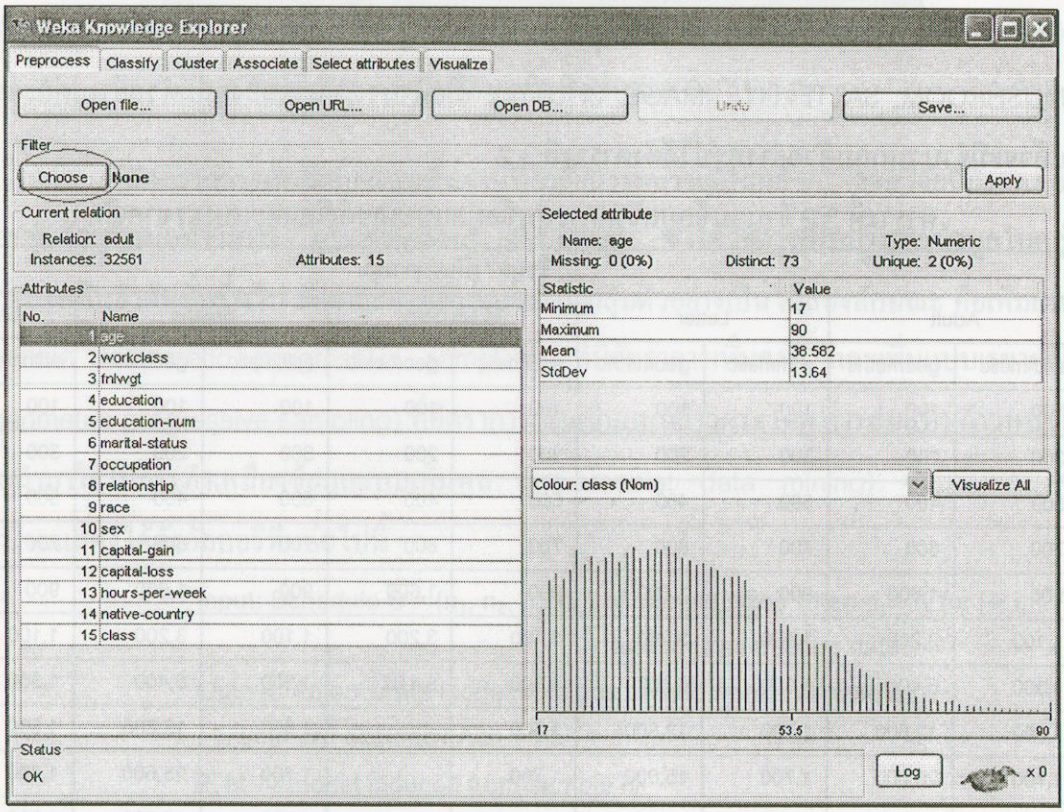
เมื่อ $a = 2, i \geq 0$ และ $n_1 = 100$ ซึ่งจะได้ข้อมูลตัวอย่างดังนี้ 100, 200, 400, 800, 1600, 3200, 6400, 12800, 25600, all data ซึ่งมีทั้งหมด 10 ชุดข้อมูลด้วยกัน ในกรณีที่ข้อมูลตั้งต้นชุดใดมีจำนวนไม่ถึง 25,600 เรคคอร์ด จะใช้ข้อมูลเท่าที่มี ชุดข้อมูลตั้งต้นทั้ง 5 ชุดคือ ADULT, LETTER, MUSHROOM, SATTELLITE IMAGE, SHUTTLE เมื่อคำนวณขนาดของกลุ่มตัวอย่างแต่ละขนาดแล้วจะมีจำนวนกลุ่มตัวอย่างสรุปได้ดังตารางที่ 3.2

ตารางที่ 3.2 จำนวนข้อมูลจากการสุ่มข้อมูลแบบเลขคณิตและแบบเรขาคณิต

ข้อมูล / รูปแบบการสุ่ม									
Adult		Letter		Mushroom		Shuttle		Satellite Image	
arithmetic	geometric	arithmetic	geometric	arithmetic	geometric	arithmetic	geometric	arithmetic	geometric
100	100	100	100	100	100	100	100	100	100
300	200	300	200	300	200	300	200	300	200
500	400	500	400	500	400	500	400	500	400
700	800	700	800	700	800	700	800	700	800
900	1,600	900	1,600	900	1,600	900	1,600	900	1,600
1,100	3,200	1,100	3,200	1,100	3,200	1,100	3,200	1,100	3,200
1,300	6,400	1,300	6,400	1,300	5,416	1,300	6,400	1,300	4,435
1,500	12,800	1,500	12,800	1,500		1,500	12,800	1,500	
1,700	25,600	1,700	15,000	1,700		1,700	25,600	1,700	
1,900	32,561	1,900		1,900		1,900	43,500	1,900	
2,100		2,100		2,100		2,100		2,100	
2,300		2,300		2,300		2,300		2,300	
2,500		2,500		2,500		2,500		2,500	
2,700		2,700		2,700		2,700		2,700	
2,900		2,900		2,900		2,900		2,900	
3,100		3,100		3,100		3,100		3,100	
3,300		3,300		3,300		3,300		3,300	
3,500		3,500		3,500		3,500		3,500	
32,561		15,000		5,416		43,500		4,435	

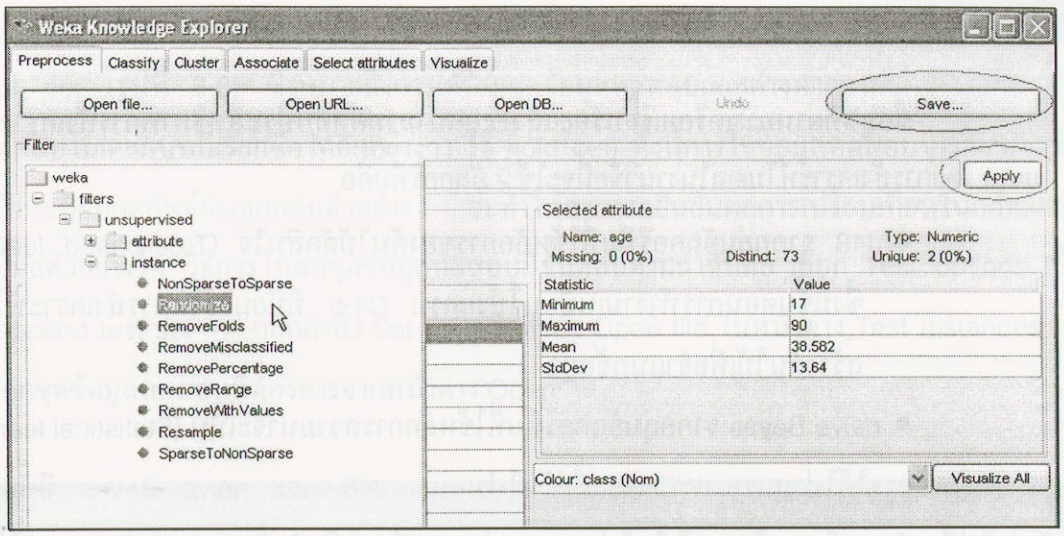
เมื่อคำนวณขนาดของกลุ่มตัวอย่างได้แล้ว ขั้นตอนต่อไปจะเป็นการเตรียมข้อมูลทั้ง 5 ชุดให้มีจำนวนของข้อมูล(จำนวนเรคคอร์ด) ในกลุ่มตัวอย่างแต่ละกลุ่มได้ตามที่กำหนดไว้ในตารางที่ 3.2 โดยจะได้ข้อมูลทั้งหมด 138 ชุด

การเตรียมข้อมูลตัวอย่างทั้ง 138 ชุดมีขั้นตอนดังต่อไปนี้ คือ เปิดใช้โปรแกรม WEKA เข้าไปในหน้าต่าง Preprocess เลือกปุ่ม Open file แล้วเลือกชื่อไฟล์ที่จะทำการสุ่มข้อมูล (เช่น ไฟล์ adult.arff) จากนั้นคลิกที่ปุ่ม Choose (รูปที่ 3.4) เพื่อเรียกใช้ฟิลเตอร์ให้ทำการสุ่มสลับลำดับเรคคอร์ด (เรคคอร์ดจะถูกเรียกว่า instance ในโปรแกรม WEKA) เพื่อลด bias ในการคัดเลือกข้อมูลเป็นกลุ่มตัวอย่าง



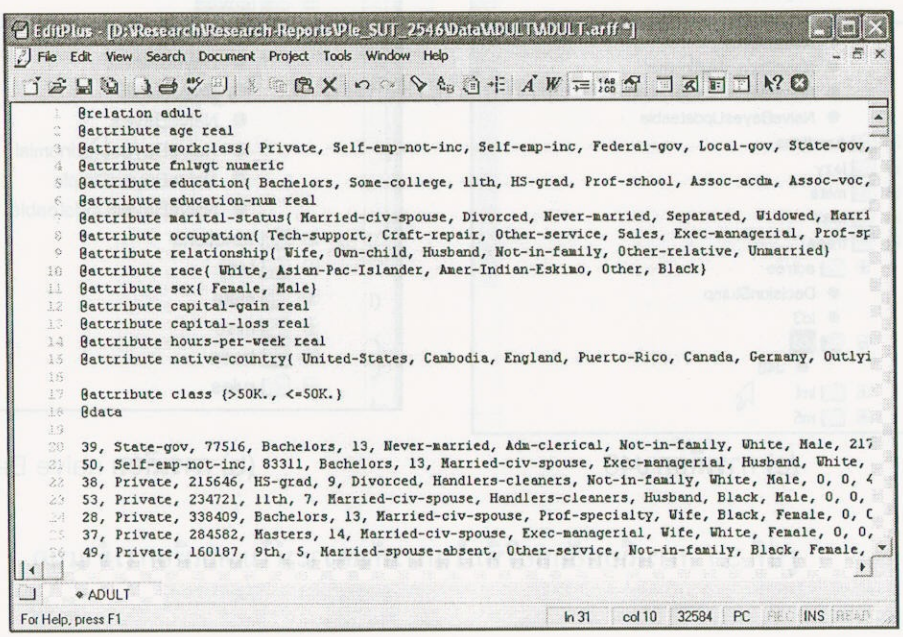
รูปที่ 3.4 การโหลดข้อมูลเข้าสู่โปรแกรม WEKA และเลือกใช้ฟิลเตอร์

วิธีการสุ่มสลับลำดับเรคคอร์ดทำได้ดังนี้ คลิกที่ปุ่ม Choose แล้วเลือก filters --> unsupervised --> instance --> Randomize (รูปที่ 3.5) เลือกปุ่ม Apply แล้วทำการบันทึกข้อมูลที่ถูกสลับตำแหน่งเรคคอร์ดแล้วด้วยการคลิกที่ปุ่ม Save โดยให้ทำการบันทึกเป็นชื่อไฟล์ชื่อใหม่



รูปที่ 3.5 การสุ่มเพื่อสลับลำดับข้อมูล

เมื่อได้ข้อมูลที่มีการสุ่มสลับลำดับแล้วขั้นตอนต่อไปจะเป็นการคัดเลือกเรคคอร์ดตามจำนวนที่ต้องการโดยใช้โปรแกรม Text Editor ที่มีการแสดงเลขบรรทัดเพื่อความสะดวกในการตัดบรรทัด โดยการวิจัยนี้ใช้โปแกรม EditPlus การคัดเลือกเรคคอร์ดจะใช้ค่าตามที่คำนวณไว้ในตารางที่ 3.1 เช่น ข้อมูล Adult กลุ่มตัวอย่างแรกต้องมี 100 เรคคอร์ด ก็จะคัดเลือกไว้เฉพาะข้อมูลบรรทัดที่ 20 ถึง 119 (รูปที่ 3.6) เสร็จแล้วก็จะทำการบันทึกเป็นไฟล์ใหม่ จากนั้นทำกระบวนการคัดเลือกกลุ่มตัวอย่างซ้ำเหมือนเดิมเพื่อคัดเลือกข้อมูล Adult กลุ่มที่สองที่ต้องมี 300 เรคคอร์ดโดยใช้ข้อมูลบรรทัดที่ 20 ถึง 319 บันทึกเป็นไฟล์ใหม่ ทำเช่นนี้จนกระทั่งได้ข้อมูลของทั้ง 5 ชุดข้อมูลครบทั้ง 138 กลุ่มตัวอย่าง



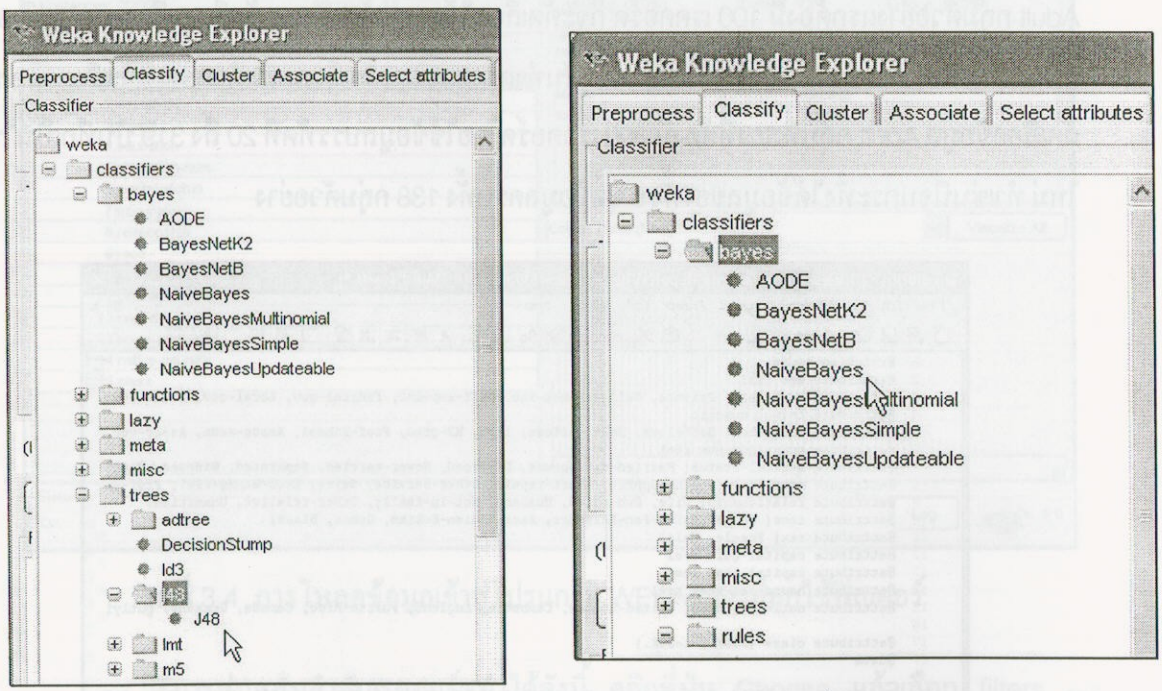
รูปที่ 3.6 การใช้โปรแกรมเอดิเตอร์เพื่อสุ่มเลือกข้อมูลตัวอย่าง

3.3 วิธีการทดสอบและวิเคราะห์ผล

ข้อมูลที่ผ่านการเตรียมเรียบร้อยแล้วจะถูกนำมาทดสอบประสิทธิภาพการสังเคราะห์โมเดล โดยการสังเคราะห์โมเดลในงานวิจัยนี้จะใช้ 2 อัลกอริทึมคือ

- J48 จากกลุ่มอัลกอริทึมที่ใช้หลักการของต้นไม้ตัดสินใจ (Tree-based learner) ซึ่งมีขั้นตอนการทำงานเหมือนโปรแกรม C4.5 ที่นิยมใช้ในการสังเคราะห์โครงสร้างต้นไม้เพื่อจำแนกข้อมูล
- naive Bayes จากกลุ่มอัลกอริทึมที่ใช้หลักการความน่าจะเป็น (Statistical learner)

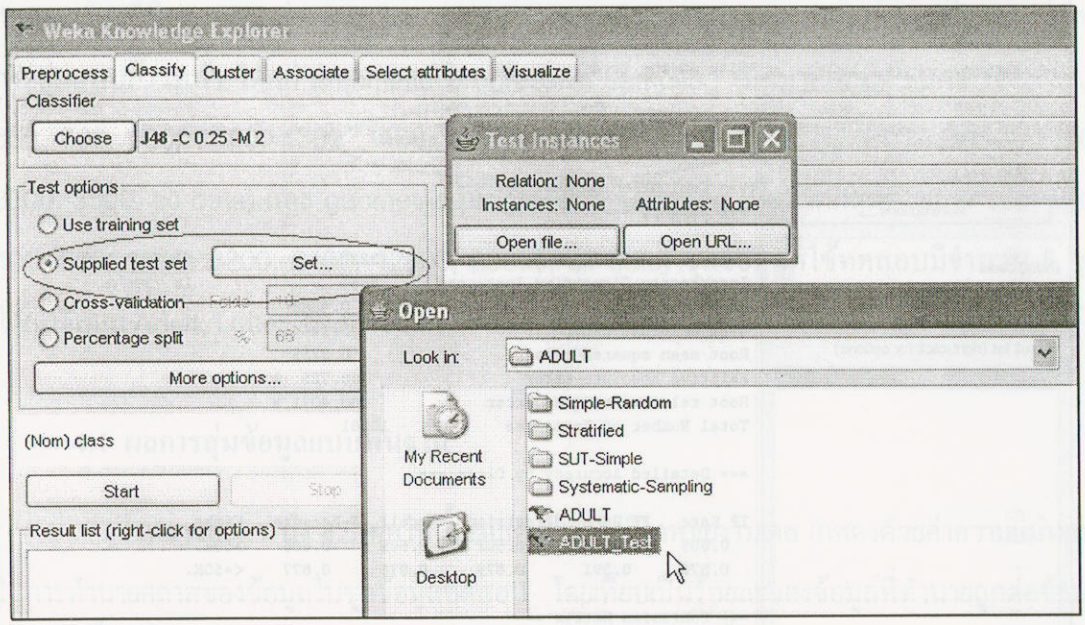
การใช้โปรแกรม WEKA เพื่อรันโปรแกรม J48 และ naive Bayes มีขั้นตอนดังต่อไปนี้ ทำการโหลดข้อมูลเข้าในโปรแกรม WEKA เลือกแท็บคำสั่ง Classify (ตามรูปที่ 3.7) เลือกปุ่ม Choose เลือก Weka --> Classifier --> trees --> j48 --> J48 สำหรับอัลกอริทึม J48 (รูปที่ 3.7a) หรือ Weka --> Classifier --> bayes --> NaiveBayes สำหรับอัลกอริทึม naive Bayes (รูปที่ 3.7b)



(a) การเลือก J48 (b) การเลือก naive Bayes

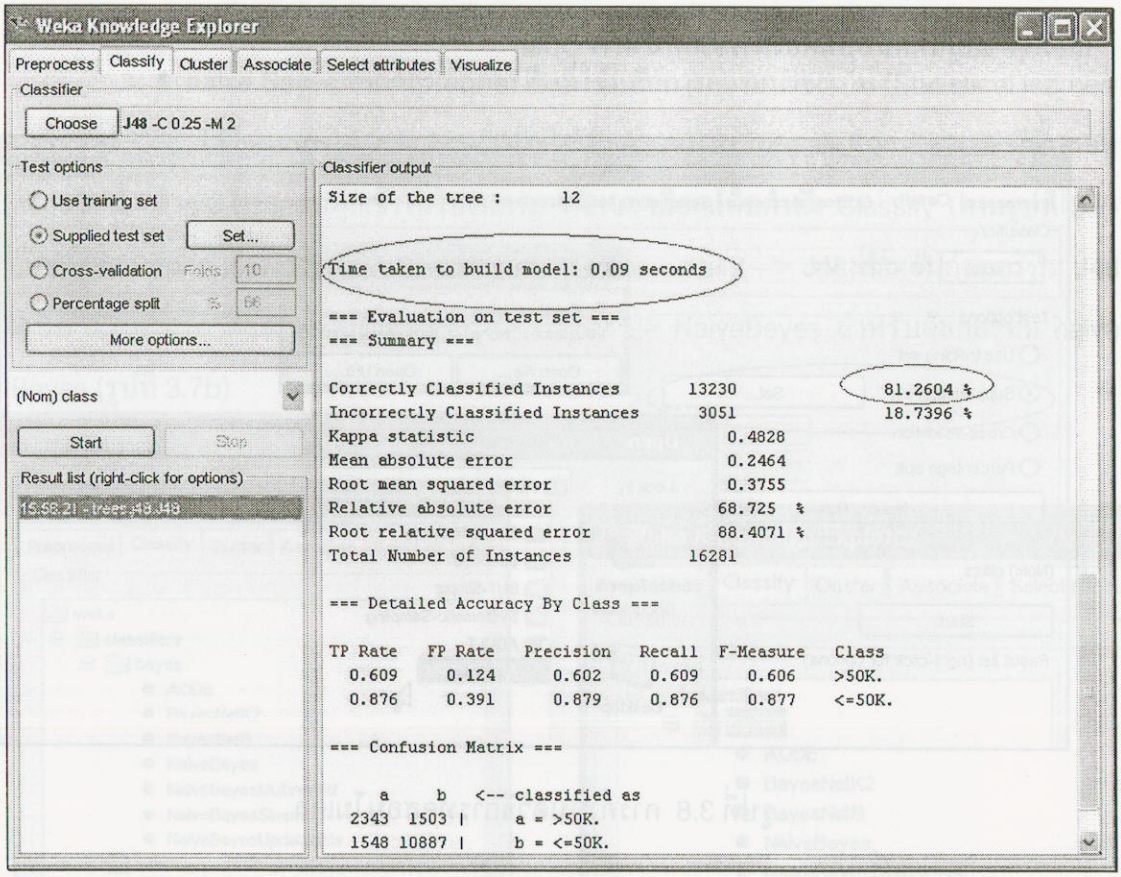
รูปที่ 3.7 การเลือกอัลกอริทึมในการสังเคราะห์โมเดลเพื่อการจำแนก

การรันโปรแกรม J48 และ naiveBayes จะเลือกการทดสอบโมเดลเป็น Supplied test set (รูปที่ 3.8) ซึ่งหมายถึงการเตรียมชุดข้อมูลทดสอบแยกต่างหากจากชุดข้อมูลฝึก (วิธีทดสอบโมเดลแบบนี้เรียกชื่อได้อีกอย่างว่า วิธี hold out) ทั้งนี้การใช้ข้อมูลทดสอบชุดเดียวกันเพื่อเปรียบเทียบผลลัพธ์กับทุกกลุ่มตัวอย่าง จะทำให้สามารถยืนยันผลการเปรียบเทียบในแต่ละกลุ่มข้อมูลตัวอย่างได้ วิธีการโหลดชุดข้อมูลทดสอบมีขั้นตอนตามลำดับคือ เลือก Test options เป็น Supplied test set --> คลิกคำสั่ง Set --> คลิกคำสั่ง Open file ในหน้าต่าง Test Instances --> เลือกชุดข้อมูลทดสอบที่ต้องการจากหน้าต่าง Open



รูปที่ 3.8 การกำหนดวิธีการทดสอบโมเดล

การวัดประสิทธิภาพของการสังเคราะห์โมเดลจะวัดจากค่าความแม่นตรง ซึ่งสังเกตได้จากค่า Correctly Classified Instances ที่รายงานในผลลัพธ์ของการรันโปรแกรม WEKA (รูปที่ 3.9) นอกเหนือจากค่าความแม่นตรงแล้ว การทดสอบนี้ยังบันทึกเวลาที่ใช้ในการสร้างโมเดล (Time taken to build model) เพื่อใช้ประโยชน์ในขั้นตอนการวิเคราะห์เปรียบเทียบผลการทำงานของกลุ่มตัวอย่างแต่ละขนาด



รูปที่ 3.9 ผลลัพธ์ของการสังเคราะห์โมเดลเพื่อการจำแนก

บทที่ 4

ผลการศึกษาการลดขนาดข้อมูลด้วยการสุ่ม

การสังเคราะห์โมเดลเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลด้วยการสุ่ม ใช้อัลกอริทึมพื้นฐาน 2 อัลกอริทึม ได้แก่ J48 (decision-tree induction) และ naive Bayes ประมวลผลด้วยโปรแกรม WEKA version 3.4 บนเครื่องคอมพิวเตอร์ส่วนบุคคลความเร็วซีพียู 700 MHz หน่วยความจำหลัก 512 MB เทคนิคการสุ่มข้อมูลจำแนกเป็นสองกลุ่มหลักคือ การสุ่มแบบพื้นฐาน (ได้แก่ simple random sampling, systematic random sampling, stratified random sampling) ที่มีขนาดการสุ่ม 1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%, 15%, 20% และการสุ่มแบบก้าวหน้า ได้แก่ arithmetic progressive sampling (มีขนาดกลุ่มตัวอย่าง 100, 300, 500, 700, 900, 1100, 1300, 1500, 1700, 1900, 2100, 2300, 2500, 2700, 2900, 3100, 3300, 3500, all data) และ geometric progressive sampling (มีขนาดกลุ่มตัวอย่าง 100, 200, 400, 800, 1600, 3200, 6400, 12800, 25600, all data) ชุดข้อมูลที่ใช้ทดสอบมีจำนวน 5 ชุด ได้แก่ข้อมูล Adult, Letter, Mushroom, Shuttle, Satellite Image

4.1 ผลการสุ่มข้อมูลแบบพื้นฐาน

ตารางที่ 4.1 ถึง 4.5 ต่อไปนี้จะแสดงประสิทธิภาพของโมเดล (แสดงด้วยค่าความแม่นตรงในการทำนายคลาสของข้อมูลในชุดข้อมูลทดสอบ โดยเทียบเป็นร้อยละของข้อมูลที่ทำนายถูกต่อข้อมูลทดสอบทั้งหมด เช่นถ้าทำนายคลาสของข้อมูลทดสอบได้ถูกต้องทั้งหมด จะมีค่าความแม่นตรงเป็น 100%) และแสดงเวลาที่ใช้ในการสร้างโมเดลด้วยอัลกอริทึม J48 และ naive Bayes (เวลาวัดในหน่วยของวินาที) แต่ละตารางจะเป็นผลการทดสอบของแต่ละข้อมูล ได้แก่ข้อมูล Adult, Letter, Mushroom, Shuttle และ Satellite Image ตามลำดับ การรายงานผลจะระบุขนาดของข้อมูลในแต่ละกลุ่มตัวอย่างเป็นสองลักษณะ คือ จำนวนเรคคอร์ด หรือ instances ในกลุ่มตัวอย่าง และ การใช้เนื้อที่เก็บในหน่วยความจำ (ใช้หน่วย kilobytes, KB)

รูปที่ 4.1 ถึง 4.5 แสดงกราฟเปรียบเทียบค่าความแม่นตรงรวดถึงเวลาที่ใช้ในการสร้างโมเดลของอัลกอริทึม J48 และ naive Bayes บนข้อมูลทั้ง 5 ชุด

ตารางที่ 4.1 ประสิทธิภาพและเวลาในการสังเคราะห์โมเดลของข้อมูล ADULT

Sample size	Sampling technique	Number of instances	Data size (KB)	Accuracy (J48 model)	Accuracy (naiveBayes model)	Time (J48)	Time (naive Bayes)
100%		32,561	3947	85.85	83.13	40.05	1.28
1%	Simple	326	37	81.26	82.82	0.03	0.02
	Systematic	326	41	82.31	82.53	0.21	0.05
	Stratified	327	41	79.98	82.96	0.08	0.02
2%	Simple	652	72	83.9	83.45	0.08	0.01
	Systematic	652	81	83.94	82.62	0.16	0.05
	Stratified	652	80	83.36	83.05	0.11	0.06
3%	Simple	977	106	83.05	82.35	0.39	0.06
	Systematic	977	120	82.95	82.61	0.24	0.01
	Stratified	978	120	83.89	83.33	0.39	0.04
4%	Simple	1,303	141	82.86	82.13	0.4	0.05
	Systematic	1,303	158	83.88	82.93	0.58	0.07
	Stratified	1,303	159	83.31	83	0.31	0.02
5%	Simple	1,629	176	84.25	82.16	1.6	0.04
	Systematic	1,628	197	84	83.45	0.34	0.04
	Stratified	1,629	198	83.69	82.67	1.49	0.07
6%	Simple	1,954	210	83.82	82.47	0.75	0.09
	Systematic	1,954	237	84.73	83.43	1.25	0.09
	Stratified	1,955	237	84.09	82.65	0.76	0.06
7%	Simple	2,280	245	84.47	82.38	0.74	0.09
	Systematic	2,279	276	84.79	82.73	0.64	0.06
	Stratified	2,280	276	84.49	82.78	2.14	0.04
8%	Simple	2,605	279	84.03	82.4	1.92	0.05
	Systematic	2,605	315	84.48	83.66	1.78	0.05
	Stratified	2,606	315	83.87	83.51	1.73	0.14
9%	Simple	2,931	314	84.59	82.3	1.16	0.15
	Systematic	2,931	353	84.16	83.6	1.1	0.13
	Stratified	2,931	354	85.21	83.53	0.88	0.08
10%	Simple	3,257	349	84.29	82.92	1.22	0.12
	Systematic	3,256	393	84.67	83.32	1.14	0.1
	Stratified	3,257	392	85	83.34	1.17	0.11
15%	Simple	4,885	523	85.23	83.45	4.64	0.15
	Systematic	4,884	590	84.98	83.51	6.52	0.18
	Stratified	4,885	589	85.4	83.04	2.5	0.2
20%	Simple	6,513	696	85.71	82.43	3.14	0.28
	Systematic	6,512	784	85.33	83.75	3.32	0.25
	Stratified	6,513	785	84.85	82.71	3.29	0.26

ตารางที่ 4.2 ประสิทธิภาพและเวลาในการสังเคราะห์โมเดลของข้อมูล LETTER

Sample size	Sampling technique	Number of instances	Data size (KB)	Accuracy (J48 model)	Accuracy (naiveBayes model)	Time (J48)	Time (naive Bayes)
100%		15,000	533	87.7	63.56	25.57	1.48
1%	Simple	150	6	39.32	47.4	0.26	0.02
	Systematic	151	6	38.82	40.28	0.1	0.02
	Stratified	160	7	33.98	42.7	0.36	0.05
2%	Simple	300	12	50.72	51.52	0.2	0.01
	Systematic	301	12	48.46	52.26	0.21	0.01
	Stratified	313	12	49.34	51.04	0.23	0.03
3%	Simple	450	17	55.44	56.28	0.24	0.01
	Systematic	451	17	56.3	54	2.2	0.02
	Stratified	464	17	59.76	56.96	0.26	0.01
4%	Simple	600	22	58.2	58.28	0.35	0.02
	Systematic	601	22	60.86	57.56	1.1	0.07
	Stratified	611	23	56.86	60.44	0.35	0.02
5%	Simple	750	28	62.06	59.72	0.61	0.07
	Systematic	751	28	61.78	57.76	0.52	0.06
	Stratified	763	28	61.64	58.86	2.36	0.02
6%	Simple	900	33	63.18	61.98	0.61	0.05
	Systematic	901	33	63.84	60.04	0.6	0.04
	Stratified	912	33	61.24	59.5	1.21	0.09
7%	Simple	1,050	38	65.46	61.04	0.69	0.03
	Systematic	1,051	38	64.08	60.28	0.73	0.04
	Stratified	1,063	39	63.9	60.08	0.63	0.09
8%	Simple	1,200	44	65.24	61.42	0.8	0.04
	Systematic	1,201	44	66.46	60.22	0.92	0.13
	Stratified	1,213	44	65.3	59.22	0.74	0.04
9%	Simple	1,350	49	66.82	61.48	3.56	0.04
	Systematic	1,351	49	66.72	60.08	0.94	0.08
	Stratified	1,362	49	66.66	60.48	0.9	0.05
10%	Simple	1,500	54	69.22	60.7	1.26	0.24
	Systematic	1,501	54	69.08	61.08	1.01	0.06
	Stratified	1,512	55	67.58	61.58	1.2	0.07
15%	Simple	2,250	81	71.18	62.2	1.74	0.11
	Systematic	2,251	81	71.04	62.68	2.6	0.08
	Stratified	2,265	81	72.56	61.62	1.72	0.1
20%	Simple	3,000	107	75.6	61.44	2.7	0.15
	Systematic	3,001	107	75.76	63.62	9.64	0.18
	Stratified	3.010	108	74.52	61.78	2.46	0.13

ตารางที่ 4.3 ประสิทธิภาพและเวลาในการสังเคราะห์โมเดลของข้อมูล MUSHROOM

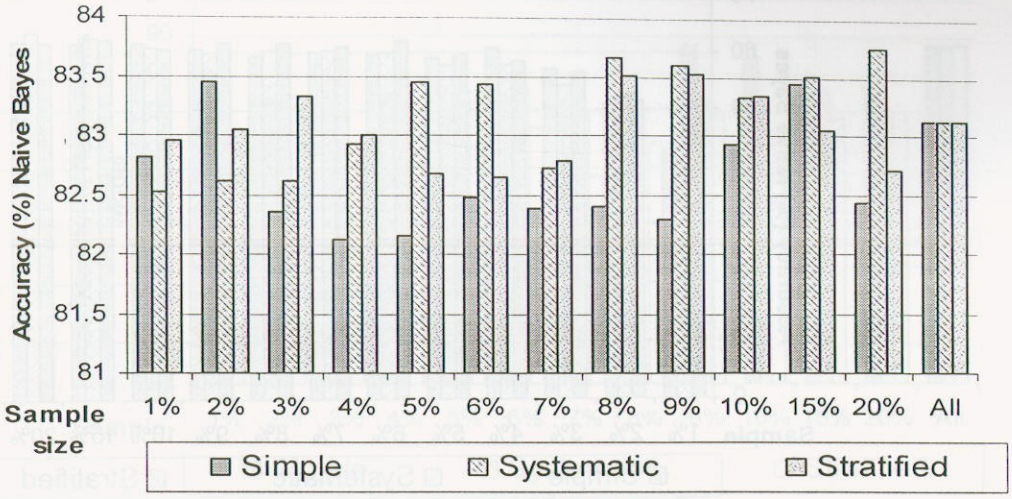
Sample size	Sampling technique	Number of instances	Data size (KB)	Accuracy (J48 model)	Accuracy (naiveBayes model)	Time (J48)	Time (naive Bayes)
100%		5,416	917	100	95.79	0.2	0.05
1%	Simple	55	10	95.42	94.98	0.07	0.01
	Systematic	55	10	97.86	90.69	0.06	0.02
	Stratified	55	11	97.86	92.39	0.01	0.02
2%	Simple	109	18	98.15	94.65	0	0
	Systematic	109	20	97.64	91.84	0.01	0.01
	Stratified	109	20	98.15	94.35	0	0.01
3%	Simple	163	26	98.15	94.53	0	0
	Systematic	163	30	98.15	91.65	0.01	0
	Stratified	163	29	98.45	90.48	0.01	0
4%	Simple	217	34	98.15	94.46	0	0
	Systematic	217	39	99.3	92.13	0.02	0
	Stratified	216	39	98.15	92.69	0	0
5%	Simple	271	42	98.15	94.5	0.01	0
	Systematic	271	48	98.15	94.53	0.03	0
	Stratified	272	48	97.86	93.13	0.03	0
6%	Simple	325	50	98.15	94.31	0.02	0.01
	Systematic	326	57	99.3	94.57	0.01	0
	Stratified	326	57	100	93.57	0.01	0.01
7%	Simple	380	58	98.15	94.28	0.03	0
	Systematic	380	66	98.15	93.98	0.01	0.02
	Stratified	380	66	98.15	92.32	0.06	0.02
8%	Simple	434	66	98.15	94.46	0.01	0
	Systematic	434	75	98.56	92.84	0.02	0.01
	Stratified	434	75	98.15	93.21	0.02	0.05
9%	Simple	488	74	98.15	95.75	0.05	0
	Systematic	488	85	99.74	94.35	0.04	0.01
	Stratified	489	85	98.15	93.91	0.03	0
10%	Simple	542	82	98.15	95.46	0.03	0.03
	Systematic	542	94	99.3	93.54	0.08	0.03
	Stratified	543	94	99.7	93.21	0.03	0.03
15%	Simple	813	122	99.7	94.53	0.04	0.01
	Systematic	813	140	99.3	94.53	0.04	0.01
	Stratified	813	139	99.74	93.87	0.03	0
20%	Simple	1,084	162	99.7	94.65	0.03	0.02
	Systematic	1,084	185	99.78	94.05	0.04	0.01
	Stratified	1,084	185	99.74	93.98	0.03	0.02

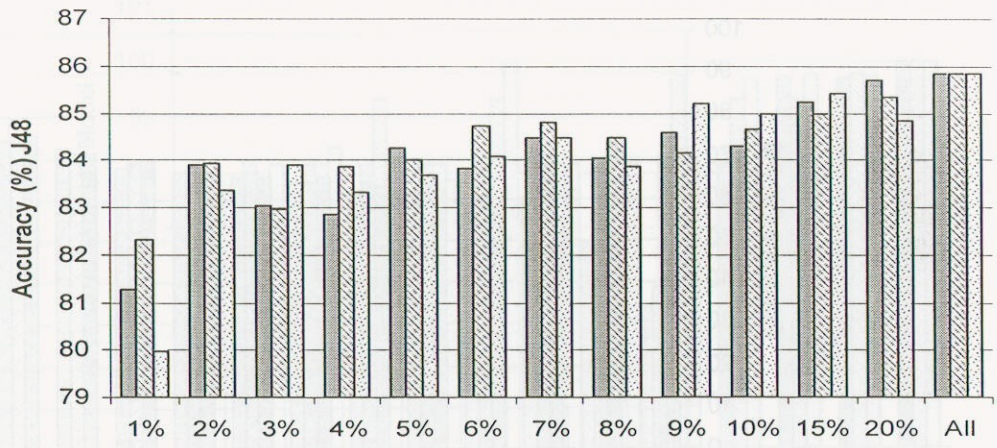
ตารางที่ 4.4 ประสิทธิภาพและเวลาในการสังเคราะห์โมเดลของข้อมูล SHUTTLE

Sample size	Sampling technique	Number of instances	Data size (KB)	Accuracy (J48 model)	Accuracy (naiveBayes model)	Time (J48)	Time (naive Bayes)
100%		43,500	1127	99.95	92.21	21.1	2.64
1%	Simple	435	12	99.58	93.72	0.06	0.02
	Systematic	436	12	99.54	94.24	0.02	0.01
	Stratified	440	12	99.57	88.25	0.15	0.01
2%	Simple	870	23	99.59	92.92	0.16	0.07
	Systematic	871	23	99.54	94.79	0.18	0.06
	Stratified	874	23	99.15	93.87	0.19	0.07
3%	Simple	1,305	35	99.59	94.76	0.08	0.05
	Systematic	1,306	34	99.62	94.44	0.16	0.05
	Stratified	1,309	35	99.59	89.19	0.13	0.05
4%	Simple	1,740	46	99.49	92.52	0.17	0.03
	Systematic	1,741	46	99.52	90.68	0.26	0.04
	Stratified	1,744	46	99.71	89.96	0.15	0.05
5%	Simple	2,175	57	99.49	91.66	0.3	0.11
	Systematic	2,176	57	99.7	93.4	0.34	0.13
	Stratified	2,178	57	99.65	95.04	0.34	0.5
6%	Simple	2,610	68	99.61	84.82	0.37	0.08
	Systematic	2,611	68	99.59	86.52	0.25	0.1
	Stratified	2,613	68	99.52	91.93	0.38	0.11
7%	Simple	3,045	80	99.59	85.95	0.37	0.11
	Systematic	3,046	79	99.66	91.31	0.32	0.08
	Stratified	3,049	80	99.64	89.37	0.38	0.1
8%	Simple	3,480	91	99.59	88.19	0.71	0.2
	Systematic	3,481	91	99.63	92.88	0.59	0.17
	Stratified	3,482	91	99.7	90.81	0.75	0.17
9%	Simple	3,915	102	99.61	87.38	0.7	0.17
	Systematic	3,916	102	99.73	90.54	0.7	0.18
	Stratified	3,918	102	99.72	88.29	0.89	0.13
10%	Simple	4,350	113	99.64	87.94	2.23	0.14
	Systematic	4,351	113	99.63	84.7	0.8	0.14
	Stratified	4,353	114	99.68	89.04	1.93	0.3
15%	Simple	6,525	170	99.77	89.26	2.21	0.33
	Systematic	6,526	170	99.81	90.51	2.16	0.32
	Stratified	6,528	170	99.83	89.52	1.59	0.34
20%	Simple	8,700	226	99.77	87.3	2.02	0.36
	Systematic	8,701	226	99.81	93.23	2.18	0.35
	Stratified	8,704	226	99.82	91.7	3.27	0.41

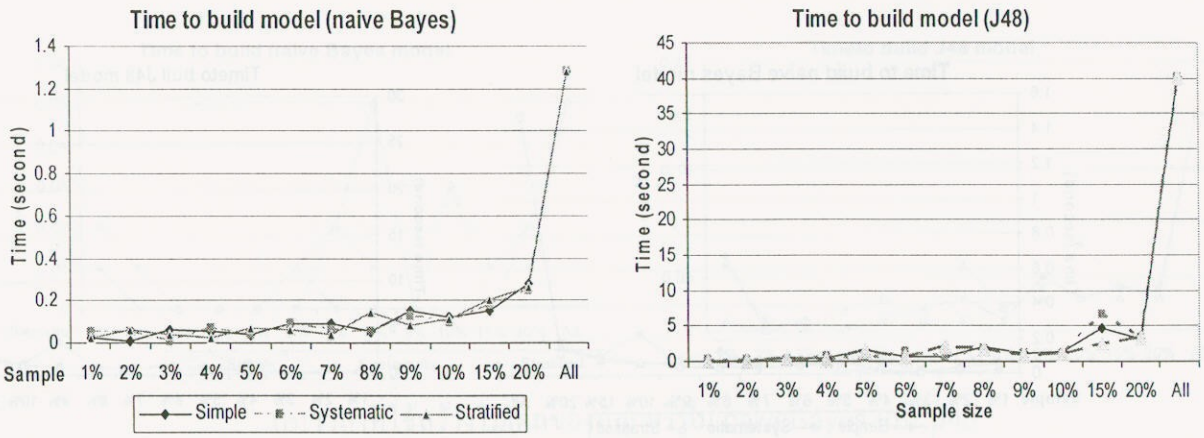
ตารางที่ 4.5 ประสิทธิภาพและเวลาในการสังเคราะห์โมเดลของข้อมูล SATELLITE IMAGE

Sample size	Sampling technique	Number of instances	Data size (KB)	Accuracy (J48 model)	Accuracy (naiveBayes model)	Time (J48)	Time (naive Bayes)
100%		4,435	515	85.35	79.6	2.82	0.31
1%	Simple	45	7	30.55	22.85	0	0.01
	Systematic	45	6	69.55	73.6	0.01	0.01
	Stratified	47	7	70.1	75.4	0.01	0
2%	Simple	89	12	51.35	41.15	0.02	0
	Systematic	89	12	76.7	81.5	0.52	0
	Stratified	92	12	74.2	79.15	0.03	0
3%	Simple	134	17	49.45	42.95	0.02	0.01
	Systematic	134	17	75.2	78.8	0.09	0.02
	Stratified	137	17	72.45	79.45	0.03	0.01
4%	Simple	178	22	50.4	42.2	0.02	0.01
	Systematic	178	22	78	80.5	0.06	0.01
	Stratified	180	22	77.2	79.8	0.04	0.01
5%	Simple	222	27	49.95	45.85	0.04	0.01
	Systematic	222	27	79.05	80.2	0.07	0
	Stratified	224	27	78.9	78.95	0.06	0.01
6%	Simple	267	32	52.15	43.9	0.05	0.01
	Systematic	267	32	78.9	79.2	0.08	0.01
	Stratified	269	32	78.1	80.05	0.07	0.01
7%	Simple	311	37	53.45	45.65	0.07	0.01
	Systematic	311	37	79.05	80	0.09	0
	Stratified	314	38	74.75	79.55	0.14	0.04
8%	Simple	355	42	52.8	45.2	0.07	0.01
	Systematic	355	42	78.35	78.5	0.1	0.01
	Stratified	358	43	77.25	79.4	0.12	0.01
9%	Simple	400	48	51.25	47.1	0.07	0.01
	Systematic	400	48	77.65	79.8	0.12	0.01
	Stratified	403	48	78.25	78.1	0.12	0.02
10%	Simple	444	53	59.45	49.75	0.11	0.01
	Systematic	444	53	80.15	79.4	0.13	0.01
	Stratified	446	53	80.65	79.05	0.13	0.01
15%	Simple	666	78	65.25	72.15	0.19	0.03
	Systematic	666	78	81.4	78.8	0.2	0.02
	Stratified	668	78	78.4	79.7	0.21	0.02
20%	Simple	887	104	75.55	73.9	0.76	0.03
	Systematic	888	105	82.65	78.75	0.26	0.02
	Stratified	889	104	80.65	79.7	0.27	0.03



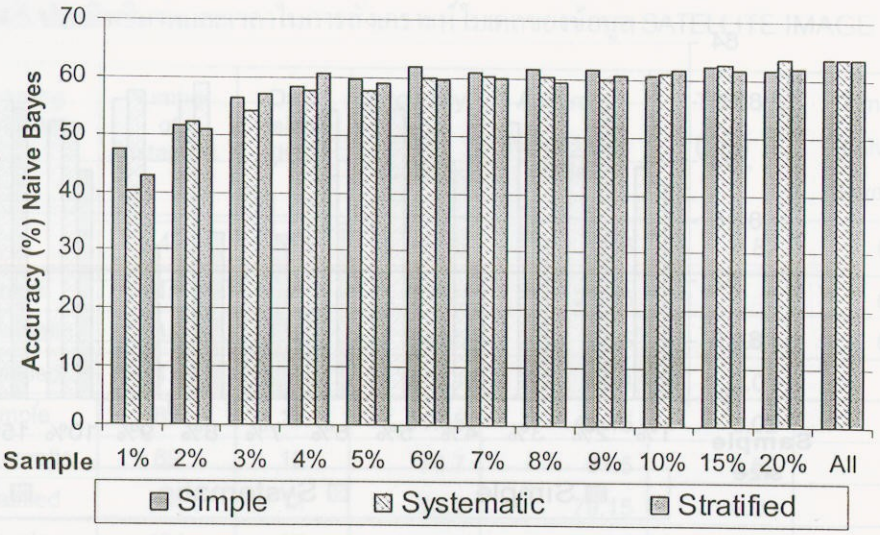


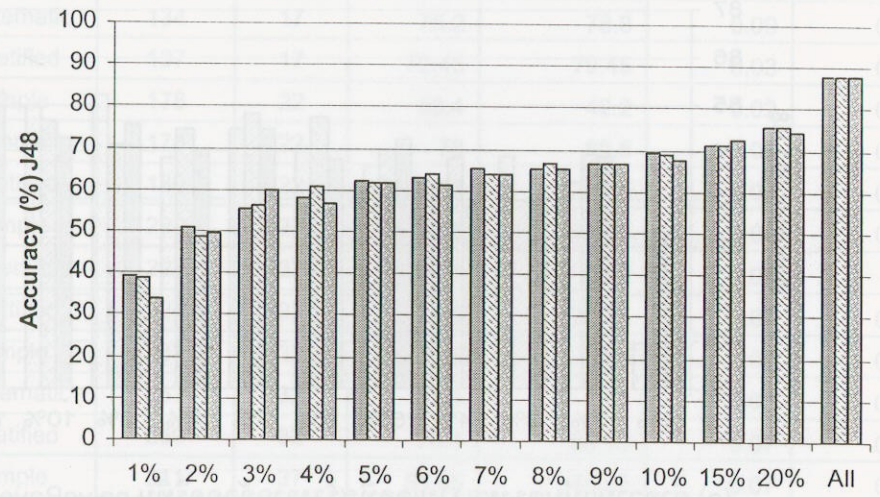
(a) ค่าความแม่นตรงของโมเดลที่สร้างจากอัลกอริทึม naiveBayes และ J48



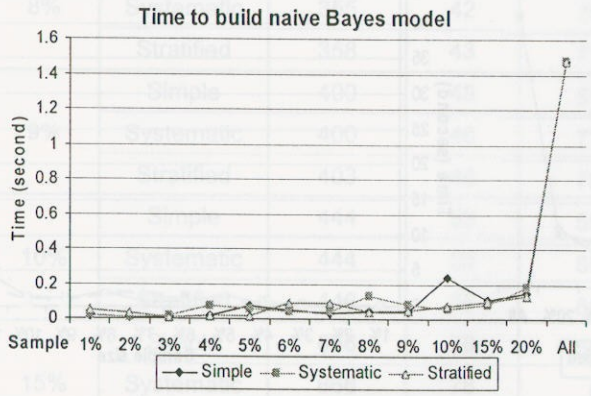
(b) เวลาที่ใช้สร้างโมเดลของอัลกอริทึม naiveBayes และ J48

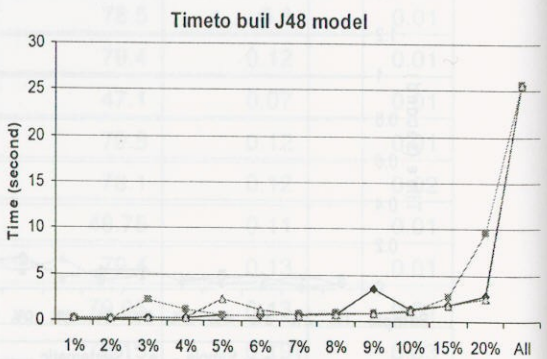
รูปที่ 4.1 เปรียบเทียบค่าความแม่นตรงของโมเดลและเวลาที่ใช้ในการสร้างโมเดลบนชุดข้อมูล ADULT





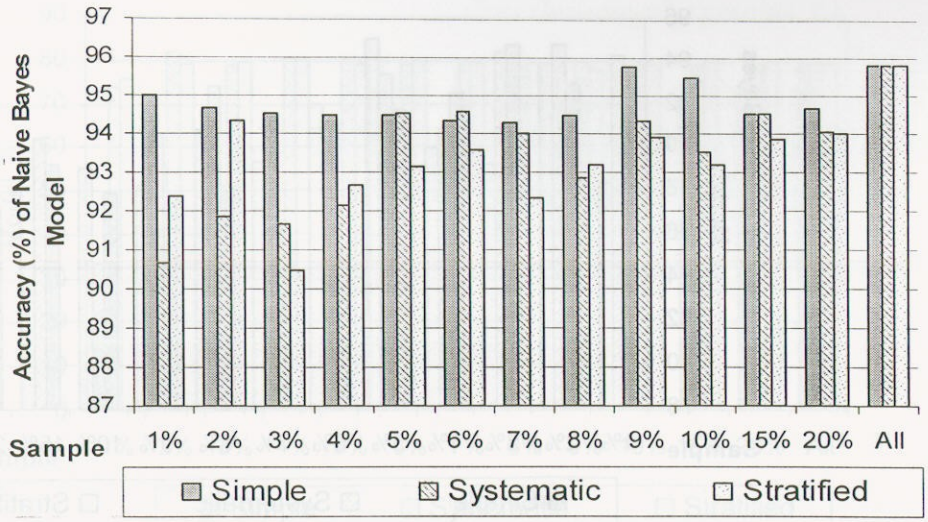
(a) ค่าความแม่นตรงของโมเดลที่สร้างจากอัลกอริทึม naiveBayes และ J48

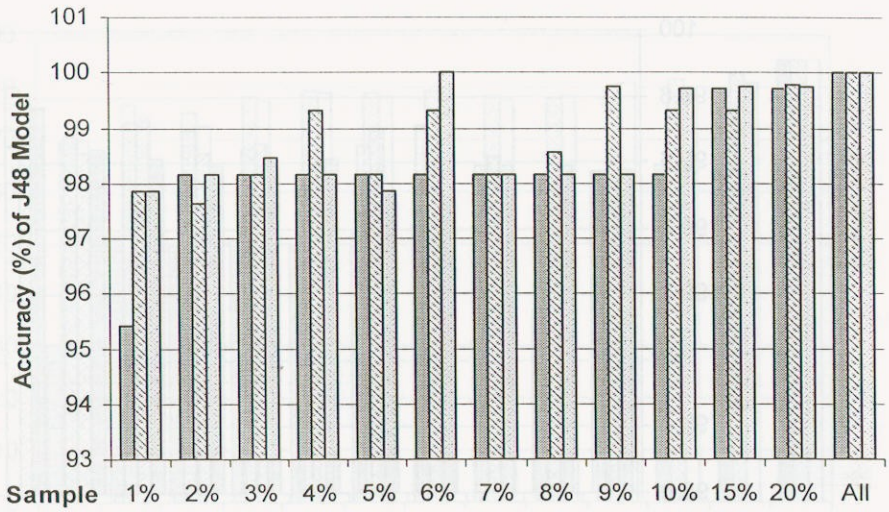




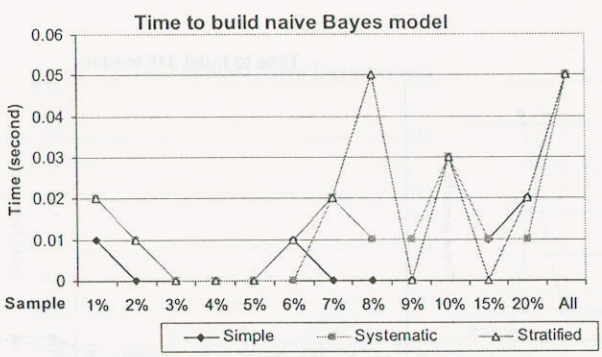
(b) เวลาที่ใช้สร้างโมเดลของอัลกอริทึม naiveBayes และ J48

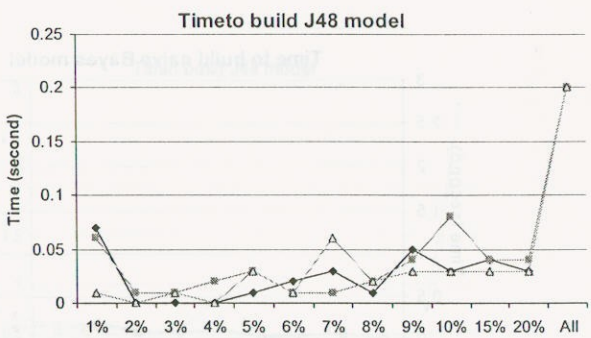
รูปที่ 4.2 เปรียบเทียบค่าความแม่นตรงของโมเดล และ เวลาที่ใช้ในการสร้างโมเดล บนชุดข้อมูล LETTER





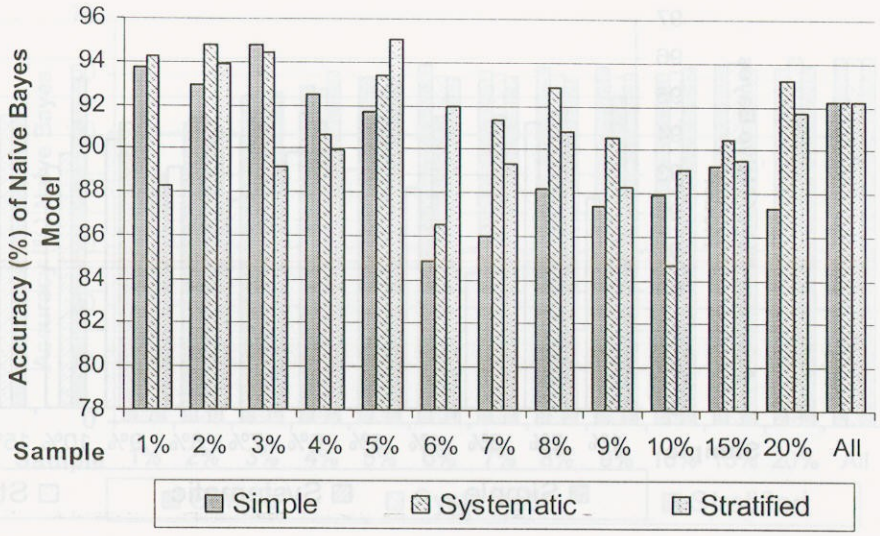
(a) ค่าความแม่นตรงของโมเดลที่สร้างจากอัลกอริทึม naiveBayes และ J48

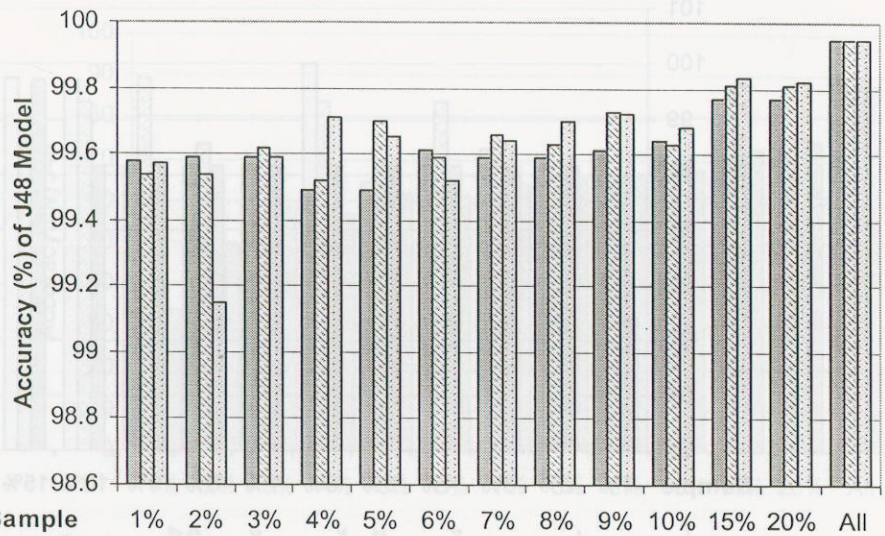




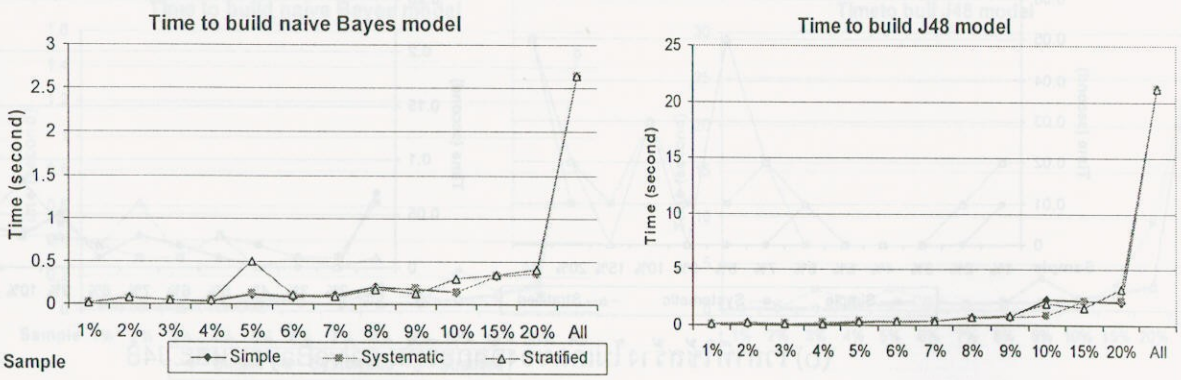
(b) เวลาที่ใช้สร้างโมเดลของอัลกอริทึม naiveBayes และ J48

รูปที่ 4.3 เปรียบเทียบค่าความแม่นตรงของโมเดล และ เวลาที่ใช้ในการสร้างโมเดล บนชุดข้อมูล MUSHROOM



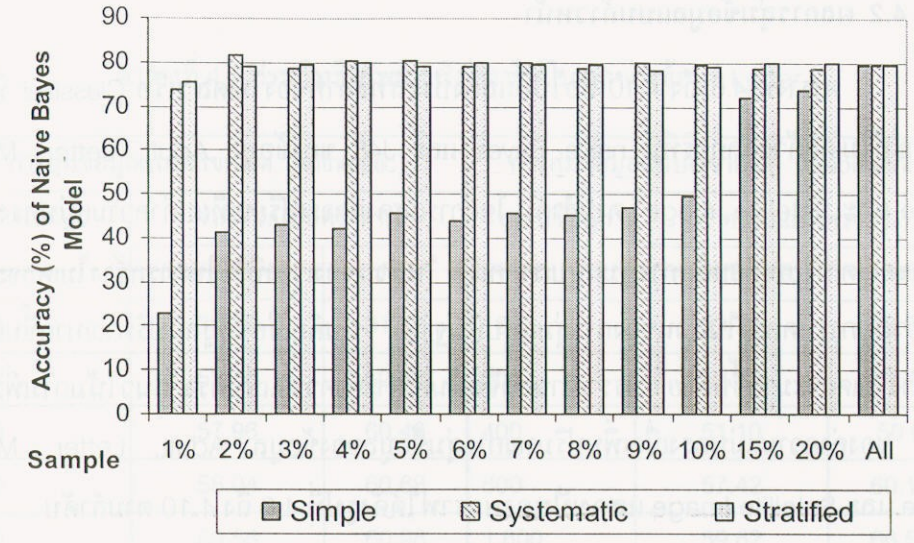


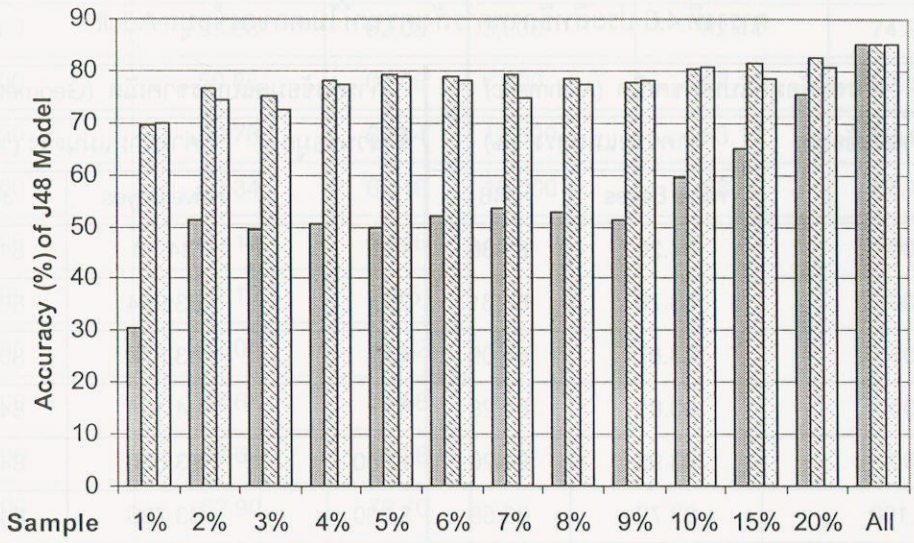
(a) ค่าความแม่นตรงของโมเดลที่สร้างจากอัลกอริทึม naiveBayes และ J48



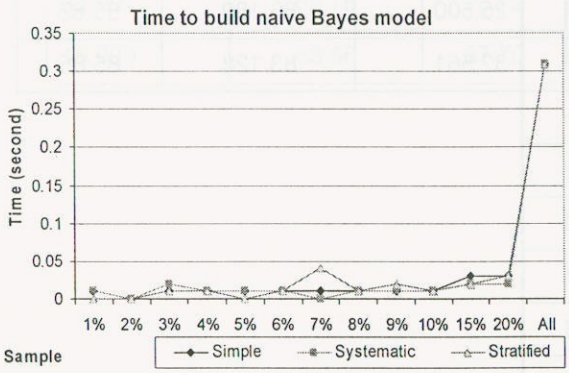
(b) เวลาที่ใช้สร้างโมเดลของอัลกอริทึม naiveBayes และ J48

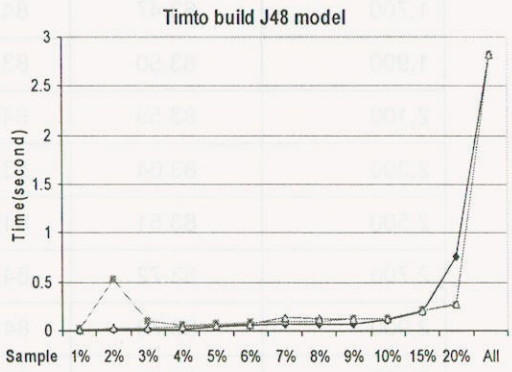
รูปที่ 4.4 เปรียบเทียบค่าความแม่นตรงของโมเดล และ เวลาที่ใช้ในการสร้างโมเดล บนชุดข้อมูล SHUTTLE





(a) ค่าความแม่นตรงของโมเดลที่สร้างจากอัลกอริทึม naiveBayes และ J48





(b) เวลาที่ใช้สร้างโมเดลของอัลกอริทึม naiveBayes และ J48

รูปที่ 4.5 เปรียบเทียบค่าความแม่นตรงของโมเดล และ เวลาที่ใช้ในการสร้างโมเดล บนชุดข้อมูล SATELLITE IMAGE

4.2 ผลการสุ่มข้อมูลแบบก้าวหน้า

ตารางที่ 4.6 ถึง 4.10 ต่อไปนี้แสดงประสิทธิภาพของโมเดล หรือ Classifier ที่ได้จากการสังเคราะห์โมเดลด้วยอัลกอริทึม naive Bayes และ J48 ของข้อมูล Adult, Letter, Mushroom, Shuttle และ Satellite Image ตามลำดับ โดยการแสดงผลจะเปรียบเทียบค่าความแม่นตรงของการสุ่มแบบเลขคณิตเปรียบเทียบกับการสุ่มแบบเรขาคณิต ในส่วนของเวลาที่ใช้ในการสร้างโมเดลจะมีแนวโน้มเช่นเดียวกับการทดลองในส่วนของการสุ่มแบบพื้นฐาน นั่นคือเมื่อข้อมูลมีปริมาณมากขึ้นเวลาที่ใช้ในการสร้างโมเดลจะมากขึ้นด้วย การรายงานผลจึงเน้นเฉพาะค่าความแม่นตรง แนวโน้มการเพิ่มขึ้น (หรือลดลง) ของค่าความแม่นตรงเมื่อเพิ่มปริมาณการสุ่มข้อมูลของข้อมูล Adult, Letter, Mushroom, Shuttle และ Satellite Image แสดงเป็นภาพกราฟได้ดังรูปที่ 4.6 ถึง 4.10 ตามลำดับ

ตารางที่ 4.6 ประสิทธิภาพการสังเคราะห์โมเดลของข้อมูล Adult

การสุ่มข้อมูลแบบเลขคณิต (Arithmetic)			การสุ่มข้อมูลแบบเรขาคณิต (Geometric)		
จำนวนข้อมูล	ค่าความแม่นตรง (%)		จำนวนข้อมูล	ค่าความแม่นตรง (%)	
	naive Bayes	J48		naive Bayes	J48
100	84.20	81.86	100	84.20	81.86
300	83.76	84.31	200	83.994	80.49
500	83.67	84.09	400	83.570	80.49
700	83.81	84.28	800	84.104	84.26
900	83.93	84.26	1,600	83.650	84.55
1,100	83.72	83.58	3,200	83.736	84.15
1,300	83.58	84.97	6,400	83.459	85.00
1,500	83.59	84.63	12,800	83.207	85.71
1,700	83.47	84.48	25,600	83.189	85.82
1,900	83.50	83.76	32,561	83.128	85.85
2,100	83.59	84.05			
2,300	83.64	83.92			
2,500	83.51	84.20			
2,700	83.72	84.03			
2,900	83.79	84.03			
3,100	83.78	83.95			
3,300	83.74	84.69			
3,500	83.68	84.65			
32,561	83.13	85.85			

ตารางที่ 4.7 ประสิทธิภาพการสังเคราะห์โมเดลของข้อมูล Letter

การสุ่มข้อมูลแบบเลขคณิต (Arithmetic)		
จำนวนข้อมูล	ค่าความแม่นตรง (%)	
	naive Bayes	J48
100	39.16	31.12
300	52.46	50.42
500	57.96	60.48
700	58.04	60.68
900	59.58	60.98
1,100	59.88	62.08
1,300	59.88	62.08
1,500	59.78	66.44
1,700	60.34	68.18
1,900	60.18	69.98
2,100	61.12	71.5
2,300	62.04	71.62
2,500	62.66	71.88
2,700	62.62	71.98
2,900	62.90	72.70
3,100	62.80	73.50
3,300	63.04	74.92
3,500	62.78	75.72
15,000	63.56	87.70

การสุ่มข้อมูลแบบเรขาคณิต (Geometric)		
จำนวนข้อมูล	ค่าความแม่นตรง (%)	
	naive Bayes	J48
100	29.28	28.64
200	43.88	41.10
400	51.10	50.6
800	57.42	60.16
1,600	59.62	66.54
3,200	62.44	74.90
6,400	62.40	82.16
12,800	63.14	86.46
15,000	63.56	87.70

ตารางที่ 4.8 ประสิทธิภาพการสังเคราะห์โมเดลของข้อมูล Mushroom

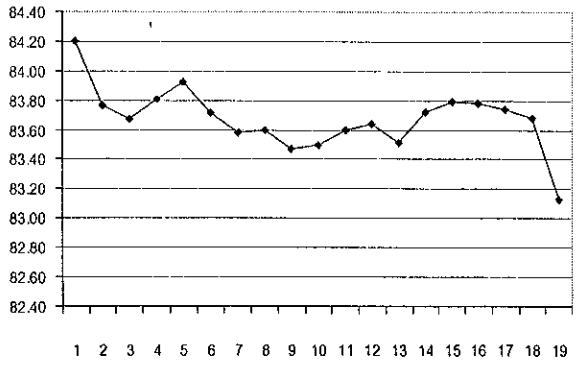
การสุ่มข้อมูลแบบเลขคณิต (Arithmetic)			การสุ่มข้อมูลแบบเรขาคณิต (Geometric)		
จำนวนข้อมูล	ค่าความแม่นตรง (%)		จำนวนข้อมูล	ค่าความแม่นตรง (%)	
	naive Bayes	J48		naive Bayes	J48
100	91.65	95.05	100	91.06	95.75
300	93.61	97.86	200	92.36	98.15
500	94.24	98.15	400	93.50	98.15
700	94.02	98.56	800	93.611	98.15
900	94.39	98.67	1,600	94.13	99.82
1,100	94.20	99.82	3,200	94.65	100
1,300	94.28	99.82	5,416	95.79	100
1,500	94.35	99.82			
1,700	94.31	99.82			
1,900	94.39	99.82			
2,100	94.57	99.82			
2,300	94.72	99.82			
2,500	94.68	99.82			
2,700	94.72	99.82			
2,900	94.94	99.82			
3,100	95.13	99.82			
3,300	95.20	99.82			
3,500	95.24	100			
5,416	95.79	100			

ตารางที่ 4.9 ประสิทธิภาพการสังเคราะห์โมเดลของข้อมูล Shuttle

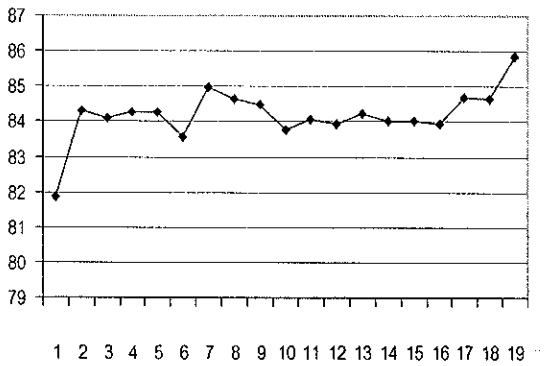
การสุ่มข้อมูลแบบเลขคณิต (Arithmetic)			การสุ่มข้อมูลแบบเรขาคณิต (Geometric)		
จำนวนข้อมูล	ค่าความแม่นตรง (%)		จำนวนข้อมูล	ค่าความแม่นตรง (%)	
	naive Bayes	J48		naive Bayes	J48
100	91.17	96.70	100	95.43	98.70
300	92.72	98.69	200	94.88	99.59
500	92.72	99.47	400	94.24	99.00
700	94.08	99.53	800	94.23	99.53
900	94.52	99.52	1,600	92.55	99.46
1,100	94.60	99.54	3,200	92.06	99.72
1,300	90.88	99.46	6,400	90.73	99.73
1,500	90.66	99.52	12,800	91.78	99.84
1,700	90.16	99.52	25,600	90.26	99.96
1,900	89.33	99.50	43,500	92.02	99.95
2,100	86.64	99.59			
2,300	87.28	99.63			
2,500	88.00	99.64			
2,700	88.61	99.63			
2,900	89.01	99.63			
3,100	89.60	99.62			
3,300	89.70	99.70			
3,500	89.61	99.67			
43,500	92.02	99.95			

ตารางที่ 4.10 ประสิทธิภาพการสังเคราะห์โมเดลของข้อมูล Satellite Image

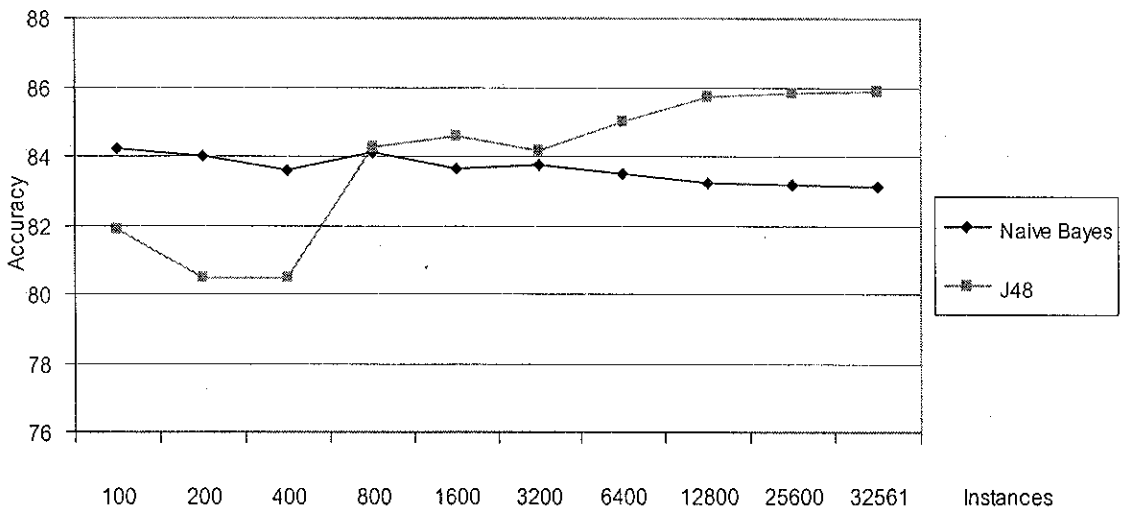
การสุ่มข้อมูลแบบเลขคณิต (Arithmetic)			การสุ่มข้อมูลแบบเรขาคณิต (Geometric)		
จำนวนข้อมูล	ค่าความแม่นตรง (%)		จำนวนข้อมูล	ค่าความแม่นตรง (%)	
	naive Bayes	J48		naive Bayes	J48
100	77.80	78.25	100	80.40	72.85
300	79.25	78.25	200	79.15	77.20
500	79.20	79.20	400	79.40	78.25
700	79.15	81.35	800	79.90	81.30
900	79.20	81.40	1,600	79.70	83.6
1,100	79.05	83.55	3,200	79.60	86.05
1,300	79.05	82.55	4,435	79.60	85.35
1,500	79.15	82.05			
1,700	79.20	83.70			
1,900	79.45	83.95			
2,100	79.50	84.35			
2,300	79.40	82.90			
2,500	79.55	84.10			
2,700	79.50	83.75			
2,900	79.60	84.80			
3,100	79.50	84.10			
3,300	79.65	85.20			
3,500	79.50	85.80			
4,435	79.60	85.35			



(a) การรัน naive Bayes กับข้อมูลสุ่มแบบเลขคณิต

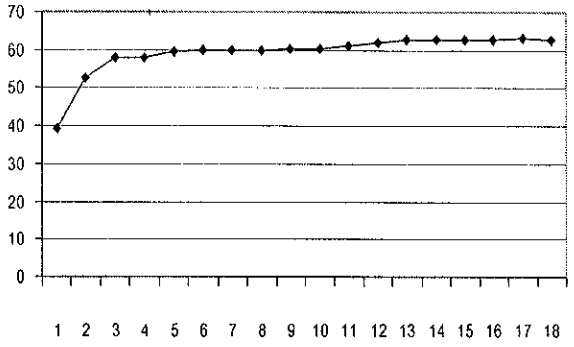


(b) การรัน J48 กับข้อมูลสุ่มแบบเลขคณิต

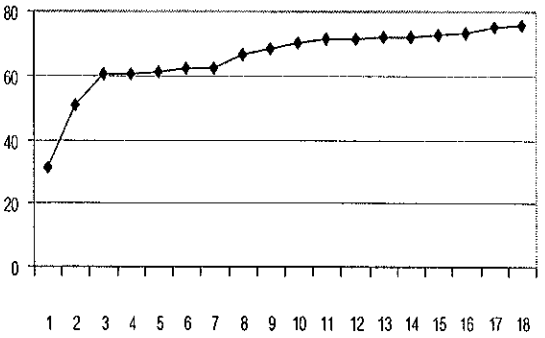


(c) การรัน naive Bayes และ J48 กับข้อมูลสุ่มแบบเรขาคณิต

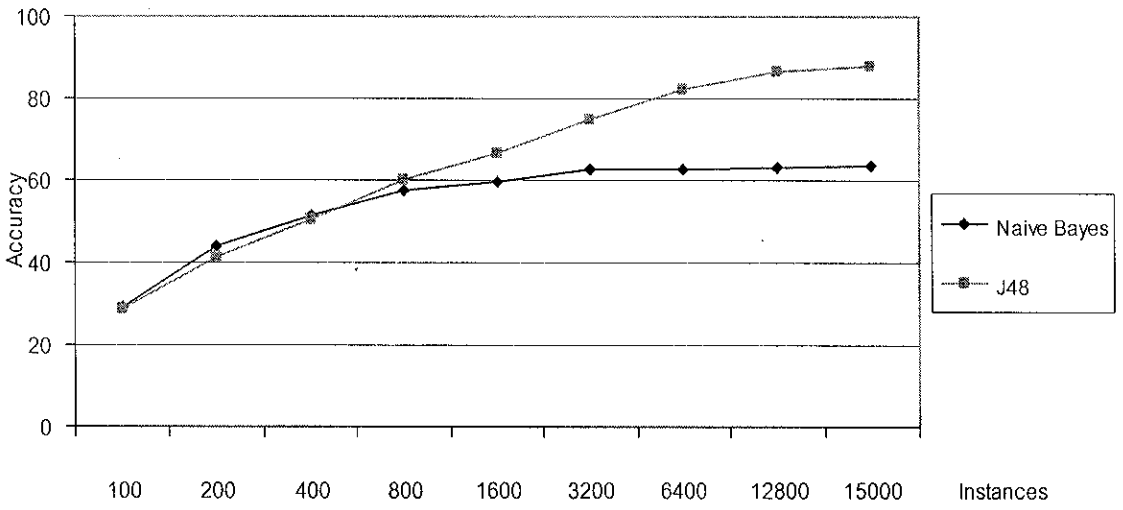
รูปที่ 4.6 เปรียบเทียบค่าความแม่นตรงของการสุ่มแบบเลขคณิตและแบบเรขาคณิตบนข้อมูล Adult



(a) การรัน naive Bayes กับข้อมูลสุ่มแบบเลขคณิต

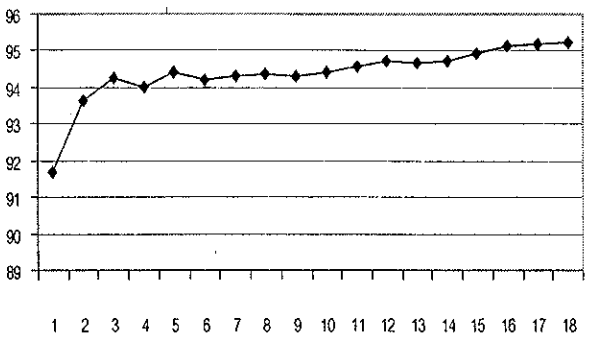


(b) การรัน J48 กับข้อมูลสุ่มแบบเลขคณิต

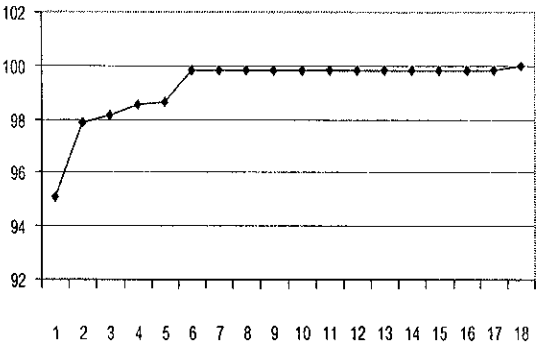


(c) การรัน naive Bayes และ J48 กับข้อมูลสุ่มแบบเรขาคณิต

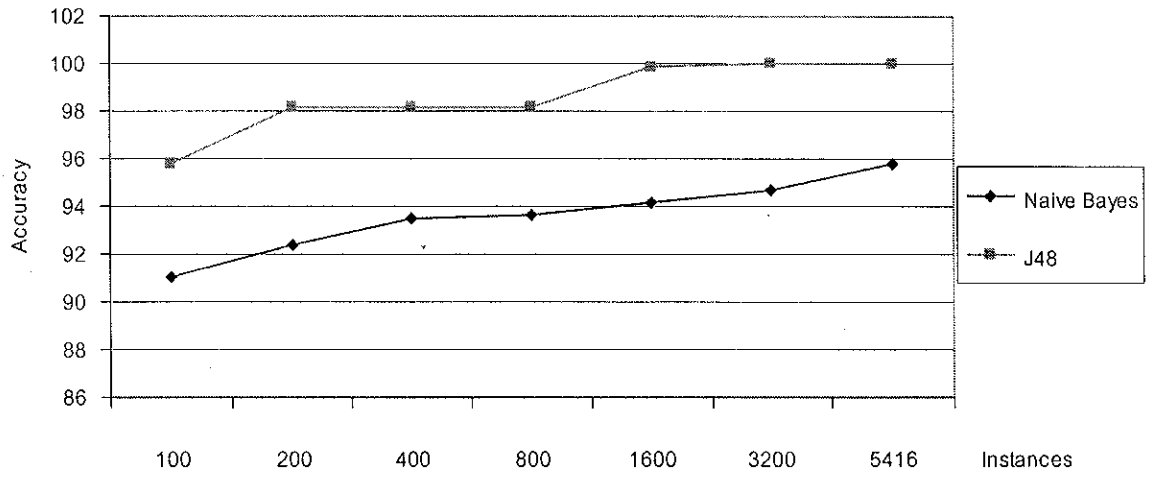
รูปที่ 4.7 เปรียบเทียบค่าความแม่นตรงของการสุ่มแบบเลขคณิตและแบบเรขาคณิตบนข้อมูล Letter



:(a) การรัน naive Bayes กับข้อมูลสุ่มแบบเลขคณิต

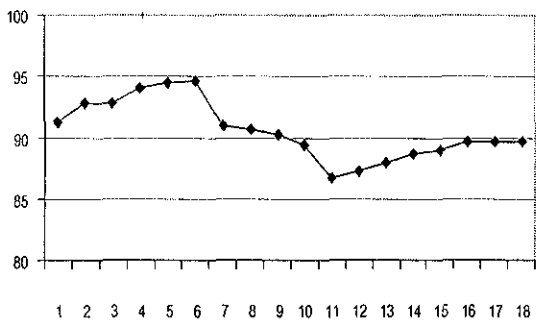


(b) การรัน J48 กับข้อมูลสุ่มแบบเลขคณิต

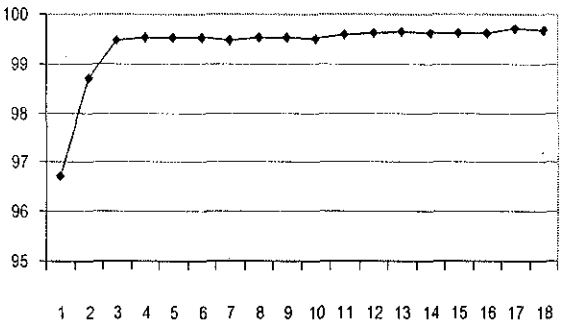


(c) การรัน naive Bayes และ J48 กับข้อมูลสุ่มแบบเรขาคณิต

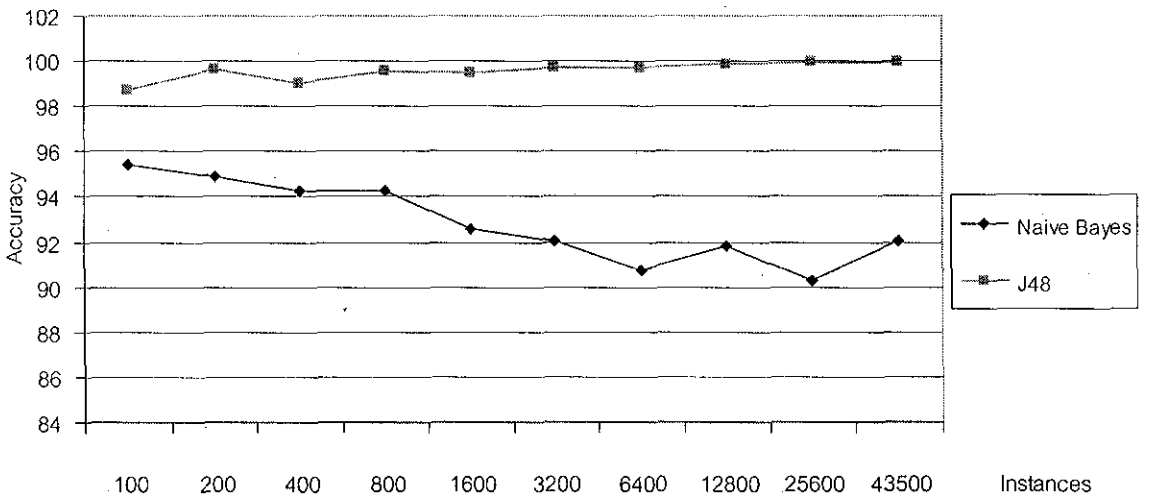
รูปที่ 4.8 เปรียบเทียบค่าความแม่นตรงของการสุ่มแบบเลขคณิตและแบบเรขาคณิตบนข้อมูล Mushroom



(a) การรัน naive Bayes กับข้อมูลสุ่มแบบเลขคณิต

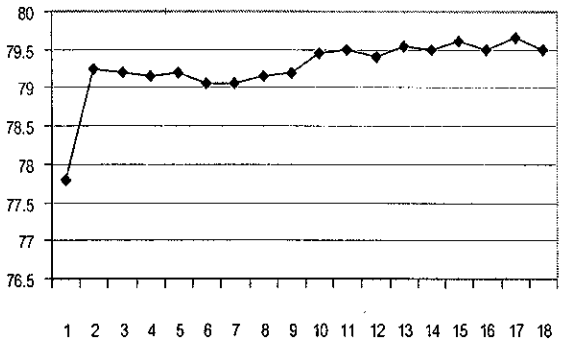


(b) การรัน J48 กับข้อมูลสุ่มแบบเลขคณิต

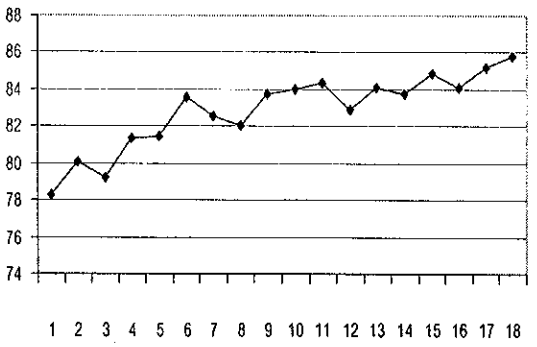


(c) การรัน naive Bayes และ J48 กับข้อมูลสุ่มแบบเรขาคณิต

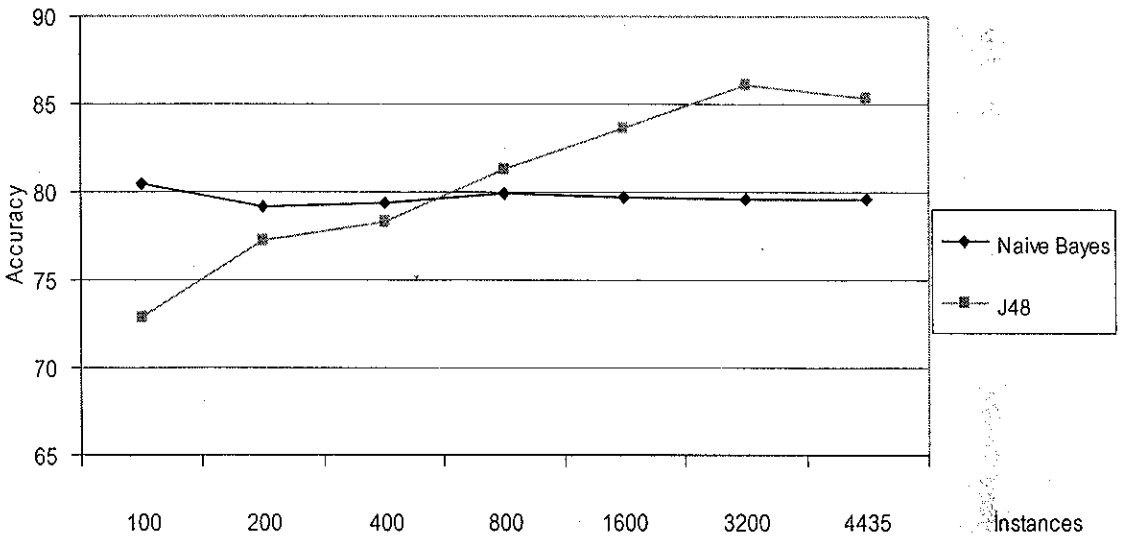
รูปที่ 4.9 เปรียบเทียบค่าความแม่นตรงของการสุ่มแบบเลขคณิตและแบบเรขาคณิตบนข้อมูล Shuttle



(a) การรัน naive Bayes กับข้อมูลสุ่มแบบเลขคณิต



(b) การรัน J48 กับข้อมูลสุ่มแบบเลขคณิต



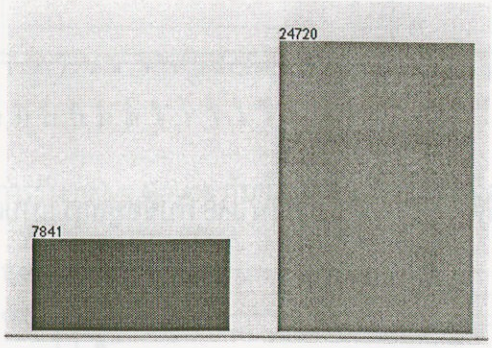
(c) การรัน naive Bayes และ J48 กับข้อมูลสุ่มแบบเรขาคณิต

รูปที่ 4.10 เปรียบเทียบค่าความแม่นตรงของการสุ่มแบบเลขคณิตและแบบเรขาคณิตบนข้อมูล Satellite Image

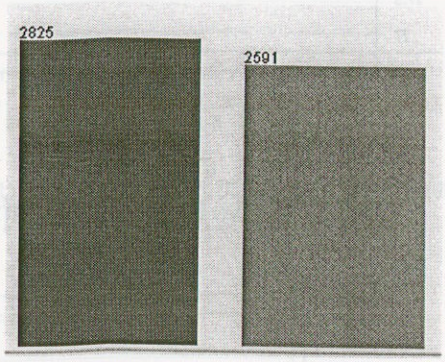
4.3 อภิปรายผล

4.3.1 การสุ่มข้อมูลแบบพื้นฐาน

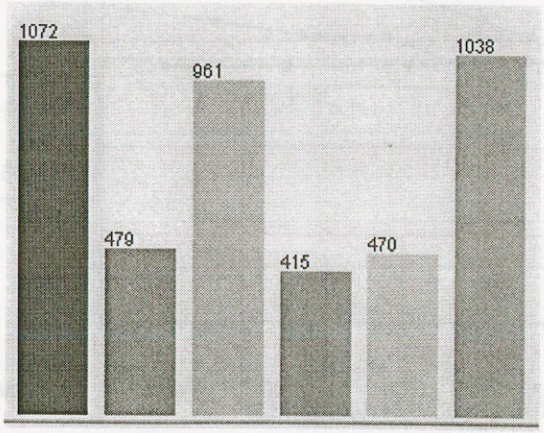
การทดสอบการสุ่มข้อมูลแบบพื้นฐาน 3 รูปแบบ คือ simple random, systematic random, stratified random กับข้อมูลทั้ง 5 ชุด ข้อมูลแต่ละชุดมีการกระจายของคลาสแสดงได้ดังรูปที่ 4.11 ให้ผลการทดสอบสรุปแยกเป็นสามประเด็นหลักได้ตามการเปรียบเทียบเทคนิคการสุ่ม การเปรียบเทียบเวลาที่ใช้สร้างโมเดล และเปรียบเทียบขนาดของกลุ่มตัวอย่าง



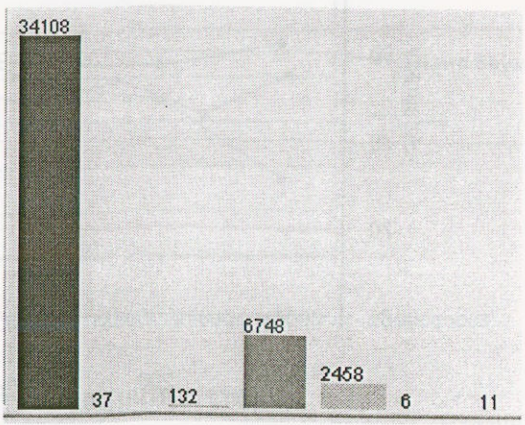
(a) Class distribution of data set Adult



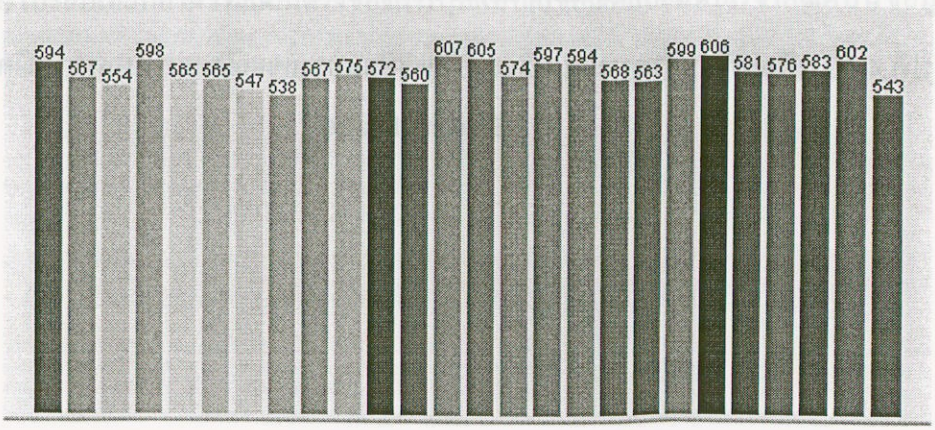
(b) Class distribution of data set Mushroom



(d) Class distribution of data set Satellite Image



(e) Class distribution of data set Shuttle



(c) Class distribution of data set Letter

รูปที่ 4.11 การกระจายของคลาสในแต่ละชุดข้อมูล

เปรียบเทียบเทคนิคการสุ่ม

- ในข้อมูล Adult ที่มีจำนวนคลาสของข้อมูล 2 คลาส (คือ คลาสของคนที่มีรายได้สูงกว่าห้าหมื่นดอลลาร์และคลาสของคนที่มีรายได้เท่ากับหรือต่ำกว่าห้าหมื่นดอลลาร์ โดยการกระจายของคลาสต่างกันมาก คลาสของคนรายได้สูงมีเพียง 24% แต่จำนวนคนที่รายได้ต่ำมีมากถึง 76%) และลักษณะของแอททริบิวต์มีปะปนกันทั้งที่เป็นข้อความ (nominal) และตัวเลข (numeric) การสุ่มแบบ systematic random จะให้ผลโดยเฉลี่ยดีกว่าการสุ่มแบบอื่น (พิจารณาจากค่าความแม่นตรง)
- ในข้อมูล Mushroom ที่มีจำนวนคลาสของข้อมูล 2 คลาสเช่นเดียวกัน (คือ คลาสของเห็ดที่รับประทานได้และคลาสของเห็ดมีพิษ การกระจายของข้อมูลทั้งสองคลาสใกล้เคียงกัน) แต่แอททริบิวต์เป็นข้อความทั้งหมด การสุ่มแบบ simple random ให้ผลที่ดีมากในกรณีจำแนกด้วยอัลกอริทึม naïve Bayes แต่ถ้าจำแนกข้อมูลด้วยอัลกอริทึม J48 การสุ่มแบบ stratified random จะให้ผลลัพธ์ที่ดีกว่า
- เมื่อข้อมูลมีจำนวนคลาสมากขึ้นเป็น 6 คลาส ในข้อมูล Satellite image (มีคลาสเป็นหมายเลข 1, 2, 3, 4, 5, 7, การกระจายของคลาสไม่สม่ำเสมอ) และข้อมูลเป็นตัวเลขทั้ง 36 แอททริบิวต์ การสุ่มข้อมูลด้วยเทคนิค systematic random และ stratified random ให้ผลลัพธ์ที่ดีกว่าการสุ่มแบบ simple random อย่างชัดเจน ในกรณีจำแนกคลาสของข้อมูลด้วยอัลกอริทึม J48 การสุ่มแบบ systematic random จะให้ผลลัพธ์ที่ดีกว่าการสุ่มแบบ stratified random เล็กน้อย
- ในข้อมูล Shuttle ที่เป็นค่าตัวเลขทั้ง 9 แอททริบิวต์ และคลาสของข้อมูลมีทั้งหมด 7 คลาส (คลาสหมายเลข 1 ถึง 7 และการกระจายของคลาสไม่สม่ำเสมอ) การสุ่มแบบ systematic random ให้ผลลัพธ์ที่ดีกว่าเมื่อจำแนกข้อมูลด้วยอัลกอริทึม naïve Bayes แต่เมื่อใช้อัลกอริทึม J48 การสุ่มแบบ stratified random จะให้ผลลัพธ์ที่ดีกว่า
- ในข้อมูล Letter ที่แอททริบิวต์เป็นจำนวนเลขทั้งหมด (จำนวน 16 แอททริบิวต์) และคลาสของข้อมูลมีจำนวนมากถึง 26 คลาส (ตามจำนวนตัวอักษร A ถึง Z) โดยการกระจายของคลาสสม่ำเสมอ การสุ่มข้อมูลทั้ง 3 แบบให้ผลลัพธ์ใกล้เคียงกัน

จากผลการเปรียบเทียบเทคนิคการสุ่มข้อมูลสรุปได้ว่า เทคนิคการสุ่มอย่างง่าย หรือ simple random sampling ใช้งานได้ดีเฉพาะเมื่อข้อมูลมีการกระจายสม่ำเสมอโดยจำนวนข้อมูลในแต่ละคลาสมีสัดส่วนที่ใกล้เคียงกัน ถ้าข้อมูลมีการกระจายไม่สม่ำเสมอโดยข้อมูลในบางคลาสมี

จำนวนมากกว่าข้อมูลในคลาสอื่น การสุ่มแบบเป็นระบบ (systematic random sampling) และการสุ่มแบบแบ่งชั้น (stratified random sampling) จะให้ผลลัพธ์ที่ดีกว่า

เปรียบเทียบเวลาที่ใช้

ความแตกต่างของเทคนิคการสุ่มไม่มีผลต่อเวลาที่ใช้ในการสร้างโมเดล เพราะแต่ละเทคนิคจะให้ปริมาณข้อมูลจากการสุ่มใกล้เคียงกัน เมื่อนำข้อมูลสุ่มไปรันด้วยโปรแกรมจำแนกข้อมูล (naïve Bayes และ J48) เวลาที่ใช้ในการสร้างโมเดลจึงใกล้เคียงกัน แต่สังเกตได้ว่าอัลกอริทึม J48 ใช้เวลาในการสร้างโมเดลนานกว่าอัลกอริทึม naïve Bayes แต่ทั้งนี้คุณภาพของโมเดลที่สร้างจาก J48 จะมีค่าความแม่นตรงสูงกว่าโมเดลที่สร้างจาก naïve Bayes

เปรียบเทียบขนาดของกลุ่มตัวอย่างที่เหมาะสม

การรันอัลกอริทึม naive Bayes และ J48 บนข้อมูลสุ่มจะใช้เวลาน้อยกว่าการรันบนชุดข้อมูลทั้งหมด ความแตกต่างของเวลาที่ใช้เห็นได้อย่างชัดเจน แต่การใช้ข้อมูลตัวอย่างที่ได้จากการสุ่มแทนที่จะใช้ข้อมูลทั้งหมดจะต้องคำนึงถึงคุณภาพกลุ่มตัวอย่างด้วย กลุ่มตัวอย่างที่ดีเหมาะสมต่อการนำไปใช้สร้างโมเดลเพื่อจำแนกคลาส จะต้องเป็นกลุ่มตัวอย่างที่ให้คุณภาพของโมเดลใกล้เคียงกับโมเดลที่สร้างจากชุดข้อมูลฉบับเต็มที่ไม่ได้ถูกลดขนาดด้วยการสุ่ม คุณภาพของโมเดลจะพิจารณาจากค่าความแม่นตรง (accuracy) ในการจำแนกคลาสของข้อมูล

ในการสร้างโมเดลเพื่อจำแนกคลาสด้วยการรันอัลกอริทึม naive Bayes พบว่าใน 3 ชุดข้อมูล คือ Adult, Shuttle, Satellite image กลุ่มตัวอย่างขนาดเพียง 2% ของข้อมูลฝึกทั้งหมดสามารถให้ผลลัพธ์เป็นโมเดลที่แม่นตรงเทียบเท่ากับโมเดลที่สร้างจากข้อมูลฝึกฉบับเต็มทั้งชุด ส่วนข้อมูล Mushroom ต้องใช้ขนาดกลุ่มตัวอย่างที่ 9% และข้อมูล Letter ต้องใช้ขนาดกลุ่มตัวอย่างที่ 20% จึงจะได้โมเดลที่แม่นตรงเทียบเท่ากับโมเดลที่สร้างจากข้อมูลฝึกฉบับเต็ม

การสร้างโมเดลเพื่อจำแนกคลาสของข้อมูลด้วยกลุ่มตัวอย่างแทนที่จะใช้ข้อมูลฝึกทั้งหมดที่มีข้อมูลปริมาณมาก จะช่วยให้สามารถลดเนื้อที่หน่วยความจำลงได้ในสัดส่วนตั้งแต่ 80% จนถึง 99% (ตามขนาดของกลุ่มตัวอย่าง) และสามารถลดเวลาที่ใช้ในการสร้างโมเดลได้มากกว่า 80%

4.3.2 การสุ่มข้อมูลแบบก้าวหน้า

การสุ่มแบบก้าวหน้า (progressive or dynamic sampling) เป็นเทคนิคการสุ่มที่พัฒนาเพิ่มเติมขึ้นจากการสุ่มแบบเป็นระบบ (systematic random sampling) โดยแทนที่จะกำหนดขนาดของกลุ่มตัวอย่างเป็นขนาดคงที่เช่น 8% ของข้อมูลฝึกทั้งหมด การสุ่มจะทำหลายครั้ง

แต่ละครั้งจะเพิ่มจำนวนข้อมูลขึ้นเป็นลำดับ เช่น การสุ่มครั้งแรกได้ข้อมูล 100 รายการ การสุ่มครั้งที่สองได้ข้อมูล 200 รายการ การสุ่มครั้งที่สามได้ข้อมูล 300 รายการ เป็นเช่นนี้ไปตามลำดับ โดยการเพิ่มของขนาดกลุ่มตัวอย่างอาจจะเพิ่มทีละน้อยเรียกว่าแบบเลขคณิตเช่น 100, 200, 300, 400, 500, ... หรืออาจจะเพิ่มแบบก้าวกระโดดเรียกว่าแบบเรขาคณิตเช่น 100, 200, 400, 800, ... การสุ่มทั้งสองแบบมีจุดมุ่งหมายที่จะทดสอบหาขนาดของกลุ่มตัวอย่างที่เหมาะสมโดยให้ผลลัพธ์เป็นโมเดลจำแนกคลาสของข้อมูลที่แม่นตรงใกล้เคียงกับโมเดลที่สังเคราะห์ขึ้นจากข้อมูลฝึกทั้งหมด

จากผลการทดสอบพบว่าเมื่อสังเคราะห์โมเดลด้วยอัลกอริทึม naive Bayes จำนวนข้อมูลที่ปริมาณ 300 รายการ จะให้ผลลัพธ์เป็นโมเดลที่คุณภาพใกล้เคียงกับโมเดลที่สังเคราะห์ขึ้นจากข้อมูลฝึกทั้งหมด ทั้งนี้ยกเว้นกรณีข้อมูล Letter ที่ต้องใช้ขนาดของข้อมูลตัวอย่างมากถึง 3,000 รายการจึงจะให้ผลลัพธ์ที่มีคุณภาพอยู่ในเกณฑ์ที่ยอมรับได้

การสังเคราะห์โมเดลด้วยอัลกอริทึม J48 ต้องใช้ขนาดของกลุ่มตัวอย่างมากถึง 3,000 รายการ จึงจะได้โมเดลที่มีคุณภาพใกล้เคียงกับโมเดลที่สังเคราะห์ขึ้นจากข้อมูลฝึกทั้งหมด ทั้งนี้ยกเว้นกรณีข้อมูล Shuttle ที่ให้คุณภาพโมเดลที่ดีที่ขนาดของกลุ่มตัวอย่างเพียง 300 รายการ

จากผลการเปรียบเทียบอัตราการเรียนรู้ (learning rate) ของอัลกอริทึม naive Bayes และ J48 พบว่า naive Bayes มีอัตราการเรียนรู้ที่เร็วมาก การใช้ข้อมูลตัวอย่างเพียงเล็กน้อยที่ประมาณ 300 รายการสามารถให้ผลลัพธ์ที่มีคุณภาพใกล้เคียงกับการเรียนรู้จากข้อมูลขนาดใหญ่ อัลกอริทึม J48 มีอัตราการเรียนรู้ที่ช้ากว่ามาก ต้องใช้ข้อมูลตัวอย่างมากถึง 3,000 รายการจึงจะเริ่มได้ผลลัพธ์ที่ดี และในข้อมูลบางชุด เช่นข้อมูล Adult และ Letter ต้องใช้ข้อมูลมากถึง 13,000 รายการจึงจะให้ผลลัพธ์ที่ดี แต่จากการเปรียบเทียบด้วยข้อมูลทั้ง 5 ชุด อัลกอริทึม J48 ให้ผลลัพธ์เป็นโมเดลจำแนกข้อมูลที่มีความแม่นตรงสูงกว่าโมเดลจำแนกข้อมูลที่สร้างขึ้นโดยอัลกอริทึม naive Bayes

บทที่ 5
บทสรุป

การทำเหมืองข้อมูลเป็นกระบวนการวิเคราะห์ข้อมูลอัตโนมัติ โดยเน้นการทำงานกับข้อมูลขนาดใหญ่ เช่น ข้อมูลภาพถ่ายดาวเทียม ข้อมูลสำมะโนประชากร การวิเคราะห์มิได้เน้นผลความเที่ยงตรงแต่ประการเดียวเหมือนการวิเคราะห์ในเชิงสถิติ แต่เน้นการค้นหาโมเดลที่สามารถอธิบายข้อมูล หรือ ค้นหาแพทเทิร์นที่ไม่ได้คาดหมายล่วงหน้าที่แฝงอยู่ในข้อมูลขนาดใหญ่เหล่านั้น ซึ่งการค้นหารูปแบบความสัมพันธ์ในลักษณะนี้จะต้องใช้เวลาในการค้นหานานมาก วิธีการปรับปรุงการค้นหาโมเดลหรือความสัมพันธ์ที่ได้ผลดีคือ การลดขนาดของข้อมูลลงด้วยวิธีการสุ่ม ข้อมูลที่ได้จะเป็นข้อมูลกลุ่มตัวอย่างหรือตัวแทนของข้อมูลทั้งหมด การวิจัยนี้มีจุดมุ่งหมายที่จะศึกษารูปแบบการลดขนาดข้อมูลด้วยการสุ่มเพื่อใช้ในงานทำเหมืองข้อมูล การศึกษาวิจัยแบ่งเป็นสองส่วน ส่วนแรกศึกษาพฤติกรรมการสุ่มพื้นฐาน 3 เทคนิคคือ การสุ่มอย่างง่าย การสุ่มแบบเป็นระบบ และการสุ่มแบบแบ่งชั้น เพื่อพิจารณาว่าในข้อมูลขนาดใหญ่การสุ่มแบบใดจะให้ผลลัพธ์ที่ดี การศึกษาในส่วนที่สองเป็นการทดลองวิธีการสุ่มแบบก้าวหน้า ที่ใช้การเพิ่มขนาดกลุ่มตัวอย่างไปตามลำดับเพื่อพิจารณาว่ากลุ่มตัวอย่างขนาดเท่าใด ที่จะให้ผลลัพธ์ที่มีคุณภาพใกล้เคียงกับผลลัพธ์ที่ได้จากการทำงานกับข้อมูลทั้งหมด คุณภาพของผลลัพธ์จะพิจารณาจากค่าความแม่นตรงของโมเดลที่สังเคราะห์ขึ้นจากข้อมูล ค่าความแม่นตรงนี้ทดสอบด้วยการสร้างชุดข้อมูลทดสอบแยกต่างหากจากชุดข้อมูลฝึก การสังเคราะห์โมเดลใช้อัลกอริทึม naive Bayes และ J48 (decision-tree induction)

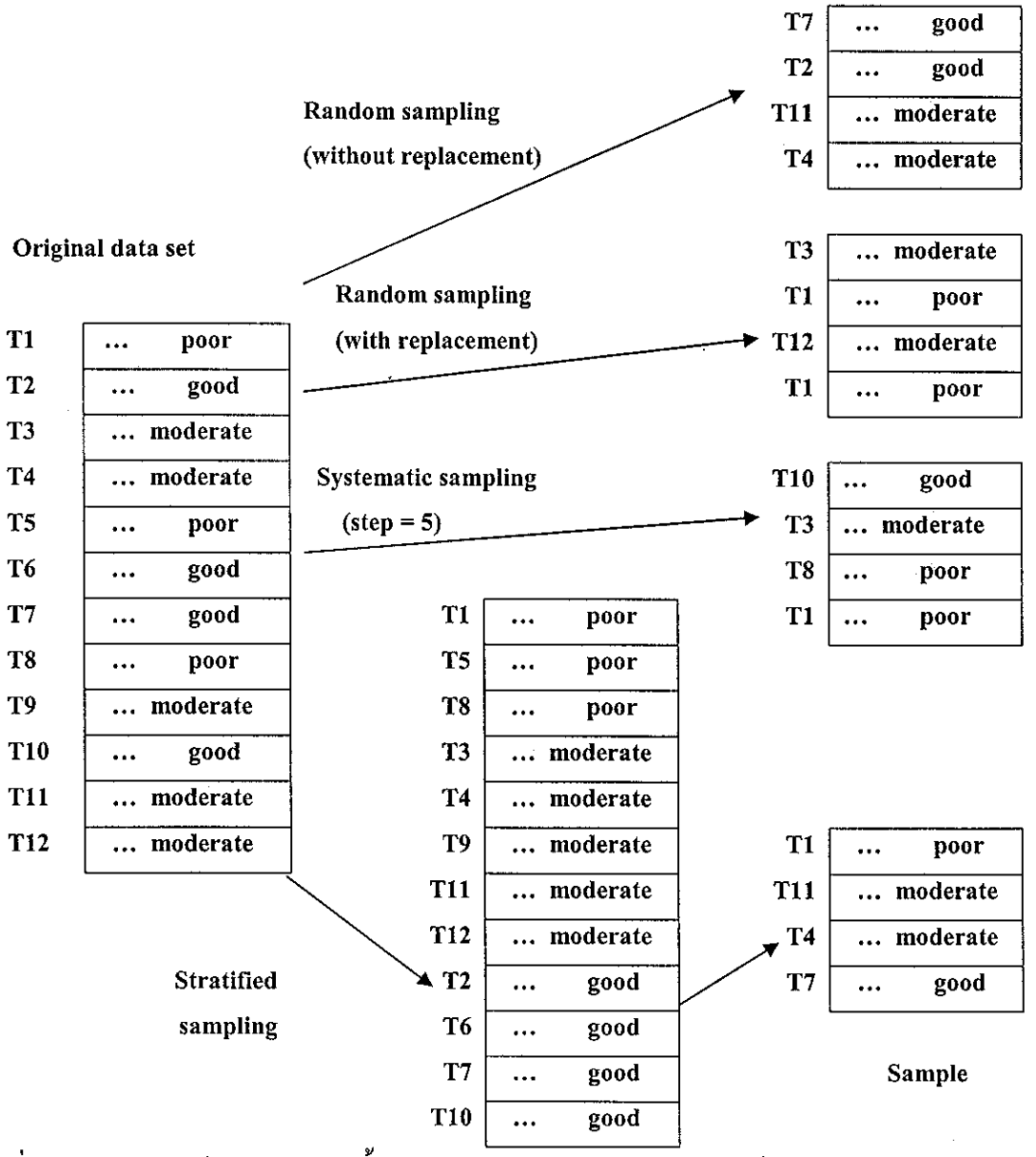
5.1 สรุปผลการวิจัย

ผลการทดลองกับข้อมูลสุ่มทั้งในส่วนแรก ได้แก่การศึกษาพฤติกรรมของเทคนิคการสุ่มแบบพื้นฐาน และส่วนที่สองได้แก่การสุ่มแบบก้าวหน้าที่ใช้การเพิ่มขนาดกลุ่มตัวอย่างไปตามลำดับ สรุปในประเด็นที่สำคัญได้ดังนี้

(1) การสุ่มอย่างง่าย (simple random sampling) จะให้ผลลัพธ์ที่ดีเฉพาะในกรณีที่ข้อมูลมีการกระจายของข้อมูลในแต่ละคลาสด้วยสัดส่วนที่ใกล้เคียงกัน โดยไม่ขึ้นกับจำนวนคลาส ข้อมูลอาจจะมีเพียง 2 คลาส เช่น ข้อมูล Mushroom หรือข้อมูลมีจำนวนคลาสมากถึง 26 คลาส เช่น ข้อมูล Letter

(2) เมื่อข้อมูลมีการกระจายในแต่ละคลาสไม่เท่ากัน เช่น ในข้อมูล Adult, Shuttle, Satellite image การสุ่มแบบเป็นระบบ (systematic random sampling) และการสุ่มแบบแบ่งชั้น (stratified random sampling) จะให้ผลลัพธ์ที่ดีกว่าการสุ่มอย่างง่าย

(3) การสุ่มแบบเป็นระบบ และการสุ่มแบบแบ่งชั้นให้ผลลัพธ์ที่มีคุณภาพไม่แตกต่างกันมากนัก แต่ขั้นตอนในการสุ่มข้อมูลแบบแบ่งชั้นใช้เวลามากกว่า (วิธีการสุ่มแสดงเป็นแผนภาพได้ดังรูปที่ 5.1) เพราะเบื้องต้นต้องแยกข้อมูลแต่ละคลาสออกจากกัน จากนั้นนับจำนวนข้อมูลในแต่ละคลาสเพื่อคำนวณสัดส่วนการสุ่ม ให้ได้กลุ่มตัวอย่างที่มีสัดส่วนของข้อมูลในแต่ละคลาสคงเดิม ดังนั้นในกรณีมีข้อมูลจำนวนมาก การสุ่มแบบเป็นระบบจะช่วยให้ได้ข้อมูลตัวอย่างในเวลาที่รวดเร็วกว่าการสุ่มแบบแบ่งชั้น



รูปที่ 5.1 วิธีการสุ่มข้อมูลแบบแบ่งชั้นเปรียบเทียบกับการสุ่มข้อมูลแบบเป็นระบบและการสุ่มอย่างง่าย

(4) ในกรณีที่ข้อมูลไม่คงที่ มีจำนวนข้อมูลเพิ่มเติมสะสมขึ้นได้ตลอดเวลา เช่นการรับข้อมูลผ่านระบบเครือข่าย หรือ stream data การสังเคราะห์โมเดลจากข้อมูลตัวอย่างสามารถทำได้เช่นเดียวกัน จากการทดสอบการสุ่มแบบก้าวหน้าพบว่าข้อมูลประมาณ 13,000 รายการ ให้ผลการสังเคราะห์โมเดลดีเทียบเท่ากับการสังเคราะห์โมเดลจากชุดข้อมูลฝึกฉบับเต็ม ที่มีข้อมูลมากถึง 43,500 รายการ

(5) จากการทดสอบการสุ่มแบบก้าวหน้าในข้อมูลทั้ง 5 ชุด และทดสอบสังเคราะห์โมเดลด้วยอัลกอริทึม naive Bayes และ J48 พบว่าเมื่อข้อมูลเพิ่มจำนวนขึ้นในช่วง 1,500 ถึง 2,000 รายการ ค่าความแม่นตรงของโมเดลมีค่าค่อนข้างสูง (แต่ยังไม่ใช่ค่าที่สูงที่สุด) การเพิ่มข้อมูลมากขึ้นกว่านี้ จะทำให้ค่าความแม่นตรงของโมเดลเพิ่มขึ้น แต่ด้วยอัตราการเพิ่มที่ไม่มากนัก ทำให้สรุปได้ว่า การสร้างโมเดลจำแนกข้อมูลกรณีที่มีข้อมูลจำนวนมาก และต้องการได้ผลลัพธ์เบื้องต้นมาใช้ในการประเมินลักษณะข้อมูล สามารถใช้ผลการสังเคราะห์โมเดลที่ข้อมูลจำนวน 1,500 ถึง 2,000 รายการ

5.2 ข้อเสนอแนะ

จากผลการทดสอบทำให้ได้แนวทางการลดขนาดข้อมูลในงานทำเหมืองข้อมูลว่า การลดขนาดข้อมูลด้วยวิธีการสุ่มสามารถใช้ได้ผลดี และเทคนิคการสุ่มที่เหมาะสมคือการสุ่มแบบเป็นระบบ ในกรณีที่ข้อมูลเพิ่มขึ้นอย่างต่อเนื่องการสุ่มแบบเป็นระบบที่ปริมาณข้อมูลประมาณ 13,000 รายการ สามารถให้ผลลัพธ์เป็นโมเดลจำแนกข้อมูลที่มีค่าความแม่นตรงสูง และถ้าต้องการได้โมเดลเบื้องต้นเพื่อศึกษาลักษณะข้อมูล การสุ่มแบบเป็นระบบให้ได้ข้อมูลประมาณ 2,000 รายการจะให้ผลลัพธ์ที่ค่อนข้างดีได้ในเวลาที่รวดเร็ว

การทดสอบนี้กระทำกับข้อมูล 5 ชุดที่มีจำนวนข้อมูล 4,000 รายการ จนถึงประมาณ 43,000 รายการ มีขนาดตั้งแต่ 500 KB จนถึง 4 GB ในการทำเหมืองข้อมูลกับข้อมูลจริงที่มีขนาดใหญ่มาก เช่น ข้อมูลภาพถ่ายดาวเทียมที่รวบรวมไว้ตลอดปี เกณฑ์ในการพิจารณาขนาดของข้อมูลสุ่มอาจจะต่างไปจากเกณฑ์ที่เสนอไว้ในรายงานฉบับนี้ได้บ้าง และเทคนิคการสุ่มแบบอื่น เช่น การสุ่มซ้ำ (resampling) อาจจะได้ผลลัพธ์ที่แตกต่างกว่านี้ ประเด็นเหล่านี้เป็นสิ่งที่ผู้วิจัยสนใจที่จะศึกษาค้นคว้าเพิ่มเติมต่อไป

บรรณานุกรม

รัชนี กัลยาวินัย, อัจฉรา ธารอุไรกุล . *Introduction to System Analysis and Design.* กรุงเทพฯ : บริษัทการศึกษาจำกัด.

ศิริลักษณ์ สุวรรณวงศ์. (2538). *ทฤษฎีและเทคนิคการสุ่มตัวอย่าง*. กรุงเทพฯ : โอเดียนสโตร์.

D. Barbara, W. DuMouchel, C. Faloutsos, P.J. Haas, J.H. Hellerstein, Y. Ioannidis, H.V. Jagadish, T. Johnson, R. Ng., V. Poosala, K.A. Ross, & K.C. Servcik. (1997). The New Jersey data reduction report. *Bulletin of the Technical Committee on Data Engineering, 20*, 3-45.

W. G. Catlett. (1992). Peepholing: Choosing Attributes Efficiently for Megainduction. In *Machine Learning: Proceedings of the Ninth International Workshop*, 49-54.

J. Cochrane. (1977). *Sampling Techniques.* John Wiley & Sons.

M. Dash & H. Liu. (1997). Feature selection methods for classification. *Intelligent Data Analysis: An International Journal*, 1.

M. Dash, H. Liu, & J. Yao. (1997). Dimensionality reduction of unsupervised data. In *Proceedings of 9th IEEE International Conference on Tools with AI (ICTAI)*, 532-539.

J. Devore & R. Peck. (1997). *Statistics: The Exploration and Analysis of Data.* New York: Duxbury Press.

G.H. John & P. Langley. (1996). Static versus dynamic sampling for data mining. In *Proceedings 1996 International Conference on Knowledge Discovery and Data Mining (KDD' 96)*, 367-370.

K. Josien, G. Wang, T.W. Liao, E. Triantaphyllou, & M.C. Liu. (2001). An evaluation of sampling methods for data mining with fuzzy c-means. In *Data Mining for Design and Manufacturing*, Chapter 15, 351-365. Kluwer Academic Publishers.

J. Kivinen & H. Mannila. (1994). The power of sampling in knowledge discovery. In *Proceedings 13th ACM Symposium on Principles of Database Systems*, 77-85.

R. Kohavi & G.H. John. (1997). Wrappers for feature subset selection. *Artificial Intelligence,* 97, 273-324.

H. Liu & H. Motoda, editors. (1998). *Feature Extraction, Construction, and Selection: A Data Mining Perspective.* Boston: Kluwer Academis Publishers.

C.J. Merz & P.M. Murphy. (1997). *UCI Repository of Machine Learning Databases.* http://www.ics.uci.edu/~mlearn/MLRepository.

J. Neter, M.H. Kutner, C.J. Nachtsheim, & L. Wasserman. (1996). *Applied Linear Statistical Models*, 4th ed. Chicago:Irwin.

ภาคผนวก

บทความวิจัยนำเสนอในการประชุมวิชาการ

- K. Kerdprasop, N. kerdprasop, P. Punpakdeewong, and P. Doungsuwan (2002). The effect of sampling techniques to accuracy estimation. *การประชุมวิชาการ วิศวกรรมศาสตร์เพื่อโลกน่าอยู่*, มหาวิทยาลัยสงขลานครินทร์, Thailand, 4-5 กรกฎาคม.

- K. Kerdprasop, N. Kerdprasop, C. Kongcharoen and T. Tippayasot (2004). Efficient progressive sampling for data mining. *Proceedings of 1st KMITL International Conference on Integration of Science and Technology for Sustainable Development*, Bangkok, Thailand, August 25-26, pp.313-316.

The effect of sampling techniques to accuracy estimation*

Kittisak Kerdprasop, Nittaya Kerdprasop, Pongden Punpakdeewong, and Petchpirin Doungsuwan

School of Computer Engineering
Suranaree University of Technology
111 Muang District
Nakorn Ratchasima 30000
Phone (044) 224352; Fax (044) 224165

ABSTRACT

Knowledge discovery is the process of extracting useful and previously unknown information from the very large data set. Among many discovering methods, decision rules extracting is one of the most extensively studied techniques. But extracting rules from a large database is computationally inefficient. Using a sample from the database can speed up the data mining process, but this is only acceptable if it does not reduce the quality of the induced rules. We thus investigate the criteria to decide whether a sample is sufficiently similar to the original database. We observe the accuracy of the induced rules extracted from training samples of decreasing sizes and use these results to determine when a sample is sufficiently small, yet maintain the acceptable accuracy rate. We evaluate random and systematic sampling methods on data from the UCI repository.

1. Introduction

Data mining (also known as knowledge discovery in databases, or KDD) is the process of applying specific learning algorithm to extract interesting and useful knowledge from data [2]. Typical data mining applications extract knowledge from databases ranging from small to moderate in size. When a data set is very large, mining process may take a very long time. Moreover, some mining algorithms may not be scalable on huge amounts of data. To handle large data sets, data reduction is one important step prior to applying the mining algorithms.

Data reduction can be achieved by reducing the number of cases and/or reducing dimensions of those cases. Our study focuses on case (or instance) reduction via the technique of sampling. Mining on reduced data set is obviously more efficient than on the original data set. On the contrary, if the sample is too small, some useful knowledge may be overlooked. Our paper addresses the question of sufficient sample size as well as the improved mining time. The rest of the paper is organized as follows. The next section describes various sampling methods. Section 3

* This research has been supported by the grant from Suranaree University of Technology.

explains the methodology of our study including experimental setup and the data sets. Section 4 discusses the results. The conclusion is presented in Section 5.

2. Sampling methods

Sampling is used as a data reduction technique because it allows a large data set to be represented by a much smaller subset of the data. Basic methods of sampling commonly used are random sampling, systematic sampling, and stratified sampling [4].

Suppose that a large data set contains N instances. Random sampling selects n instances ($n < N$) at a random choice. The probability of drawing any instance in the data set is $1/N$, that is, all instances are equally likely. This is the case of random sampling without replacement. If the sampling is done with replacement, an instance has a chance to be drawn more than one times.

The systematic sampling method draws n instances from the data set by their fixed stepping positions. This sampling method draws the first instance at a random position. Then iteratively draws subsequent instance at the next k position, when k is a stepping size.

Stratified sampling method first divides the data set into mutually disjoint subsets called strata. Then draws samples from each stratum independently by applying the simple random sampling technique. The three sampling methods are in illustrated in Figure 2.1.

3. Methodology

3.1 Experimental Setup

In order to conduct an experiment to investigate the sufficient size of a sample obtained from different sampling methods, we use OneR as a learning algorithm to induce decision rules. OneR algorithm [3] induces decision rules based on the value of a single attribute. It is shown that we can get reasonably accurate decision rules by simply looking at one attribute, as opposed to a more sophisticated top-down decision-tree induction algorithms such as C4.5 [6]. The average accuracy of OneR for the data sets tested by Holte [3] is just 5.7% lower than that of C4.5.

We choose two data sets from the UCI Repository [1]. The two data sets represent the variety of data characteristics, that is, numeric data, nominal data, and data with missing values. Each data set is sampled using two different sampling methods: random sampling (without replacement) and systematic sampling.

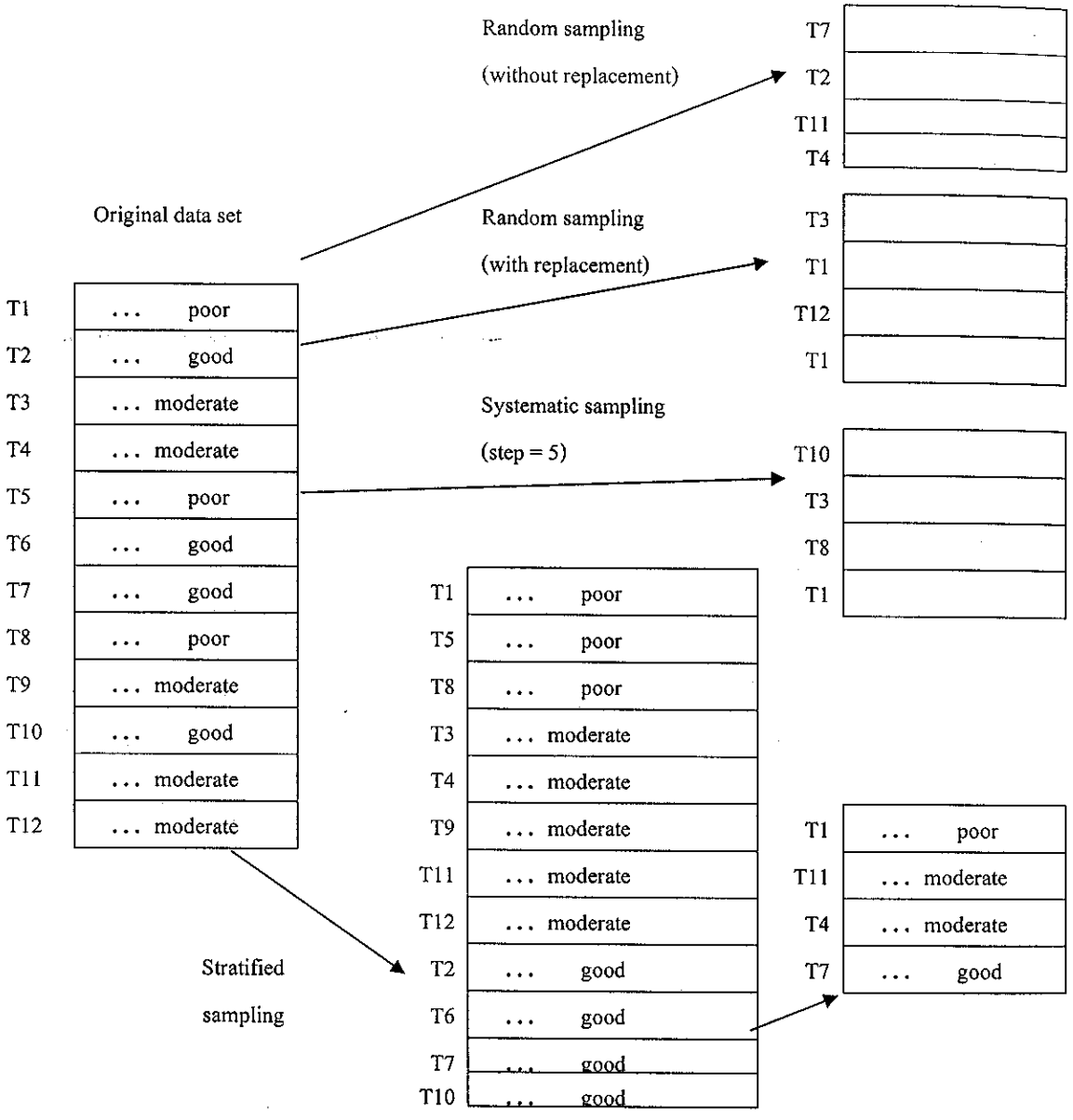


Figure 2.1 Different sampling methods to draw 4 samples

For each sampling method, a data set is drawn for five different sample sizes: 80%, 70%, 60%, 50%, and 40% of the original data set. Then runs the learning algorithm on each sample. The learning algorithm is also run on the original data set to observe the accuracy and the learning time. These two criteria will be used as a benchmark to compare against those obtained from the various samples.

The experiments are performed on the WEKA (Waikato Environment for Knowledge Analysis) system [7]. WEKA system is an open-source Java-based

machine learning environment that provides tools and algorithms to be used as a data- mining workbench.

3.2 Methods for Accuracy Estimation

Accuracy estimation refers to the process of approximating the future performance of a set of decision rules induced by a learning algorithm. This process helps to evaluate how accurately the induced rules will predict on the future data. The common accuracy estimation methods used extensively are [5] holdout, cross-validation, leave-one-out cross-validation, stratified cross-validation, and bootstrap. In our experiments, we estimate the accuracy of the induced rules using the holdout and stratified 10-fold cross-validation methods.

The holdout method partitions the original data set into two mutually disjoint sets: a training set and a test set (or holdout set). Typically, two thirds of the data ($2/3 \approx 66.6\%$) are allocated to the training set, and the remaining ($1/3 \approx 33.3\%$) is allocated to the test set. The training set is used to train the learning algorithm, and the induced decision rules are tested on the test set. Since only 33.3% of the original data are used for estimating accuracy, this method is not a good estimator for a small data set.

Ten-fold cross-validation method is a variation of the holdout method in which the method is repeated 10 times. The data set is randomly split into ten mutually disjoint subsets, called the folds. The ten folds are approximately equal in size. To induce decision rules, nine folds are used to train the learning algorithm, the remaining one fold will be used as a test set. The process is repeated 10 times with a different set of test data at each iteration. The overall accuracy estimation is the average of the accuracy obtained from each iteration.

Stratified 10-fold cross-validation is a cross-validation technique in which the original data is stratified before it is partitioned into ten folds. This is to guarantee the equal distribution of data in each class. This method is a good estimator for a small data set [5].

In order to observe the predicting accuracy of different sampling sizes, we employ both the holdout and the stratified 10-fold cross-validation methods since we range the sampling size from 100% (i.e., no sampling at all) to as small as 40% of the original data size.

4. Results and discussion

For each data set, we first apply sampling methods to generate the samples of different sizes. Each sample is then applied to the OneR learning algorithm to induce the decision rules. The predicting accuracy of these rules is tested with the two estimating methods: holdout and stratified 10-fold cross-validation. Tables 4.1 and 4.2 show the results of the accuracy estimation for the data sets using different sampling methods at various sizes. The columns 'Correct Rules' indicate the sampling sizes that generate the same set of rules as the original data (i.e., no sampling data set).

Table 4.1 Accuracy estimates for the Iris data set (numeric data)

Sampling Method	Sampling Size	Number of Instances	Accuracy (Holdout)	Correct Rules	Accuracy (stratified 10-fold CV)
No sampling	100%	150	98.0392%		92.6667%
Random Sampling	80%	120	87.8049%		92.5%
	70%	106	89.1892%	3	94.3396%
	60%	90	93.5484%		96.6667%
	50%	75	92.3077%	3	93.3333%
	40%	60	100%		100%
Systematic Sampling	80%	120	92.6829%		92.5%
	70%	105	97.2222%	3	98.0952%
	60%	90	90.3226%		96.6667%
	50%	74	100%	3	97.2973%
	40%	60	100%	3	96.6667%

Table 4.2 Accuracy estimates for the Primary-Tumor data set (nominal data with missing values)

Sampling Method	Sampling Size	Number of Instances	Accuracy (Holdout)	Correct Rules	Accuracy (stratified 10-fold CV)
No sampling	100%	339	28.4483%		27.4336%
Random Sampling	80%	264	26.6667%	3	28.0303%
	70%	231	26.5823%		24.6753%
	60%	199	32.3529%	3	30.6533%
	50%	165	28.0702%		30.303%
	40%	132	35.5556%		32.5758%
Systematic Sampling	80%	264	26.6667%	3	30.6818%
	70%	231	22.7848%	3	29.4372%
	60%	198	27.9412%	3	28.7879%
	50%	165	31.5789%		30.303%
	40%	132	31.1111%		26.5152%

For each data set varying in the characteristics, the results indicate that:

(1) For the small data set (Iris data), stratified 10-fold cross-validation estimating method on sampled data predicts with a higher accuracy than predicting with the original data.

(2) Sampling sizes of 50-70% produce the same set of decision rules as the original data, but the accuracy rate is higher.

(3) For a larger data set (Primary-Tumor data) with missing values, sampling sizes of 60-80% generate the same results as the original data, but with a

higher accuracy.

(4) On average, random sampling generates slightly more accurate samples than the systematic sampling method on a larger data set.

(5) The time to induce the decision rules is 0 second on every sample. So, we cannot conclude that the smaller samples help decreasing the learning time.

5. Summary

We review common sampling methods that are naturally used as a technique to reduce the data size for the purpose of improving learning time in the data mining process. We design the experiments to vary the sampling sizes in order to observe the smallest sampling size that yields the learning result as accurate as the original data. The accuracy is estimated with two different methods: the holdout and the stratified 10-fold cross-validation.

Our results indicate that random sampling of the size approximately 60-80% of the original data produces the same result as the original data with a high accuracy. However, to draw the exact conclusion requires a further investigation on a larger data set as well as with two or more learning algorithms.

References

[1] C. L. Blake and C. J. Merz: "UCI Repository of machine learning databases," University of California, Irvine, Department of Information and Computer Science, 1998. [http:// www.ics.uci.edu/~mlearn/MLRepository.html].

[2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth: "From data mining to knowledge discovery in databases," AI Magazine, pp.37-54, 1996.

[3] R.C. Holt: "Very simple classification rules perform well on most commonly used datasets," Machine Learning, Vol. 11, pp.63-90, 1993.

[4] K. Josien, G. Wang, T.W. Liao, E. Triantaphyllou, and M.C. Liu: "An evaluation of sampling methods for data mining with fuzzy c-means," In Dan Braha, editor, Data Mining for Design and Manufacturing, Chapter 15, pp.351-365, Kluwer Academic, 2001.

[5] R. Kohavi: "A study of cross-validation and bootstrap for accuracy estimation and model selection," Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), pp.1137-1143, 1995.

[6] J. R. Quinlan: "C4.5: Programs for Machine Learning," Morgan Kaufmann, 1993.

[7] WEKA (Waikato Environment for Knowledge Analysis), University of Waikato, Department of Computer Science, New Zealand. [http://www.cs. waikato.ac.nz/~ml].

EFFICIENT PROGRESSIVE SAMPLING FOR DATA MINING

Kittisak Kerdprasop, Nittaya Kerdprasop, Chaknarin Kongcharoen and Tippaya Tippayasot

Data Engineering and Knowledge Discovery (DEKD) Research Unit
School of Computer Engineering
Suranaree University of Technology
Nakhon Ratchasima 30000, Thailand

ABSTRACT

Learning from a very large data causes a performance problem even with the most efficient algorithm. Natural solution is to use a sample, but it is not obvious to determine the small but sufficient sample size. We explore two methods of progressive sampling: arithmetic and geometric. Progressive sampling is defined as a technique that starts with a small sample and repeatedly uses progressively larger samples until the performance of the resulting model does not further improve. The results of our studies show that geometric progressive sampling works well with the large data sets.

Keywords Progressive sampling, Knowledge discovery, Data mining, Data analysis

1. INTRODUCTION

With the current advancing technology, we are capable of collecting and storing huge amount of data. Data in electronic form grow to the point where a hundred gigabyte is considered small. Analyzing and discovering the knowledge hidden in these massive data sets become an arduous task and require either advanced parallel hardware, or a very efficient data-analysis algorithm. However, most data analysis and mining algorithms are not scaling efficiently to a very large data set. This is due to the fact that the computational complexity of even the fastest algorithm is linear in the number of instances.

The most natural way of dealing with the ever-increasing data is to use a sample from the data set. Learning from a small sample can certainly speed up the knowledge discovery process, but the mined knowledge must be as accurate as learning from the whole data set. To determine the sufficient but small sample size is not obvious. Therefore, we empirically explore a method of progressive sampling on large data sets using different learning algorithms with the purpose to study the efficiency of the method.

2. PROGRESSIVE SAMPLING

Using sampling[2] to reduce the size of the data set has long been an issue in data-intensive applications. John and Langley[3] study static and dynamic sampling methods for data mining. They refer to static sampling as a method to obtain a required sample by

determining statistically whether a sample is sufficiently a representative of its original data set. The word static is used to convey the information that this sampling technique is independent of the following analysis or learning technique. John and Langley use chi-square and large-sample tests as the criteria to confirm that each attribute in the sample comes from the same distribution as the original population.

Dynamic sampling,on the contrary, is a sampling technique that uses the prior knowledge about the mining algorithm to choose an optimal sample size. Dynamic sampling has been shown[1,3,5] an efficient and preferable technique to static sampling.

Provost, Jensen and Oates[6] study the dynamic sampling approach and rename it progressive sampling. They define progressive sampling as a sampling technique that starts with a small sample and repeatedly uses progressively larger samples until the performance of the resulting model no longer improves. The progressive sampling algorithm can be defined as follows.

Algorithm Progressive Sampling
Input: Schedule S = $\{n_1, n_2, ..., n_k\}$ of sample sizes when $n_j > n_i$ for $j > i$, and n_k is all training data.
Output: An accurate model M that is trained from a sample n_i.
1. Initialize accuracy Acc = 10, i = 1
2. M ← model induced from sample n_i
3. While $Acc_M > Acc$
   // Acc_M is accuracy of model M
   3.1) i ← i + 1
   3.2) Acc ← Acc_M
   3.3) M ← model induced from sample n_i
   3.4) end while
4. Return M

3. EMPIRICAL COMPARISON OF SAMPLING SCHEDULE

John and Langley[3] compare dynamic sampling with static sampling and show that dynamic sampling produces more accurate model than does static sampling. We extend their study by examining two different schedules of dynamic sampling, which we refer to as progressive sampling in this paper. We study arithmetic progressive sampling and geometric progressive sampling. According to Provost, Jensen and Oates[6], arithmetic progressive sampling is a sampling method using the schedule $S_a = n_1 + (i * n_\delta)$ when $i \geq 0$ and n_δ is some fixed number of instances. In our empirical study, S_a = {100, 300, 500, 700, ..., 3300, 3500, all instances}. Geometric progressive sampling, on the other hand,uses the schedule $S_g = a^i * n_1$. We set the schedule S_g = {100, 200, 400, 800, 1600, 3200, 6400, 12800, 25600, all instances}, that is, a = 2, $i \geq 0$ and n_1 = 100.

We compare arithmetic versus geometric progressive sampling using the learning algorithm naive Bayes and C4.5 trained on three large data sets: shuttle (43,500 instances in training data and 14,500 test data instances), adult (32,561 instances in training data and 16,281 test data instances), and letter (15,000 instances in training data and 5,000 test data instances). These data sets are taken from the UCI Repository[4]. The comparative results running with naive Bayes and C4.5 algorithms are shown in Figure 1 and 2, respectively.

4. DISCUSSION

For the letter data set running with naive Bayes algorithm, the highest model accuracy is 63.88%. The geometric progressive sampling can reach this highest point in the seventh sampling step (sample size = 6400 instances), whereas with the arithmetic method, more than eighteen sampling steps are required. The observation that geometric progressive sampling reaches the highest model accuracy faster than the arithmetic method is also applied to the adult data set. It is also true with the shuttle data set except that the highest accuracy found by the geometric method is 95.06%,but the arithmetic method can reach the accuracy 95.49% with more sampling steps, though.

The same notice is also applicable to the experimental results obtained from C4.5 algorithm. Empirical studies on large three data sets shows that C4.5 can induce a more accurate model than does the naive Bayes but with the price of more sampling steps.

Final remark regarding learning with large data sets is that using a sample of approximately 1,500-2,000 instances should yield a reasonable accurate model. This absolutely requires further investigation.

5. CONCLUSION

Learning from a very large data set incurs computational complexity even with the fastest learning algorithm. The most natural way of dealing with the ever-increasing data is to use a sample from the data set. However, determining the small but sufficient sample size (in the sense that the accurate model should be induced) is not obvious.

We empirically investigate two kinds of progressive sampling methods: arithmetic and geometric. Arithmetic sampling grows sample sizes linearly, whereas the geometric method exponentially grows sample data. The experimantal results show that with only five to nine trials, the geometric progresive sampling method produce the highest accurate model. We plan to extend our study to make the sampling schedule adaptive to the resulting model accuracy.

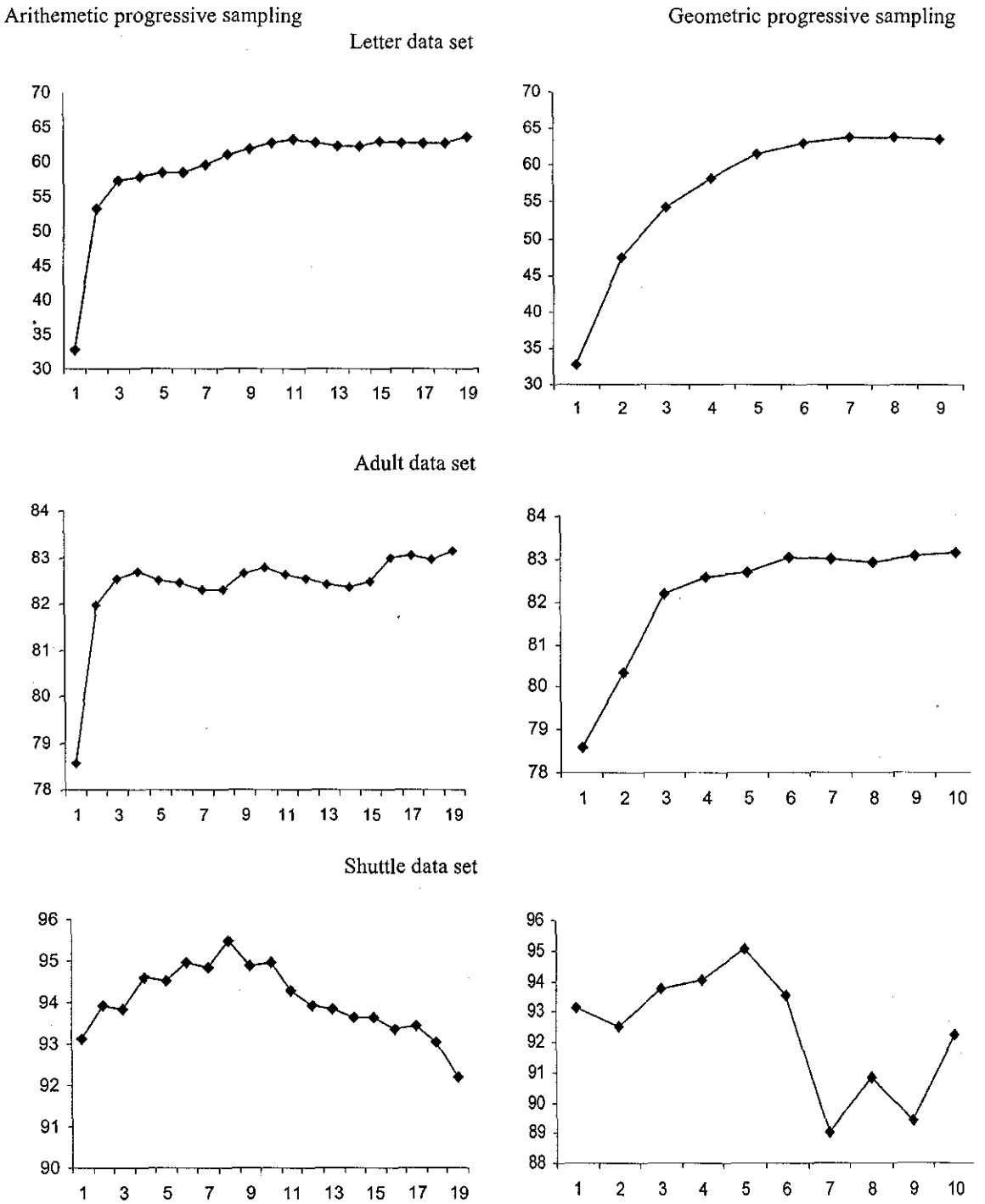


Figure 1 Arithmetic and geometric progressive sampling running with naive Bayes

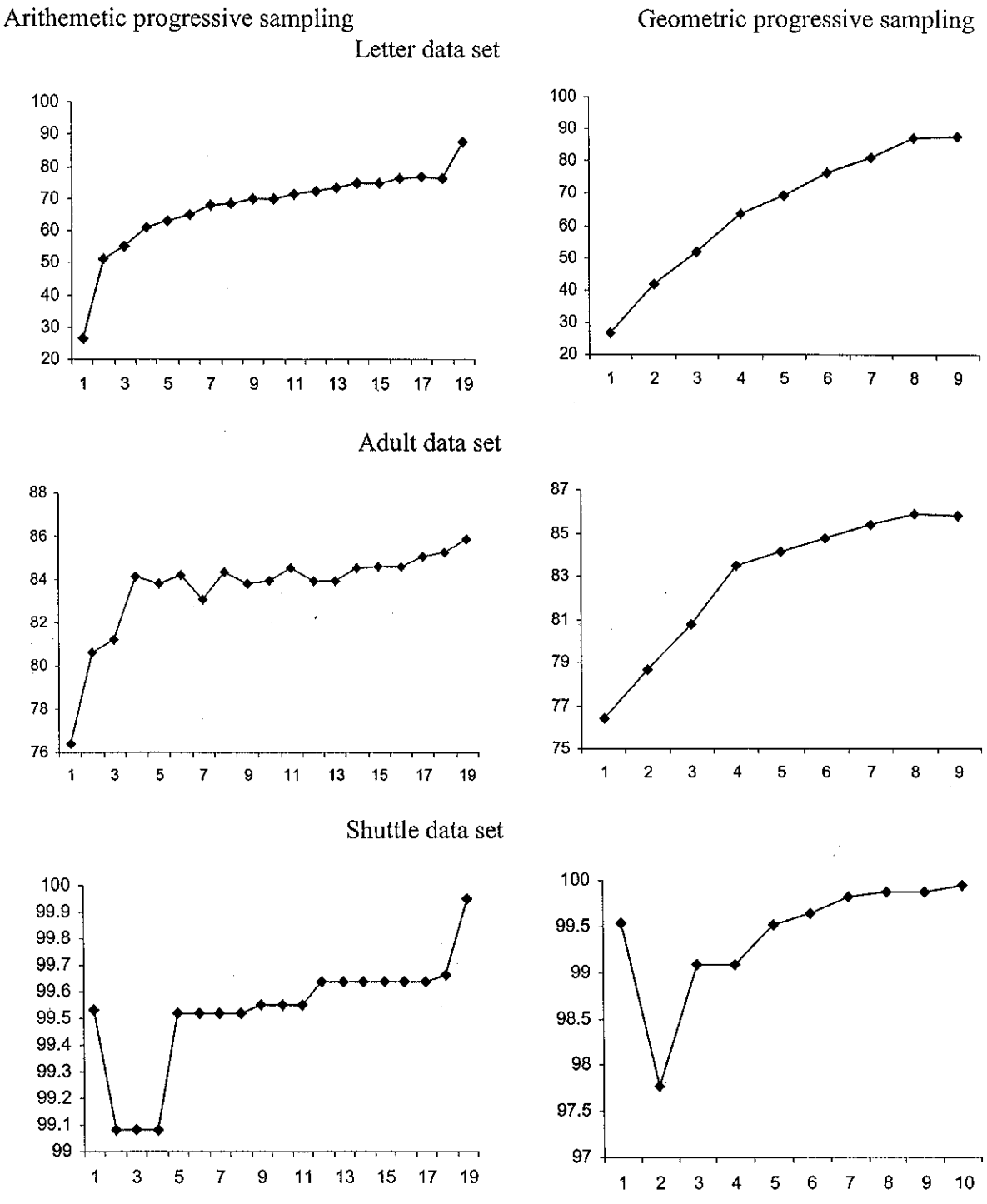


Figure 2 Arithmetic and geometric progressive sampling running with C4.5

6. REFERENCES

1. Catlett, W.G. (1992). *Peepholing: Choosing Attributes Efficiently for Megainduction.* In Machine Learning: Proceedings of the Ninth International Workshop. 49-54. Morgan-Kaufmann.

2. Cochrane, J. (1977). *Sampling Techniques.* John Wiley & Sons.

3. John, G.H. and Langley, P. (1996). *Static Versus Dynamic Sampling for Data Mining.* In Proceedings of the Second International Conference on Knowledge Discovery in Databases an Data Mining. 367-370. AAAI/MIT Press.

4. Merz, C.J. and Murphy, P.M. (1997). *UCI Repository of Machine Learning Databases.* http://www.ics.uci.edu/~mlearn/MLRepository.

5. Moore, A. and Lee, M. (1994). *Efficient Algorithms for Minimizing Cross-Validation Error.* In Machine Learning: Proceedings of the Eleventh International Conference. 190-198. Morgan-Kaufmann.

6. Provost, F., Jensen, D. and Oates, J. (1999). *Efficient Progressive Sampling.* In Proceedings of the Fifth International Conference on Knowledge Discovery in Databases an Data Mining. 23-32.

ประวัติผู้วิจัย

ผู้ช่วยศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ สำเร็จการศึกษาในระดับปริญญาเอกสาขา Computer Science จาก Nova Southeastern University เมือง Fort Lauderdale รัฐฟลอริดา สหรัฐอเมริกา เมื่อปีพุทธศักราช 2542 (ค.ศ. 1999) ด้วยทุนการศึกษาของทบวงมหาวิทยาลัย (หรือสำนักงานคณะกรรมการอุดมศึกษาในปัจจุบัน) โดยทำวิทยานิพนธ์ระดับปริญญาเอกในหัวข้อเรื่อง "Active database rule set reduction by knowledge discovery" หลังสำเร็จการศึกษาได้ปฏิบัติงานในตำแหน่งอาจารย์ ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ปัจจุบันดำเนินการวิจัยเกี่ยวกับการพัฒนาระบบเหมืองข้อมูลประสิทธิภาพสูงที่สามารถทนต่อข้อมูลรบกวน และการวิจัยพื้นฐานเกี่ยวกับเทคนิคการจัดกลุ่มข้อมูล และการวิเคราะห์ข้อมูลโดยวิธีอัตโนมัติ โดยมีผลงานวิจัยตีพิมพ์ในวารสารวิชาการและเอกสารการประชุมวิชาการ จำนวนมากกว่า 20 เรื่อง ในสาขาฐานข้อมูลแอคทีฟ ฐานข้อมูลนิรนัย การทำเหมืองข้อมูลและการค้นหาความรู้