การจำแนกประเภทข้อมูลด้วยเทคนิคต้นไม้ตัดสินใจและการจัดกลุ่ม

DATA CLASSIFICATION WITH DECISION TREE AND CLUSTERING

นายชินพัฒน์ แก้วชินพร

นางสาวณัฐกานต์ วงศ์สุโชโต

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

DATA CLASSIFICATION WITH DECISION TREE AND CLUSTERING

MR. CHINNAPAT KAEWCHINPORN

MISS NATTAKAN VONSUCHOTO

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR DEGREE OF BACHELOR OF SCIENCE

IN COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

หัวข้อโครงงานพิเศษ การจำแนกประเภทข้อมูลด้วยเทคนิคต้นไม้ตัดสินใจและการจัดกลุ่ม
DATA CLASSIFICATION WITH DECISION TREE AND CLUSTERING

ชื่อนักศึกษา นายชินพัฒน์ แก้วชินพร 50050117
นางสาวณัฐกานต์ วงศ์สุโชโต 50050124

ปริญญา วิทยาศาสตรบัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา ดร.อนันตพร ศรีสวัสดิ์

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2553

คณะกรรมการสอบ	ลายมือชื่อ
อาจารย์วิสันต์ ตั้งวงษ์เจริญ ประธานกรรมการ	
ดร.วรางคณา กิ้มปาน กรรมการ	
ดร.อนันตพร ศรีสวัสดิ์ กรรมการและอาจารย์ที่ปรึกษา	

หัวข้อโครงงานพิเศษ การจำแนกประเภทข้อมูลด้วยเทคนิคต้นไม้ตัดสินใจและการจัดกลุ่ม
ชื่อนักศึกษา นายชินพัฒน์ แก้วชินพร 50050117
นางสาวณัฐกานต์ วงศ์สุโชโต 50050124
ปริญญา วิทยาศาสตรบัณฑิต
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2553
อาจารย์ที่ปรึกษา ดร.อนันตพร ศรีสวัสดิ์

บทคัดย่อ

งานวิจัยนี้นำเสนอขั้นตอนวิธีการจำแนกประเภทข้อมูลแบบใหม่ที่นำเทคนิคต้นไม้ตัดสินใจและการจัดกลุ่มมาทำงานร่วมกัน ซึ่งในงานวิจัยนี้ได้เสนอขั้นตอนวิธีที่เรียกว่า Tree Bagging and Weighted Clustering Algorithm (TBWC) ที่พัฒนาขึ้นเพื่อเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลด้วยเทคนิคการจัดกลุ่ม ในการทดลองมีการใช้ชุดข้อมูลจำนวน 5 ชุด จากผลการทดลองพบว่า ขั้นตอนวิธี TBWC ให้ค่าความแม่นยำสูงกว่าขั้นตอนวิธีต้นไม้ตัดสินใจและการจัดกลุ่มในทุกชุดข้อมูลเรียนรู้ สามารถปรับปรุงประสิทธิภาพในการทำนายได้ดีมากในชุดข้อมูลเรียนรู้ที่มีจำนวนหลายกลุ่ม ซึ่งสามารถเพิ่มค่าความแม่นยำในการทำนายได้ดีขึ้นมากถึง 36.67% และสามารถลดทอนคุณลักษณะได้สูงสุดถึง 59.82%

คำสำคัญ : ขั้นตอนวิธีการจำแนกประเภทข้อมูล, ต้นไม้ตัดสินใจ, การจัดกลุ่ม, การรวมกันของตัวจำแนกประเภท

Title	DATA CLASSIFICATION WITH DECISION TREE AND CLUSTERING
Students	Mr.Chinnapat Kaewchinporn 50050117
	Miss Nattakan Vongsuchoto 50050124
Degree	Bechelor of Science
MajorProgram	Computer Science
Academic Year	2010
Advisor	Dr.AnantapornSrisawat

ABSTRACT

This special problem presents a new classification algorithm which is a combination of decision tree learning and clustering called Tree Bagging and Weighted Clustering (TBWC). The TBWC algorithm was developed to enhance a classification performance of a clustering algorithm. In the experiments, five datasets were used to evaluate the predictive performance. The experimental results show that the TBWC algorithm yields the highest accuracies when compared with decision tree learning and clustering for all datasets. In addition, this algorithm can improve the predictive performance especially for multi-class datasets which can increase the accuracy up to 36.67%. Finally, it can reduce attributes up to 59.82%.

Keywords :data classification, decision tree, clustering, combination algorithm

กิตติกรรมประกาศ

ในการจัดทำคู่มือการทำปัญหาพิเศษนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความช่วยเหลือและสนับสนุนจากบุคคลหลายท่าน ดร.อนันตพร ศรีสวัสดิ์ กรรมการและอาจารย์ที่ปรึกษา ซึ่งเป็นผู้เสียสละเวลาในการแนะแนวทางพัฒนา ชี้ให้เห็นถึงปัญหา และคอยตรวจสอบความเรียบร้อยของงานมาโดยตลอด อาจารย์วิสันต์ ตั้งวงษ์เจริญ และ ดร.วรางคณา กิ้มปาน ประธานกรรมการ และกรรมการ ซึ่งเป็นผู้ให้คำแนะนำและชี้จุดบกพร่องที่ควรแก้ไข ทางคณะผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างยิ่งในความกรุณาของทุกท่านไว้ ณ ที่นี้

สุดท้ายนี้ขอขอบพระคุณคณาอาจารย์ในภาควิชา วิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ ซึ่งได้ให้ความรู้ทางวิชาการ จนกระทั่งผู้จัดทำพอมีความสามารถที่จะดำเนินปัญหาพิเศษสำเร็จลุล่วงได้เช่นนี้ ขอบพระคุณทุกท่านจากใจจริง

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญภาพ

สารบัญภาพ (ต่อ)

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากในปัจจุบันการจำแนกประเภทข้อมูล (Classification) เป็นปัญหาที่มีบทบาทสำคัญในเรื่องการค้นหาความรู้จากข้อมูลปริมาณมหาศาล (Knowledge Discovery in Database : KDD) เพื่อจัดการปัญหานี้จึงได้มีผู้คิดค้นวิธีการ เทคนิคต่างๆ มากมาย ไม่ว่าจะเป็น ต้นไม้ตัดสินใจ (Decision Tree) ทฤษฎีความน่าจะเป็นแบบเบย์ (Naïve Bayes) การเรียนรู้โดยตัวอย่าง (Instance-based learning) โครงข่ายประสาทเทียม (Artificial Neural Network) ซึ่งเทคนิคเหล่านี้ล้วนมีแนวคิดและวิธีการทำงานที่ต่างกัน แต่เทคนิคทั้งหมดที่กล่าวมานั้นมีจุดหมายที่จะทำนายกลุ่มของข้อมูลที่เพิ่มเข้ามาใหม่ได้อย่างแม่นยำ

เทคนิคในการจำแนกประเภทข้อมูลโดยทั่วไปที่กล่าวมานั้น ยังมีจุดด้อยบางประการ อาทิ การจัดการข้อมูลแปลกแยก (Outliner Analysis) ความถูกต้องแม่นยำจะมีแนวโน้มต่ำลง เมื่อข้อมูลทดสอบ (Test Set) มีปริมาณสูงขึ้น เวลาที่ใช้ในการจำแนกประเภทข้อมูลออกมาจะมีแนวโน้มนานขึ้น เมื่อข้อมูลมีค่าคุณลักษณะ (Attribute) ปริมาณมาก ซึ่งถึงแม้บางเทคนิคจะรองรับการจัดการปัญหาได้ แต่ก็ยังไม่สามารถจัดการปัญหาทั้งหมดนี้ได้

โครงงานนี้จึงเกิดขึ้นเพื่อปรับปรุงเทคนิคในการจำแนกข้อมูลแบบเดิมๆ ด้วยวิธีการประยุกต์เทคนิคหลายๆ วิธีเข้าด้วยกัน นำจุดเด่นที่สำคัญมาใช้ร่วมกับแนวคิดที่แปลกใหม่ เพื่อจัดการปัญหาการจำแนกประเภทข้อมูล ซึ่งคิดว่าจะสามารถรองรับและจัดการกับปัญหาดังกล่าวได้อย่างมีประสิทธิภาพมากยิ่งขึ้น และทำนายกลุ่มของข้อมูลได้แม่นยำกว่าเดิม

1.2 วัตถุประสงค์

1) เพื่อศึกษาค้นคว้าเกี่ยวกับขั้นตอนวิธีการจำแนกประเภท การประยุกต์ใช้การรวมกันของขั้นตอนวิธีต่างๆ และการวัดประสิทธิภาพการทำงานของขั้นตอนวิธี

2) เพื่อสร้างขั้นตอนวิธีการจำแนกประเภทข้อมูลใหม่ที่มีประสิทธิภาพมากยิ่งขึ้น

1.3 ข้อจำกัดและขอบเขต

ข้อมูลทดสอบจะต้องเป็นชนิดข้อมูลแบบจำแนกประเภท (Classification) เท่านั้น และมีประเภทของข้อมูลแบบหลายกลุ่ม (Multiple Classes)

1.4 ขั้นตอนการดำเนินการ

1) ศึกษาข้อมูลเกี่ยวกับขั้นตอนวิธีต่างๆ ในการจำแนกประเภทข้อมูล รวมถึงผลงานวิจัยเกี่ยวกับด้านนี้

2) ศึกษาข้อมูลเกี่ยวกับการใช้งานโปรแกรมวิเคราะห์ข้อมูล (MATLAB) ในด้านการทำงาน การประยุกต์ใช้งาน การโปรแกรมเพื่อทำงานตามขั้นตอนวิธีที่ได้คิดค้นขึ้นมา

3) เขียนโปรแกรมเพื่อทำงานตามขั้นตอนวิธีที่ได้คิดค้นขึ้นมา

4) รวบรวมและทำความเข้าใจชุดข้อมูลเพื่อนำมาใช้สอนและทดสอบขั้นตอนวิธีที่คิดค้น

5) ทำการทดลองจากชุดข้อมูลกลาง เพื่อทดสอบประสิทธิภาพของขั้นตอนวิธีที่คิดค้น

6) ประเมิน และสรุปผลจากผลลัพธ์ที่ได้จากการทดลองทั้งหมด ในรูปแบบของกราฟและคำอธิบาย

7) จัดทำรูปเล่มคู่มือการทำปัญหาพิเศษ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ขั้นตอนวิธีใหม่ ที่มีความสามารถในการจำแนกประเภทข้อมูลได้อย่างมีประสิทธิภาพ

2) สามารถนำขั้นตอนวิธีที่พัฒนาขึ้นมาแก้ไขปัญหาการจำแนกประเภทข้อมูล เมื่อมีข้อมูลที่มีปริมาณมากได้อย่างมีประสิทธิภาพ

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงเรื่องของทฤษฎีพื้นฐานซึ่งจะมีความเกี่ยวข้องในการทำ การทดลอง และสรุปงานวิจัยที่เกี่ยวข้องกับปัญหาที่กำลังศึกษา โดยจะเริ่มอธิบายถึง "การจำแนกประเภทข้อมูล" ว่าคืออะไร มีความสำคัญอย่างไร ต่อมาจะพูดถึงวิธีการพื้นฐานที่นำมาใช้ในการจำแนกประเภทข้อมูลนั้นคือ เรื่องของ "ต้นไม้ตัดสินใจ" และ "การจัดแบ่งกลุ่ม" ส่วนถัดไปจะเป็นเรื่องของวิธีการประยุกต์นำวิธีการพื้นฐานมาใช้งานร่วมกัน หรือเรียกว่า "การรวมกันของตัวจำแนกประเภท" ต่อจากนั้นจะพูดถึง "การวัดประสิทธิภาพ" และสุดท้ายจะอธิบายสรุปงานวิจัยต่างๆที่เกี่ยวข้องกับปัญหาที่กำลังศึกษา ซึ่งจะมีการแบ่งงานวิจัยออกเป็น 2 ส่วนหลักๆ คือ งานวิจัยการปรับปรุงวิธีการต้นไม้ตัดสินใจและงานวิจัยการปรับปรุงวิธีการจัดแบ่งกลุ่ม

2.1 การจำแนกประเภทข้อมูล

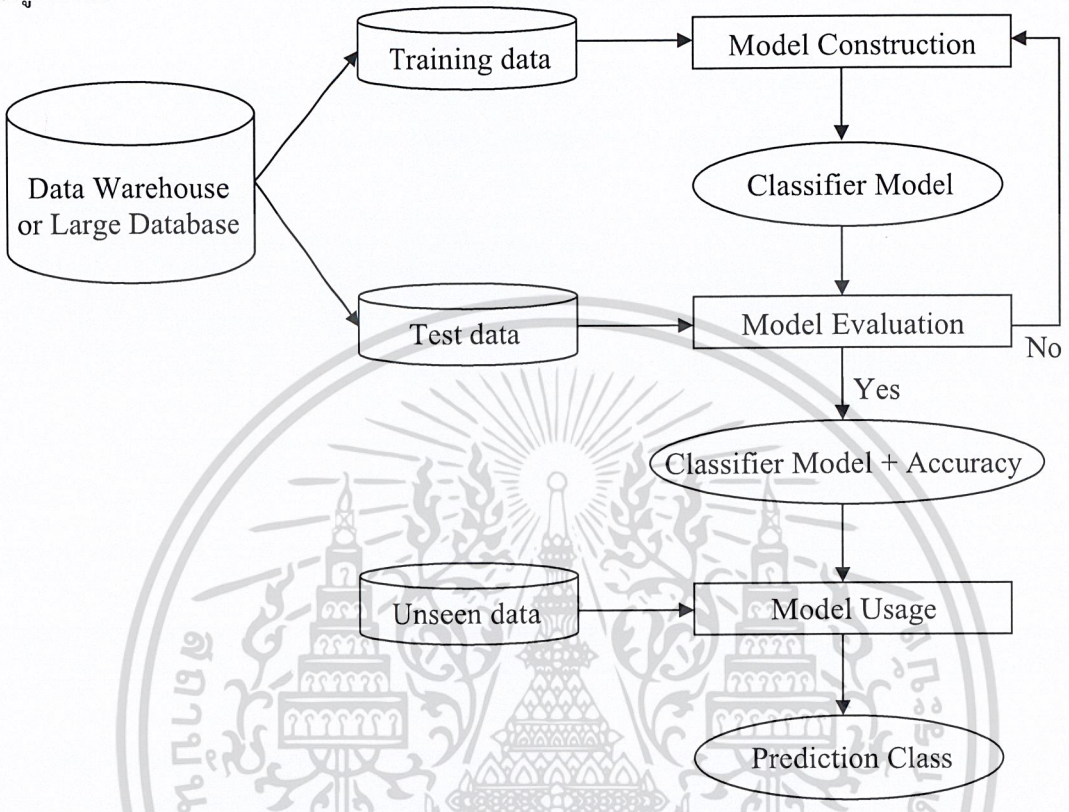
การจำแนกประเภทข้อมูลคือกระบวนการสร้างโมเดลจำแนกประเภทข้อมูล (Data Classification Model) เพื่อทำนายกลุ่มของข้อมูลใหม่ (Unseen data) ตัวอย่างของกลุ่มเช่น กลุ่มของลูกค้าที่ซื้อคอมพิวเตอร์-ไม่ซื้อคอมพิวเตอร์ กลุ่มของลูกค้าที่ฐานะดี-ปานกลาง-แย่ กลุ่มของการผลิตสินค้าผ่านเกณฑ์-ไม่ผ่านเกณฑ์ ในที่นี้คำว่ากลุ่มจะเรียกว่า class ของข้อมูล ซึ่งใน class เดียวกันนั้นจะต้องมีข้อมูลที่มีความเหมือนหรือคล้ายคลึงกันมากกว่าข้อมูลที่อยู่ใน class ที่แตกต่างกัน

การสร้างโมเดลจำแนกประเภทข้อมูล จะเกิดขึ้นมาจากการหาความสัมพันธ์ของข้อมูลในฐานข้อมูลขนาดใหญ่ โดยข้อมูลทั้งหมดจะมีการแบ่งออกเป็น 2 กลุ่มคือกลุ่มข้อมูลเรียนรู้ (Training set) เป็นชุดข้อมูลที่มีบทบาทในการสร้างโมเดลจำแนกประเภทข้อมูลขึ้นมา และมีกลุ่มข้อมูลทดสอบ (Test set) เป็นชุดข้อมูลประเมินความถูกต้องของโมเดลจำแนกประเภทข้อมูล

โมเดลจำแนกประเภทข้อมูลได้ถูกนำมาประยุกต์ใช้งานหลายๆ ด้าน ไม่ว่าจะเป็นการวิเคราะห์หุ้น เพื่อหาว่าหุ้นแต่ละบริษัทมีคุณภาพเป็นอย่างไร เมื่อมีปัจจัยที่เกี่ยวข้อง ไม่ว่าจะเป็น การเติบโตของรายได้ ความสามารถในการควบคุมต้นทุน ความผันผวนของรายได้และกำไร และผู้บริหาร หรือจะเป็นการพยากรณ์อากาศ การจัดสรรกฎหมายที่เหมาะสมในการพิจารณาคดีความ การจัดการความสัมพันธ์ของลูกค้า (CRM) และอื่นๆ

2.1.1 กระบวนการสร้างตัวโมเดลจำแนกประเภทข้อมูล

แบ่งออกเป็น 3 ขั้นตอน ซึ่งภาพรวมของกระบวนการสร้างโมเดลจำแนกประเภทข้อมูลแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 กระบวนการสร้างโมเดลจำแนกประเภทข้อมูล

กระบวนการของแต่ละขั้นตอนมีดังนี้

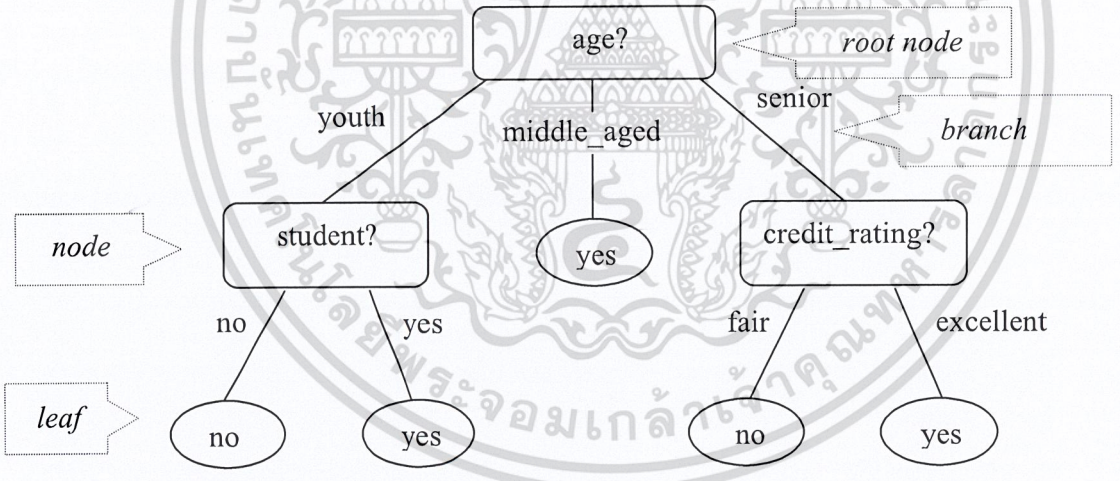
1) Model Construction (Learning) เป็นขั้นตอนการสร้างโมเดลจำแนกประเภท โดยอาศัยการเรียนรู้จากข้อมูลที่ได้กำหนด class ไว้เรียบร้อยแล้วหรือเรียกว่าข้อมูลเรียนรู้ (Training data) ซึ่งโมเดลจำแนกประเภทที่ได้จะแสดงด้วยวิธีการพื้นฐานทางเหมืองข้อมูล (Data mining) ยกตัวอย่างเช่น ต้นไม้ตัดสินใจ (Decision Tree) โมเดลจำแนกประเภทที่ได้จะมีลักษณะคล้ายต้นไม้จริงกลับหัวที่มีโหนดรากอยู่ด้านบนสุดและโหนดใบอยู่ล่างสุดของต้นไม้ แต่ละโหนดบนต้นไม้จะมีคุณลักษณะ (attribute) เป็นตัวเลือกทดสอบ ซึ่งจะมีกิ่งซึ่งเป็นค่าที่เป็นไปได้ของคุณลักษณะ (attribute value) ที่ถูกเลือกทดสอบไว้และมีโหนดใบแสดง class ที่กำหนดไว้

2) Model Evaluation (Accuracy) เป็นขั้นตอนตรวจสอบความถูกต้อง โดยอาศัยข้อมูลที่ใช้สำหรับทดสอบเรียกว่าข้อมูลทดสอบ (Testing data) ซึ่งกลุ่มที่แท้จริงของข้อมูลที่ใช้ทดสอบจะถูกนำมาเปรียบเทียบกับกลุ่มที่หามาได้จากโมเดลจำแนกประเภท เพื่อทดสอบว่าโมเดลจำแนกประเภทนี้สามารถจัดกลุ่มประเภทข้อมูลได้อย่างถูกต้องมากน้อยเพียงใด และมีการปรับปรุงโมเดลจำแนกประเภทจนกว่าจะได้ค่าความถูกต้องในระดับที่ยอมรับได้

3) Model Usage (Classification) เป็นขั้นตอนการนำโมเดลจำแนกประเภทที่สร้างขึ้นมาใช้กับข้อมูลที่ไม่เคยเห็นมาก่อน (unseen data) เพื่อทำนายและกำหนดกลุ่มให้กับข้อมูลนั้น

2.2 ต้นไม้ตัดสินใจ (Decision Tree)

ต้นไม้ตัดสินใจ (Decision Tree) เป็นโครงสร้างข้อมูลชนิดเป็นลำดับชั้น (hierarchy) ใช้สนับสนุนการตัดสินใจ โดยจะมีลักษณะคล้ายต้นไม้จริงกลับหัวที่มีโหนดรากอยู่ด้านบนสุดและโหนดใบอยู่ล่างสุดของต้นไม้ โดยที่ภายในต้นไม้จะประกอบไปด้วยโหนด (node) ซึ่งแต่ละโหนดจะมีคุณลักษณะ (attribute) เป็นตัวทดสอบ กิ่งของต้นไม้ (branch) แสดงถึงค่าที่เป็นไปได้ของคุณลักษณะที่ถูกเลือกทดสอบ และใบ (leaf) ซึ่งเป็นสิ่งที่อยู่ล่างสุดของต้นไม้ตัดสินใจแสดงถึงกลุ่มของข้อมูล (class) หรือนั่นก็คือผลลัพธ์ที่ได้จากการทำนาย โหนดที่อยู่บนสุดของต้นไม้เรียกว่าโหนดราก (root node) ดังแสดงโครงสร้างของต้นไม้ตัดสินใจตัดสินใจดังรูปที่ 2.2 ซึ่งเป็นต้นไม้ที่ใช้ในการตัดสินใจว่าจะเลือกซื้อคอมพิวเตอร์หรือไม่ (Quinlan, 1986) มีคุณลักษณะที่พิจารณาคืออายุ (age) นักศึกษา (student) และอัตราเครดิต (credit_rating) โดยที่โหนดสี่เหลี่ยมมุมโค้งจะเป็นการทดสอบคุณลักษณะของข้อมูล ท้ายสุดจะได้ผลลัพธ์ของการทำนายว่าจะซื้อคอมพิวเตอร์ (yes) หรือไม่ซื้อคอมพิวเตอร์ (no) จากการทดสอบตามเส้นทางของต้นไม้ตัดสินใจตั้งแต่โหนดรากไปจนถึงใบ



รูปที่ 2.2 ต้นไม้ตัดสินใจที่ใช้ในการตัดสินใจการเลือกซื้อคอมพิวเตอร์ (Han and Kamber, 2006, p.291)

2.2.1 การสร้างต้นไม้ตัดสินใจ

การสร้างต้นไม้ตัดสินใจจะสร้างในลักษณะจากบนลงล่าง (top-down) นั้นก็คือเริ่มจากการหาคุณลักษณะที่เหมาะสมที่สุดเพื่อนำมาเป็นรากของต้นไม้แล้วจึงแตกกิ่งไปจนถึงใบ โดยขั้นตอนการสร้างต้นไม้ตัดสินใจจะมีดังนี้ (Han and Kamber, 2006, p.293)

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition *D*.

Input: - Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- Attribute_selection_method, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_ attribute* and, possibly, either a *split point* or *splitting subset.*

Output: A decision tree

Method:

(1) Create a node *N*;
(2) **if** tuples in *D* are all of the same class, *C* **then**
(3) return *N* as a leaf node labeled with the class *C*;
(4) **if** *attribute_list* is empty **then**
(5) return *N* as a leaf node labeled with the majority class in *D*; // majority voting
(6) apply **Attribute_selection_method**(*D, attribute_list*) to find the "best" *splitting_criterion*;
(7) label node *N* with *splitting_criterion*;
(8) **if** *splitting_ attribute* is discrete-valued **and**
multiway splits allowed **then** // not restricted to binary trees
(9) *attribute_list* ← *attribute_list – splitting_attribute*; // remove splitting_ attribute
(10) **for each** outcome *j* of *splitting_criterion*
// partition the tuples and grow subtrees for each partition
(11) let D_j be the set of data tuples in *D* satisfying outcome j; // a partition
(12) **if** D_j is empty **then**
(13) attach a leaf labled with the majority class in *D* to node *N*;
(14) **else** attach the node return by Generate_decision_tree (D_j, attribute_list) to node *N*;
end for
(15) return *N*;

รูปที่ 2.3 ขั้นตอนวิธีพื้นฐานในการสร้างต้นไม้ตัดสินใจด้วยข้อมูลเรียนรู้ (Han and Kamber, 2006, p.293)

การสร้างต้นไม้ตัดสินใจจะสร้างในลักษณะจากบนลงล่าง (top-down) นั้นก็คือเริ่มจากการหาคุณลักษณะที่เหมาะสมที่สุดเพื่อนำมาเป็นรากของต้นไม้แล้วจึงแตกกิ่งไปจนถึงใบ โดยขั้นตอนการสร้างต้นไม้ตัดสินใจจะมีดังนี้

1) เริ่มต้นสร้างโหนดขึ้นมาหนึ่งโหนด
2) ถ้าข้อมูลทั้งหมดอยู่ในกลุ่มเดียวกันแล้ว ให้โหนดที่สร้างขึ้นนั้นเป็นโหนดใบและกำหนดค่าด้วยกลุ่มของข้อมูลนั้น
3) ถ้าข้อมูลไม่มีคุณลักษณะใดที่เหมาะสมในการแบ่งกลุ่ม ให้โหนดที่สร้างขึ้นนั้นเป็นโหนดใบและกำหนดค่าด้วยกลุ่มที่มีข้อมูลสนับสนุนมากที่สุด
4) ถ้าข้อมูลมีหลากหลายกลุ่มปะปนกัน จะทำการเลือกคุณลักษณะที่มีความเหมาะสมที่สุดเป็นตัวทดสอบการตัดสินใจ โดยการวัดค่าเกน (gain) ของแต่ละคุณลักษณะ และกำหนดค่าให้โหนดที่สร้างขึ้นด้วยตัวทดสอบการตัดสินใจที่ได้
5) เมื่อได้ตัวทดสอบการตัดสินใจ หลังจากนั้นให้สร้างกิ่งของต้นไม้ด้วยค่าต่างๆที่เป็นไปได้ของตัวทดสอบ และแบ่งข้อมูลออกตามกิ่งต่างๆที่สร้างขึ้น
6) พิจารณาข้อมูลแต่ละกิ่ง หากพบว่าข้อมูลทั้งหมดอยู่ในกลุ่มเดียวกัน ให้ต่อกิ่งด้วยโหนดใบและกำหนดค่าด้วยกลุ่มของข้อมูลนั้น แต่ถ้าพบว่าข้อมูลมีหลากหลายกลุ่มปะปนกัน ให้ทำการวนซ้ำการหาตัวทดสอบการตัดสินใจที่เหมาะสมต่อไป
7) ทำการวนซ้ำเพื่อแบ่งข้อมูลและแตกกิ่งของต้นไม้ไปเรื่อยๆ โดยการวนซ้ำจะสิ้นสุดก็ต่อเมื่อเงื่อนไขข้อใดข้อหนึ่งต่อไปนี้เป็นจริง
    a. ข้อมูลทั้งหมดในโหนดอยู่ในกลุ่มเดียวกัน ให้โหนดที่สร้างขึ้นนั้นเป็นโหนดใบและกำหนดค่าด้วยกลุ่มของข้อมูลนั้น
    b. ไม่มีคุณลักษณะใดที่เหมาะสมในการแบ่งกลุ่ม ให้โหนดที่สร้างขึ้นนั้นเป็นโหนดใบและกำหนดค่าด้วยกลุ่มที่มีข้อมูลสนับสนุนมากที่สุด

2.2.2 ตัวอย่างการสร้างต้นไม้ตัดสินใจด้วยเทคนิค ID3

ตารางที่ 2.1 ข้อมูลเรียนรู้ที่ใช้ประกอบการตัดสินใจซื้อคอมพิวเตอร์ (Han and Kamber, 2006, p.299)

RID	*age*	*income*	*student*	*credit_rating*	Class: *buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

เนื่องจากข้อมูลที่มีนั้นประกอบด้วยข้อมูลหลากหลายกลุ่มปะปนกัน ฉะนั้นจะต้องวัดมาตรฐานเกน (gain) ของแต่ละคุณลักษณะ (attribute) ซึ่งค่ามาตรฐานเกนนี้คำนวณได้โดยใช้ความรู้จากทฤษฎีสารสนเทศ (information gain) ซึ่งค่าสารสนเทศของข้อมูลจะขึ้นอยู่กับความน่าจะเป็นของข้อมูล สามารถเขียนในรูปสมการที่ 2.1 (Han and Kamber, 2006, p.297) ได้ดังนี้

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \qquad (2.1)$$

โดยที่ p_i เป็นความน่าจะเป็นที่ข้อมูลในฐานข้อมูล D อยู่ในกลุ่ม C_i ซึ่งมีค่า $\frac{|C_{i,D}|}{|D|}$

m เป็นจำนวนกลุ่มทั้งหมดที่ต่างกันของข้อมูลชุดนั้น

C_i เป็นกลุ่มในลำดับที่ i โดยที่ i มีค่าระหว่าง 1 ถึง m

$|C_{i,D}|$ เป็นจำนวนข้อมูลในฐานข้อมูล D ที่อยู่ในกลุ่ม C_i

$|D|$ เป็นจำนวนข้อมูลในฐานข้อมูล D

ค่า $Info(D)$ ที่ได้สามารถเรียกอีกชื่อหนึ่งว่าค่า Entropy ของฐานข้อมูล D

ค่าความรู้จากทฤษฎีสารสนเทศจะช่วยในการแยกแยะข้อมูลทำให้ลดจำนวนครั้งของการทดสอบได้ อีกทั้งยังรับประกันว่าต้นไม้ตัดสินใจที่ได้จะไม่มีความซับซ้อนมากจนเกินไป

เมื่อทำการพิจารณาเลือกคุณลักษณะเป็นตัวเลือกทดสอบ จะใช้ค่าความรู้จากทฤษฎีสารสนเทศของคุณลักษณะ สามารถเขียนในรูปสมการที่ 2.2 (Han and Kamber, 2006, p.298) ได้ดังนี้

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \tag{2.2}$$

โดยที่ v เป็นจำนวนค่าที่เป็นไปได้ของคุณลักษณะ

$|D|$ เป็นจำนวนข้อมูลในฐานข้อมูล D

$|D_j|$ เป็นจำนวนข้อมูลในฐานข้อมูล D ที่มีค่าที่ j ของคุณลักษณะ A

ค่ามาตรฐานเกนที่จะใช้พิจารณาเลือกคุณลักษณะ A มาเป็นโหนดของต้นไม้มีค่าเท่ากับผลต่างของความรู้จากทฤษฎีสารสนเทศ กับ ความรู้จากทฤษฎีสารสนเทศของคุณลักษณะ สามารถเขียนในรูปสมการที่ 2.3 (Han and Kamber, 2006, p.298) ได้ดังนี้

$$Gain(A) = Info(D) - Info_A(D) \tag{2.3}$$

เริ่มต้นการสร้างต้นไม้ตัดสินใจเราต้องพิจารณาคุณลักษณะที่มีของข้อมูลเรียนรู้เพื่อเลือกเป็นโหนดราก โดยการคำนวณค่ามาตรฐานเกนของคุณลักษณะที่มีทั้งหมด แล้วจึงตัดสินใจเลือกคุณลักษณะที่มีค่ามาตรฐานเกนสูงที่สุด ในที่นี้จะสังเกตได้ว่าจะมีคุณลักษณะที่สามารถนำมาใช้ตัดสินใจได้คือ *age*, *income*, *student* และ *credit_rating* ส่วนคุณลักษณะ RID มีลักษณะเป็นค่าไม่ซ้ำ (unique value) จึงไม่เหมาะสมในการนำมาใช้ตัดสินใจ และ Class ก็เป็นกลุ่มของข้อมูล ก็ไม่เหมาะสมเช่นกัน

จากตัวอย่างข้อมูลคุณลักษณะประกอบการตัดสินใจซื้อคอมพิวเตอร์ในตารางที่ 2.1 เซตของข้อมูลเรียนรู้ T ประกอบด้วยข้อมูลจำนวน 14 แถว แบ่งข้อมูลออกเป็น 2 กลุ่มคือ ข้อมูลที่ตัดสินใจซื้อคอมพิวเตอร์ (*Class = yes*) จำนวน 9 แถว และตัดสินใจไม่ซื้อคอมพิวเตอร์ (*Class = no*) จำนวน 5 แถว

$$Info(T) = -\left(\frac{9}{14}\right) \times \log_2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \times \log_2\left(\frac{5}{14}\right)$$
$$= 0.940$$

พิจารณาแต่ละคุณลักษณะโดยการหาค่าความรู้จากทฤษฎีสารสนเทศของคุณลักษณะ และค่ามาตรฐานเกนออกมาโดยใช้สมการที่ 2.2 และ 2.3 ตามลำดับ ดังนี้

$$Info_{age}(T) = \left(\frac{5}{14}\right) \times \left(-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right)$$
$$+ \left(\frac{4}{14}\right) \times \left(-\frac{4}{4}\log_2\left(\frac{4}{4}\right) - \frac{0}{4}\log_2\left(\frac{0}{4}\right)\right)$$
$$+ \left(\frac{5}{14}\right) \times \left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right)$$
$$= 0.693$$

$$Gain(age) = Info(T) - Info_{age}(T)$$

$$= 0.940 - 0.693$$

$$= 0.247$$

$$Gain(income) = Info(T) - Info_{income}(T)$$

$$= 0.940 - 0.911$$

$$= 0.029$$

$$Gain(student) = Info(T) - Info_{student}(T)$$

$$= 0.940 - 0.788$$

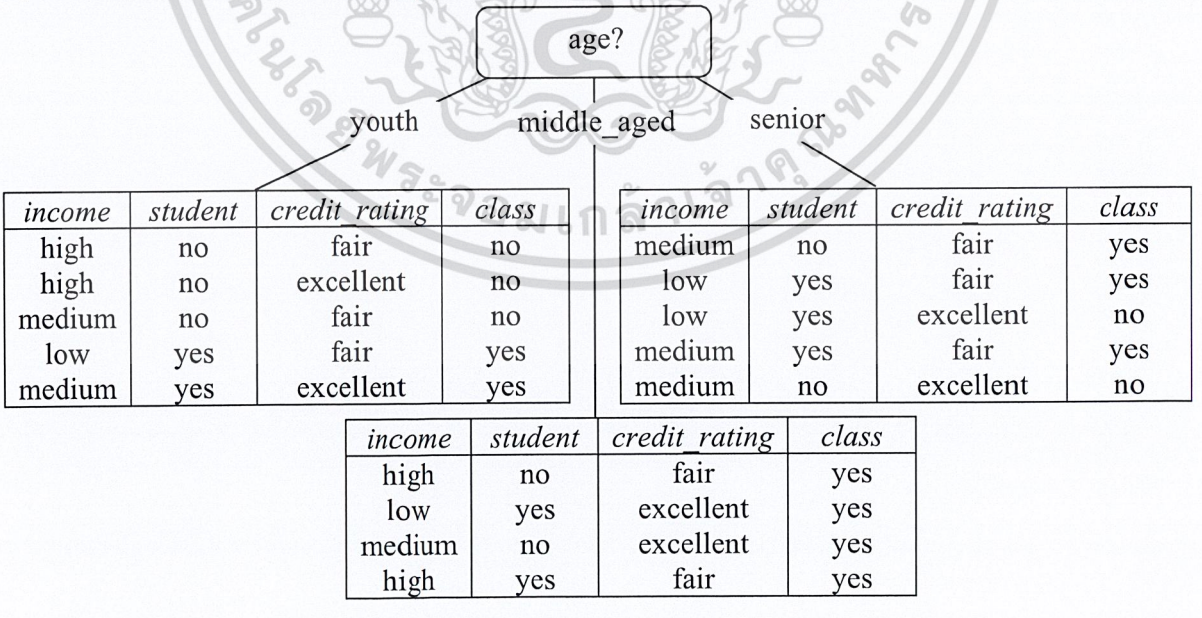
$$= 0.152$$

$$Gain(credit\_rating) = Info(T) - Info_{credit\_rating}(T)$$

$$= 0.940 - 0.892$$

$$= 0.048$$

ซึ่งจะเห็นได้ว่าคุณลักษณะที่ให้ค่ามาตรฐานเกนสูงที่สุดคือ *age* ดังนั้นคุณลักษณะ *age* จึงถูกเลือกเป็นโหนดรากของต้นไม้ตัดสินใจ แต่เนื่องจากข้อมูลทั้งหมดของแต่ละกิ่งไม่ได้อยู่ในกลุ่มเดียวกันทั้งหมด จึงต้องมีการเลือกโหนดสร้างต้นไม้ตัดสินใจต่อไปบนกิ่งของโหนดราก ยกเว้นกรณี *age = middle_aged* จะไม่ต้องสร้างต้นไม้ตัดสินใจเพิ่มเติมเนื่องจากสามารถจัดกลุ่มของข้อมูลที่เป็นกลุ่ม yes ได้ทั้งหมดแล้ว ซึ่งจะแสดงได้ดังรูปที่ 2.4



income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

รูปที่ 2.4 ต้นไม้ตัดสินใจที่ได้จากการเลือกคุณลักษณะ *age* เป็นโหนดราก (Han and Kamber,2006, p.300)

การเลือกโหนดระดับที่ 2 จะมีเพียง *income*, *student* และ *credit_rating* เท่านั้นที่สามารถเป็นตัวทดสอบการตัดสินใจได้ ได้ การสร้างต้นไม้ระดับที่ 2 จะแบ่งพิจารณาเป็นทีละส่วนคือ *age = youth* และ *age = senior* ในที่นี้เราจะพิจารณาต้นไม้ *age = youth* ก่อนโดยใช้วิธีการหาคุณลักษณะที่เหมาะสมซึ่งจะมีการใช้ค่ามาตรฐานเกนดังเดิม ซึ่งแสดงได้ดังนี้

$$Info(age = youth) = -\left(\frac{2}{5}\right) \times \log_2(\frac{2}{5}) - \left(\frac{3}{5}\right) \times \log_2(\frac{3}{5})$$

$$= 0.971$$

พิจารณาแต่ละคุณลักษณะโดยการหาค่าความรู้จากทฤษฎีสารสนเทศของคุณลักษณะ และค่าเกนออกมาโดยใช้สมการที่ 2.2 และ 2.3 ตามลำดับ ดังนี้

$$Info_{income}(age = youth) = \left(\frac{2}{5}\right) \times \left(-\frac{0}{2}\log_2(\frac{0}{2}) - \frac{2}{2}\log_2(\frac{2}{2})\right)$$
$$+ \left(\frac{2}{5}\right) \times \left(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})\right)$$
$$+ \left(\frac{1}{5}\right) \times \left(-\frac{1}{1}\log_2(\frac{1}{1}) - \frac{0}{1}\log_2(\frac{0}{1})\right)$$
$$= 0.4$$

$$Gain(income) = Info(age = youth) - Info_{income}(age = youth)$$
$$= 0.971 - 0.4$$
$$= 0.571$$

$$Gain(student) = Info(age = youth) - Info_{student}(age = youth)$$
$$= 0.971 - 0$$
$$= 0.971$$

$$Gain(credit\_rating) = Info(age = youth) - Info_{credit\_rating}(age = youth)$$
$$= 0.971 - 0.951$$
$$= 0.020$$

จะเห็นว่าคุณสมบัติที่ให้ค่ามาตรฐานเกนสูงที่สุดคือ *student* ดังนั้นจึงเลือกคุณสมบัตินี้เป็นโหนดระดับที่ 2 ต่อจาก *age = youth* และเนื่องจากข้อมูลทั้งหมดของแต่ละกิ่งอยู่ในกลุ่มเดียวกันทั้งหมด ดังนั้นจึงไม่ต้องสร้างต้นไม้ตัดสินใจต่อไป แต่ยังคงเหลือโหนดระดับ 2 ทางขวา (*age = senior*) ที่ต้องพิจารณาเลือกคุณสมบัติด้วยวิธีการเดียวกับที่ผ่านมา ซึ่งจะได้คุณสมบัติ *credit_rating* มีค่าเกนสูงที่สุด จึงจะถูกเลือกเป็นโหนดระดับ 2 ต่อจาก *age = senior* ซึ่งทำให้ข้อมูลทั้งหมดของแต่ละกิ่งอยู่ในกลุ่มเดียวกันทั้งหมด และจะได้โครงสร้างเป็นดังรูปที่ 2.2

2.2.3 ขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5

ขั้นตอนวิธี C4.5 เป็นวิธีการที่มีชื่อเสียงและเป็นที่รู้จักอย่างแพร่หลาย พัฒนาโดย Ross Quinlan (1993) โดยพัฒนาต่อมาจากขั้นตอนวิธี ID3 ที่เขาได้พัฒนาขึ้น (Ross Quinlan, 1986) ซึ่งขั้นตอนวิธีนี้ใช้เพื่อสร้างต้นไม้ตัดสินใจสำหรับจัดแบ่งกลุ่มข้อมูล และมีการใช้หลักการของ Information gain เช่นเดียวกับ ID3 แต่จะมีส่วนเพิ่มเติมจาก ID3 เข้ามา ซึ่งสามารถแก้ไขจุดด้อยของ ID3 ได้เป็นอย่างดี ดังนี้

1) สามารถใช้งานได้ทั้งข้อมูลแบบต่อเนื่อง (Continuous data) และแบบไม่ต่อเนื่อง (Discrete data) โดยในส่วนของข้อมูลแบบต่อเนื่องนั้น C4.5 จะสร้างจุดแบ่ง (Threshold) แยกคุณลักษณะนั้นออกเป็น 2 ส่วน คือส่วนที่มีค่ามากกว่ากับน้อยกว่าเท่ากับค่าที่ใช้ในการสร้างจุดเริ่ม
2) สามารถใช้กับชุดข้อมูลทดสอบ ที่มีค่าข้อมูลขาดหายได้ (missing data) โดยจะแทนค่าด้วย "?" และไม่นำค่านั้นมาคำนวณในกฎของความรู้จากทฤษฎีสารสนเทศ
3) สามารถใช้กับชุดข้อมูลทดสอบที่มีค่าผิดปกติหรือมีความเสียหายได้
4) สามารถทำการตัดกิ่งต้นไม้ตัดสินใจในขณะสร้างได้ โดยไม่ทำให้ความถูกต้องลดลง

การเลือกคุณลักษณะที่ใช้เป็นโหนดรากหรือโหนดบนต้นไม้ตัดสินใจนั้นขั้นตอนวิธี ID3 จะใช้ค่าเกนเป็นหลักในการเลือก แต่ขั้นตอนวิธี C4.5 นั้นได้เพิ่มการใช้ค่ามาตรฐานอัตราส่วนเกน (Gain ratio criterion) ในการตัดสินใจเลือกคุณลักษณะ เนื่องจากค่าเกนจะมีการเอนเอียง (Bias) อย่างมากกับข้อมูลที่ประกอบด้วยคุณลักษณะที่มีค่าที่เป็นไปได้จำนวนมากๆ

การแก้ไขความเอนเอียงของค่าเกนสามารถทำได้โดยการปรับค่ามาตรฐานเกนให้ถูกต้องโดยใช้ค่าสารสนเทศของการแบ่งแยก (Split information) ของคุณลักษณะแต่ละตัว ค่าสารสนเทศของการแบ่งแยกสามารถเขียนในรูปสมการที่ 2.4 (Han and Kamber, 2006, p.301) ได้ดังนี้

$$SplitInfo(A) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right) \tag{2.4}$$

ค่าสารสนเทศของการแบ่งแยกนี้จะแสดงถึงระดับการกระจายของข้อมูล เมื่อนำค่านี้ไปหารค่าเกนจะได้ค่ามาตรฐานอัตราส่วนเกน สามารถเขียนในรูปสมการที่ 2.5 (Han and Kamber, 2006, p.301) ได้ดังนี้

$$GainRatio(D) = \frac{Gain(A)}{SplitInfo(A)} \tag{2.5}$$

ค่ามาตรฐานอัตราส่วนเกนช่วยแก้ไขความเอนเอียงของค่าเกนได้ โดยทำให้ค่ามาตรฐานอัตราส่วนเกนในแบ่งด้วยคุณลักษณะที่มีการกระจายสูงถูกปรับลดลง ดังนั้นค่ามาตรฐานอัตราส่วนเกนในคุณลักษณะที่มีการกระจายตัวของข้อมูลสูงดังที่กล่าวมาจึงไม่มีค่าสูงที่สุดเสมอ

2.3 การจัดกลุ่ม(Clustering)

การจัดกลุ่ม (Clustering) เป็นวิธีการที่พิจารณาข้อมูลแต่ละแถวเสมือนเป็นวัตถุ (object) ซึ่งจะมีหลักการเหมือนกับการจำแนกประเภทข้อมูล คือจะทำการแบ่งข้อมูลออกเป็นกลุ่ม (คลัสเตอร์) โดยจะจัดให้ข้อมูลที่มีความคล้ายคลึงกันอยู่ในคลัสเตอร์เดียวกัน และข้อมูลที่อยู่ต่างคลัสเตอร์กันจะมีความคล้ายคลึงกันน้อยที่สุด ซึ่งความเหมือนหรือต่างกันสามารถเปรียบเทียบได้กับความใกล้ชิดกันของวัตถุใดๆ โดยใช้ระยะทางเป็นตัวชี้วัด คุณภาพของแต่ละคลัสเตอร์สามารถอธิบายได้จากเส้นผ่านศูนย์กลางของคลัสเตอร์ (diameter) ซึ่งแสดงระยะห่างมากสุดของวัตถุสองชิ้นที่อยู่ในคลัสเตอร์เดียวกัน แต่ละคลัสเตอร์จะมีตัวแทนที่สามารถแทนวัตถุทุกชิ้นของคลัสเตอร์นั้นได้ เช่น การใช้จุดศูนย์กลางคลัสเตอร์ (centroid) แทนคลัสเตอร์นั้น สำหรับบางเทคนิคตัวแทนของคลัสเตอร์อาจมีได้หลายตัวแทน ทั้งนี้ขึ้นอยู่กับความเหมาะสมของแต่ละเทคนิคที่เลือกใช้

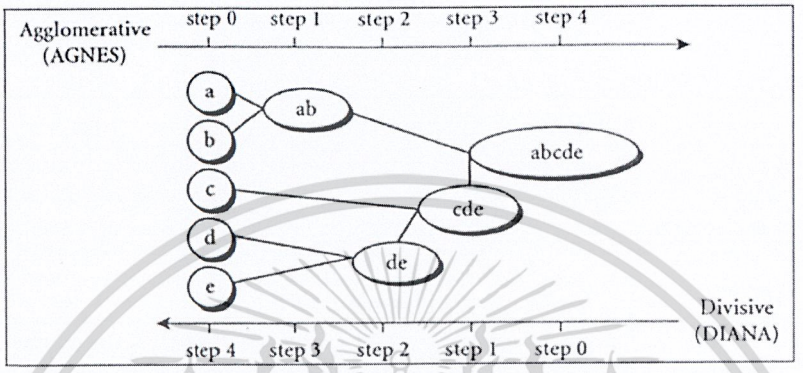
รูปที่ 2.5 ข้อมูลตัวอย่างที่ประกอบด้วย 3 คลัสเตอร์ (Han and Kamber, 2006, p.26)

2.3.1 ประเภทของขั้นตอนวิธีการจัดแบ่งกลุ่มข้อมูล

1) Partition Method การจัดกลุ่มข้อมูลประเภทนี้จะทำการสร้าง k พาร์ทิชันบนฐานข้อมูลจำนวน n เรคคอร์ด โดยแต่ละพาร์ทิชันจะแสดงถึงข้อมูลที่ถูกแบ่งออกเป็นกลุ่ม ในแต่ละกลุ่มจะประกอบไปด้วยข้อมูลอย่างน้อยที่สุด 1 แถว และข้อมูลแต่ละแถวจะต้องถูกจัดให้อยู่ในกลุ่มข้อมูลเพียงกลุ่มเดียวเท่านั้น (สำหรับบางเทคนิคอาจอนุโลมให้ข้อมูลใดๆสามารถถูกจัดอยู่ในกลุ่มข้อมูลได้มากกว่า 1 กลุ่ม) ตัวอย่างเทคนิคของการจัดกลุ่มแบบนี้ได้แก่ k-means algorithm, k-medoids algorithm, CLARA

(Clustering LARge Application), CLARANS(Clustering LARge Application based upon RANdomized Search)

2) Hierarchical Method การจัดกลุ่มข้อมูลประเภทนี้จะอาศัยหลักการแบ่งข้อมูลออกเป็นลำดับชั้นคล้ายกับต้นไม้ ซึ่งวิธีการแบ่งกลุ่มข้อมูลแบบนี้สามารถแบ่งออกเป็น 2 แนวทางตามลักษณะการสร้างลำดับชั้นคือ Agglomerative approach กับ Divisive approach



รูปที่ 2.6 การจัดกลุ่มข้อมูลโดยใช้ AGNES และ DIANA (Han and Kamber, 2006, p.409)

ตัวอย่างเทคนิคของการจัดกลุ่มแบบนี้ได้แก่ AGNES (Agglomerative NESting) ซึ่งจะเป็น agglomerative hierarchical clustering, DIANA (Divisive ANAlysis) ซึ่งจะเป็น divisive hierarchical clustering, BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies), CURE (Clustering Using REpresentatives)

3) Density-Based Method การจัดกลุ่มข้อมูลประเภทนี้จะพิจารณาความหนาแน่นของข้อมูลเป็นเกณฑ์ในการค้นหาคลัสเตอร์ หลักการทั่วไปของเทคนิคนี้คือการแผ่ขยายขอบเขตของคลัสเตอร์ไปเรื่อยๆ ตราบใดที่ความหนาแน่นของข้อมูลยังมีค่าน้อยกว่าหรือเท่ากับค่าที่ผู้ใช้กำหนด นั่นคือแต่ละข้อมูลของคลัสเตอร์ใดๆ จะต้องประกอบด้วยข้อมูลซึ่งอยู่ใกล้กันภายในรัศมีที่กำหนด (neighborhood) ด้วย เทคนิคนี้สามารถใช้ในการกรองข้อมูลรบกวน (noisy) ซึ่งเป็นข้อมูลที่มีความหนาแน่นเบาบางได้ และยังสามารถค้นหาคลัสเตอร์ที่มีรูปทรงซับซ้อนได้อีกด้วย ตัวอย่างเทคนิคของการจัดกลุ่มแบบนี้ได้แก่ DBSCAN

2.3.2 ขั้นตอนวิธีการจัดกลุ่ม k-means

ขั้นตอนวิธีการจัดกลุ่ม k-means เป็นเทคนิคหนึ่งที่จัดอยู่ในประเภท Partition Method มีการใช้ค่าเฉลี่ยของข้อมูลที่ถูกจัดให้อยู่ในคลัสเตอร์เดียวกันเป็นตัวแทนของทุกข้อมูลในคลัสเตอร์นั้น ขั้นตอนวิธีเริ่มต้นจากการรับค่าพารามิเตอร์ k ซึ่งค่านี้คือจำนวนคลัสเตอร์ที่ต้องการค้นหา จากนั้นขั้นตอนวิธีจะทำการสุ่มเลือกข้อมูลเริ่มต้นจำนวน k ชุด ซึ่งแต่ละชุดที่ได้มานั้นจะเป็นจุดศูนย์กลางเริ่มต้นของแต่ละคลัสเตอร์ (centroid) จากนั้นทำการจัดกลุ่มให้กับข้อมูลที่เหลือ ข้อมูลจะถูกจัดให้อยู่ในคลัสเตอร์เดียวกันเมื่อข้อมูลนั้นมีความคล้ายกับตัวแทนของคลัสเตอร์นั้นมากที่สุด จากนั้นจึงทำการคำนวณหา

ค่าเฉลี่ยของคลัสเตอร์ใหม่ และดำเนินกระบวนการเดียวกันกับข้อมูลที่เหลือต่อไป จนกระทั่งทุกข้อมูลถูกจัดกลุ่มอย่างสมบูรณ์และข้อมูลไม่มีการเปลี่ยนกลุ่มอีกต่อไป

Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the object in the cluster.

Input: - *k*, the number of cluster;
- *D*, a data set containing *n* objects;

Output: A set of *k* clusters.

Method:

(1) arbitrarily choose *k* objects form *D* as the initial cluster center;
(2) **repeat**
(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
(5) **until** no change;

รูปที่ 2.7 ขั้นตอนวิธี k-means (Han and Kamber, 2006, p.403)

การทำงานของ k-means จะมีประสิทธิภาพสูงก็ต่อเมื่อข้อมูลเกาะกลุ่มกันหนาแน่น แต่ละกลุ่มแยกจากกันอย่างชัดเจน และความหนาแน่นของข้อมูลในแต่ละกลุ่มใกล้เคียงกัน

จุดเด่นของ k-means คือง่ายและสามารถใช้ได้กับข้อมูลหลายประเภท และยังมีประสิทธิภาพในด้านความเร็ว แต่จุดด้อยของ k-means ก็พบว่ายังไม่เหมาะสมกับข้อมูลทุกประเภท และไม่สามารถจัดการกลุ่มที่มีรูปร่างไม่เป็นรูปทรงกลมหรือกลุ่มที่มีขนาดหรือความหนาแน่นแตกต่างกันได้ นอกจากนี้ k-means ยังถูกจำกัดสำหรับข้อมูลที่มีตัวแทนข้อมูลที่คลุมเครือหรือไม่ชัดเจน

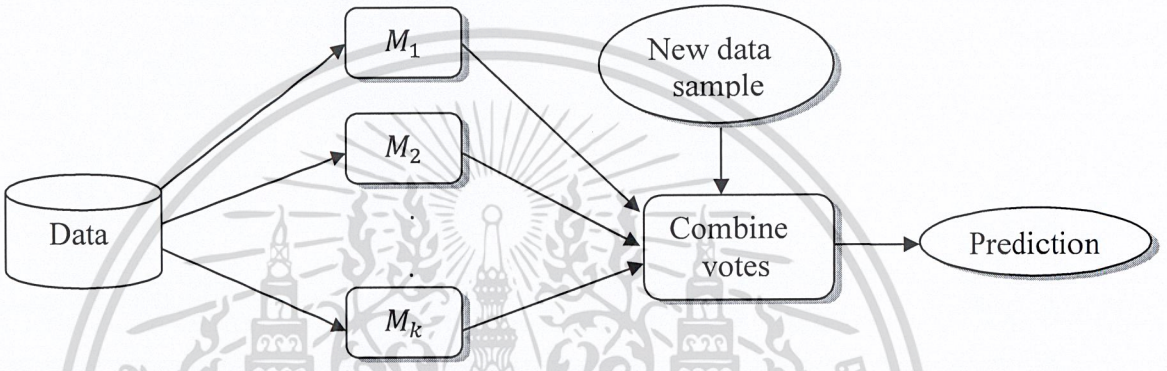
2.4 การรวมกันของตัวจำแนกประเภท (Combining classifier)

ตัวจำแนกประเภท (Classifier) ทั่วไปนั้นจะให้ผลลัพธ์โดยการนำข้อมูลมาผ่านขั้นตอนวิธีการตัดสินใจของตัวจำแนกประเภทนั้นๆ ซึ่งแน่นอนว่าคำตอบนั้นจะได้มาด้วยการทำงานเพียงกระบวนการเดียว ทำให้ความถูกต้องและความแม่นยำโดยเฉลี่ยมีค่าไม่สูงมากนัก (ยกเว้นกรณีที่ข้อมูลมีรูปแบบที่เรียบง่าย และแต่ละกลุ่มแยกจากกันอย่างชัดเจน) ด้วยเหตุนี้จึงมีงานวิจัยจำนวนมากที่ได้คิดค้นวิธีการที่จะปรับปรุงขั้นตอนวิธีใหม่แทนที่จะใช้ตัวจำแนกประเภทเพียงเทคนิคเดียว ก็จะหันมาใช้หลายๆเทคนิคหรือใช้หลายๆโมเดลจำแนกประเภทเข้ามาทำงานร่วมกันแทน หรือเรียกว่า "การรวมกันของตัวจำแนกประเภท"

การรวมกันของตัวจำแนกประเภท (Combining classifier) สามารถแบ่งออกเป็นกลุ่มใหญ่ได้ 2 ประเภท ดังนี้

2.4.1 การรวมกันของตัวจำแนกประเภทเดียวกัน (Combining homogeneous classifier)

วิธีการนี้จะใช้โมเดลจำแนกประเภทหลายๆ โมเดล ซึ่งแต่ละโมเดลจะใช้ขั้นตอนวิธีเดียวกันในการสร้างโมเดล เช่นถ้าเลือกใช้ขั้นตอนวิธีตัดไม้ตัดสินใจ C4.5 ตัวจำแนกประเภททุกตัวก็จะใช้ขั้นตอนวิธีตัดไม้ตัดสินใจ C4.5 ทั้งหมด ส่วนที่แตกต่างกันคือ ข้อมูลเรียนรู้ (Training Data) ที่จะถูกแบ่งให้กับตัวจำแนกประเภทแต่ละแบบ ในท้ายสุดเราจะได้โมเดลจำแนกประเภททั้งหมด k แบบ ($M_1, M_2, \ldots, M_k$) ดังรูปที่ 2.8



รูปที่ 2.8 โครงสร้างการรวมกันของตัวจำแนกประเภทเดียวกัน (Han and Kamber, 2006, p.366)

ในการทำนายข้อมูลใหม่นั้น (New data sample) จะใช้การโหวตเสียงข้างมาก เพื่อทำนายกลุ่มออกมา ตัวอย่างของวิธีการรวมกันของตัวจำแนกประเภทเดียวกัน ได้แก่ขั้นตอนวิธี Bagging และ Boosting

2.4.1.1 ขั้นตอนวิธี Bagging คือการสร้างโมเดลจำแนกประเภทหลายโมเดล ด้วยชุดข้อมูลเรียนรู้ที่แตกต่างกัน แต่จะใช้เทคนิคในการสร้างโมเดลด้วยอัลกอริทึมเดียวกัน ซึ่งจะช่วยปรับปรุงประสิทธิภาพในการทำนายข้อมูลทั้งในปัญหาการจำแนกประเภท และการประมาณค่าได้ ขั้นตอนวิธีของ Bagging แสดงดังรูปที่ 2.9

แต่ละโมเดลจำแนกประเภท M_i จะถูกสอนด้วยชุดข้อมูลเรียนรู้ D_i

คำตอบสุดท้ายของการทำนายนั้น วิธี Bagging จะนับผลโหวตที่ได้จากโมเดลจำแนกประเภทที่มีทั้งหมด และกำหนดกลุ่มด้วยผลโหวตที่มากที่สุดให้กับข้อมูลใหม่

Algorithm: **Bagging**. The bagging algorithm – create an ensemble of models (classifiers or predictors) for a learning scheme where each model gives an equally-weight prediction

Input: - D, a set of d training tuple;
- k, the number of models in the ensemble;
- a learning scheme (e.g., decision tree algorithm, backpropagation, etc.)

Output: A composite model, M^*.

Method:

(1) **for** $i = 1$ to k **do** // create k models:
(2) create bootstrap sample, D_i, by sampling D with replacement
(3) use D_i to derive a model, M_i;
(4) **end for**

To use the composite model on a tuple, X:

(1) **if** classification **then**
(2) let each of the k models classify X and return the majority vote;
(3) **if** prediction **then**
(4) let each of the k models predict a value for X and return the average predicted value;

รูปที่ 2.9 ขั้นตอนวิธี Bagging (Han and Kamber, 2006, p.367)

2.4.1.2 ขั้นตอนวิธี Boosting คือการใช้ขั้นตอนวิธีการเรียนรู้หลายๆโมเดลจำแนกประเภทในการตัดสินใจ โดยจะใช้การถ่วงน้ำหนักเข้ามาให้แต่ละโมเดลจำแนกประเภท ซึ่งค่าน้ำหนักนั้นจะได้มาจากความแม่นยำบนข้อมูลเรียนรู้ และสำหรับคำตอบสุดท้ายของการทำนายนั้น วิธี Boosting จะใช้การโหวตแบบถ่วงน้ำหนัก เพื่อกำหนดกลุ่มให้กับข้อมูลใหม่

ขั้นตอนวิธีที่ได้รับความนิยมอย่างมากของ Boosting คือ “Adaboost” ขั้นตอนวิธีของ Adaboost แสดงดังรูปที่ 2.10

Algorithm: **Adaboost**. A boosting algorithm—creates an ensemble of classifiers. Each one gives a weighted vote

Input: - D, a set of d class-labeled training tuples;
- k, the number of rounds (one classifier is generated per round);
- a classification learning scheme.

Output: A composite model.

Method:

(1) initialize the weight of each tuple in D to $1/d$;
(2) **for** i = 1 to k **do** // for each round:
(3) sample D with replacement according to the tuple weights to obtain D_i ;
(4) use training set D_i to derive a model, M_i ;
(5) compute error(M_i), the error rate of M_i (Equation 6.66)
(6) **if** error(M_i) > 0.5 **then**
(7) reinitialize the weights to $1/d$
(8) go back to step 3 and try again;
(9) **end if**
(10) for each tuple in D_i that was correctly classified do
(11) multiply the weight of the tuple by $error(M_i)/(1 - error(M_i))$; // update weights
(12) normalize the weight of each tuple;
(13) **end for**

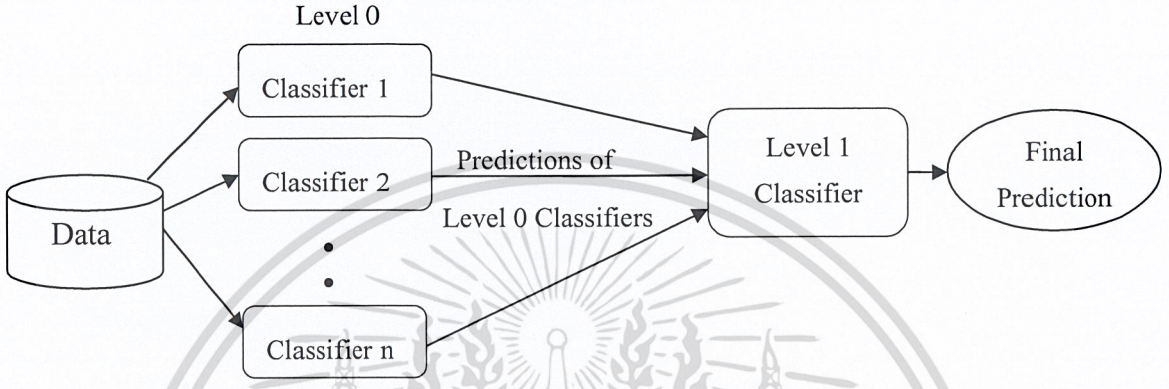
To use the composite model on a tuple, X:

(1) initialize weight of each class to 0;
(2) **for** i = 1 to k **do** // for each classifier:
(3) $w_i = \log \frac{1-error(M_i)}{error(M_i)}$; // weight of the classifier's vote
(4) $c = M_i(X)$; // get class prediction for X from M_i
(5) add w_i to weight for class c
(6) **end for**
(7) return the class with the largest weight;

รูปที่ 2.10 ขั้นตอนวิธี Adaboost (Han and Kamber, 2006, p.369)

2.4.2 การรวมกันของตัวจำแนกประเภทที่แตกต่างกัน (Combining heterogeneous classifier)

วิธีการนี้จะใช้ตัวจำแนกประเภทหลายตัว ซึ่งแต่ละตัวจะใช้ขั้นตอนวิธีที่ต่างกันในการสร้างโมเดล การทำงานนั้นตัวโมเดลจำแนกประเภทจะถูกแบ่งออกเป็นระดับ (level) โดยที่ระดับสูงสุดจะเป็นตัวตัดสินใจผลลัพธ์สุดท้ายให้กับข้อมูลใหม่ ซึ่งการตัดสินใจนั้นจะอาศัยผลจากการทำนายของโมเดลจำแนกประเภทก่อนหน้าทั้งหมด แสดงได้ดังรูปที่ 2.11



รูปที่ 2.11 การทำงานของการรวมกันของตัวจำแนกประเภทที่แตกต่างกัน

ขั้นตอนการทำงานจะแบ่งออกเป็น 2 ขั้นตอน

ขั้นตอนการเรียนรู้ (Training Phase) ในระยะนี้นั้นจะมีการให้แต่ละตัวจำแนกประเภทระดับ 0 เรียนรู้โดยใช้เทคนิค (leave-one-out cross validation) แล้วจึงสร้างเมทริกซ์ เพื่อจัดเก็บผลทำนาย จากตัวจำแนกประเภทระดับ 0 ซึ่งจะมีขนาด n+1 แถว i คอลัมน์ โดย n คือจำนวนโมเดลจำแนกประเภทในระดับ 0 และอีก 1 ค่าคือกลุ่มที่แท้จริง (actual class) ของข้อมูล ส่วน i คือจำนวนข้อมูลที่ใช้ในการเรียนรู้ ต่อมาจึงให้ตัวจำแนกประเภทระดับ 1 เรียนรู้โดยใช้เมทริกซ์ที่ได้มาจากก่อนหน้า สุดท้ายให้แต่ละตัวจำแนกประเภทระดับ 0 เรียนรู้อีกครั้งโดยใช้ข้อมูลเรียนรู้ทั้งหมด

ขั้นตอนประยุกต์ใช้งาน (Application Phase) ในระยะนี้จะใช้ในการจำแนกกลุ่มตัวอย่างใหม่ โดยเริ่มต้นจะใช้ตัวจำแนกประเภทระดับ 0 ทั้งหมดแล้วจึงเก็บผลลัพธ์ที่ได้ในเวกเตอร์ ต่อมาเวกเตอร์นี้จะเป็นส่วนนำเข้าให้กับตัวจำแนกข้อมูลระดับ 1 เพื่อทำนายผลลัพธ์สุดท้ายให้กับตัวอย่างใหม่

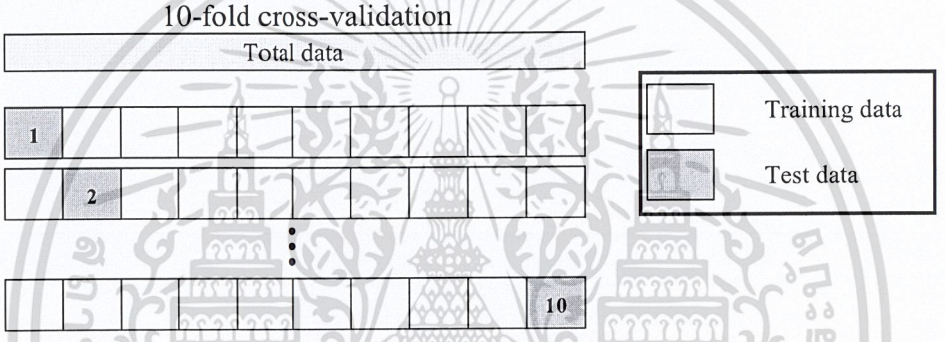
การรวมกันของตัวจำแนกประเภทที่ต่างกันสามารถนำมาใช้ในกรณีที่จำนวนตัวอย่างในชุดข้อมูลมีจำนวนน้อย ซึ่งแตกต่างกับการทำงานของการรวมกันของตัวจำแนกประเภทเดียวกันที่ต้องการความหลากหลายของตัวอย่างมาช่วยในการสร้างโมเดล

2.5 การวัดประสิทธิภาพ (Performance Evaluation Measurement)

2.5.1 k-fold cross-validation

เทคนิค k-fold cross-validation (Ron, 1995) เป็นวิธีการวัดประสิทธิภาพในการทำนายตัวอย่างของโมเดล โดยพื้นฐานของเทคนิคนี้คือการสุ่มตัวอย่าง (resampling) โดยเริ่มจากแบ่งชุดข้อมูลออกเป็นส่วนๆหรือเรียกว่า fold และนำบางส่วนจากชุดข้อมูลนั้นมาทดสอบผลลัพธ์จากการทำนายข้อมูลทดสอบของโมเดล

กรณีการเลือกสุ่มข้อมูลแบบความเที่ยงตรง k กลุ่ม เราจะแบ่งข้อมูลออกเป็น k ชุดเท่าๆกัน และทำการคำนวณค่าความแม่นยำจากการทำนาย k รอบ โดยแต่ละรอบจะมีการสร้างโมเดลจำแนกประเภทหนึ่งตัว จากข้อมูลเรียนรู้ k-1 ชุด และใช้ข้อมูลทดสอบ 1 ชุด (ชุดที่ไม่ได้นำมาเรียนรู้)



รูปที่ 2.12 10-fold cross-validation

จากรูปที่ 2.12 ในการทำงานรอบแรก ข้อมูลในชุดที่ 1 จะใช้เป็นข้อมูลทดสอบ ส่วนข้อมูลในชุดที่ 2 ถึง 10 จะนำมาใช้เป็นชุดข้อมูลสำหรับการเรียนรู้ ซึ่งจะได้โมเดลจำแนกประเภท 1 ตัว ต่อมารอบที่สอง ก็จะใช้ข้อมูลในชุดที่ 2 เป็นข้อมูลทดสอบ ส่วนข้อมูลในชุดที่ 1 และ 3 ถึง 10 จะนำมาใช้เป็นชุดข้อมูลสำหรับการเรียนรู้ ซึ่งจะได้โมเดลจำแนกประเภทอีก 1 ตัว จะมีการทำงานลักษณะนี้ไปเรื่อยๆ จนถึงรอบที่สิบ จะใช้ข้อมูลในชุดที่ 10 เป็นชุดข้อมูลทดสอบ ส่วนข้อมูลในชุดที่ 1 ถึง 9 จะนำมาใช้เป็นชุดข้อมูลสำหรับการเรียนรู้ และจะได้โมเดลจำแนกประเภทอีก 1 ตัว

2.5.2 มาตรวัดประสิทธิภาพของโมเดล

ในการวัดประสิทธิภาพของโมเดลจำแนกประเภทข้อมูลนั้น จะอาศัย Confusion Matrix ในการเก็บข้อมูลจำนวนแถวที่จำแนกจากกลุ่มข้อมูลจริงและกลุ่มข้อมูลจากการทำนาย โดยที่ตารางนั้นจะมีขนาด $m \times m$ โดยที่ m คือจำนวนของกลุ่ม

ตารางที่ 2.2 Confusion Matrix

		Predicted class	
Actual class	Class	C_1	C_2
	C_1	TP	FN
	C_2	FP	TN

ค่าต่างๆ ภายใน Confusion Matrix มีความหมายดังนี้

- True positive (TP) คือจำนวนข้อมูลที่โมเดลจำแนกกลุ่มเป็น C_1 และคำตอบเป็น C_1
- True negative (TN) คือจำนวนข้อมูลที่โมเดลจำแนกกลุ่มเป็น C_2 และคำตอบเป็น C_2
- False positive (FP) คือจำนวนข้อมูลที่โมเดลจำแนกกลุ่มเป็น C_1 แต่คำตอบเป็น C_2
- False negative (FN) คือจำนวนข้อมูลที่โมเดลจำแนกกลุ่มเป็น C_2 แต่คำตอบเป็น C_1

ตารางที่ 2.3 Confusion Matrix ของการจำแนกประเภทการซื้อคอมพิวเตอร์ (Han and Kamber, 2006, p.360)

Classes	buy_computer = yes	buy_computer = no	Total	Accuracy (%)
buy_computer = yes	6,954	46	7,000	99.34
buy_computer = no	412	2,588	3,000	86.27
Total	7,366	2,634	10,000	95.37

จากตารางที่ 2.3 ชุดข้อมูลการซื้อคอมพิวเตอร์มีจำนวนข้อมูล 10,000 ตัวอย่าง จำแนกออกเป็น 2 กลุ่มคือ buy_computer = yes จำนวน 7,000 ตัวอย่าง และ buy_computer = no จำนวน 3,000 ตัวอย่าง จากการจำแนกประเภทพบว่ามีจำนวนข้อมูลที่โมเดลจำแนกกลุ่มเป็น buy_computer = yes และคำตอบเป็น buy_computer = yes เท่ากับ 6,954 ตัวอย่าง (True positive = 6,954) มีจำนวนข้อมูลที่โมเดลจำแนกกลุ่มเป็น buy_computer = no และคำตอบเป็น buy_computer = no เท่ากับ 2,588 ตัวอย่าง (True negative = 2,588) มีจำนวนข้อมูลที่โมเดลจำแนกกลุ่มเป็น buy_computer = yes แต่คำตอบเป็น buy_computer = no เท่ากับ 412 ตัวอย่าง (False positive = 412) มีจำนวนข้อมูลที่โมเดลจำแนกกลุ่มเป็น buy_computer = no แต่คำตอบเป็น buy_computer = yes เท่ากับ 46 ตัวอย่าง (False negative = 46) คำนวณค่าความแม่นยำการจำแนกประเภทกลุ่ม buy_computer = yes ได้เท่ากับ 99.34% ค่าความแม่นยำการจำแนกประเภทกลุ่ม buy_computer = no ได้เท่ากับ 86.27% และค่าความแม่นยำการจำแนกประเภททั้งหมดได้เท่ากับ 95.37%

ค่าความแม่นยำ (Accuracy) คืออัตราการทำนายถูกต้อง มีสูตรในการคำนวณ คือ

$$accuracy = \frac{TN + TP}{TN + FP + FN + TP} \quad (2.6)$$

2.6 งานวิจัยที่เกี่ยวข้อง

2.6.1 งานวิจัยเกี่ยวกับต้นไม้ตัดสินใจ

เป็นงานหนึ่งที่ได้รับการศึกษาค้นคว้า และมีผู้ให้ความสนใจเป็นจำนวนมาก ทั้งในเรื่องของการพัฒนาเทคนิคใหม่เพื่อปรับปรุงประสิทธิภาพของวิธีการต้นไม้ตัดสินใจที่มีอยู่ (Weizhao Guo and Jian Yin, 2009) การศึกษาเปรียบเทียบวิธีการต้นไม้ตัดสินใจ (My Chau Tu, GDongil Shin and Dong Kyoo Shin, 2009) และนำวิธีการต้นไม้ตัดสินใจสร้างคุณค่าให้กับธุรกิจ (Rong Cao and Lizhen Xu, 2009)

งานวิจัยเกี่ยวกับการพัฒนาเทคนิคใหม่เพื่อปรับปรุงประสิทธิภาพของวิธีการต้นไม้ตัดสินใจ เป็นงานวิจัยที่มองเห็นถึงจุดอ่อนของวิธีการต้นไม้ตัดสินใจที่มีอยู่ ไม่ว่าจะเป็น ID3 (Ross Quinlan, 1986), C4.5 (Ross Quinlan, 1993), CART(L.breiman, J.Friedman, R.Olshen, and C.Stone, 1984) แล้วทำการปรับปรุงจุดอ่อนที่พบด้วยการเพิ่มเติมการทำงานเพื่อให้วิธีการต่างๆ สามารถให้ผลลัพธ์ที่มีประสิทธิภาพมากขึ้น ตัวอย่างของงานวิจัยได้แก่

วินิจฉัยการปรับปรุงต้นไม้ตัดสินใจบนพื้นฐานการให้น้ำหนัก (Weizhao Guo, Li Huang, Zhimin Yang, Xiaobo Yang and Li Huang, 2009) จากงานวิจัยได้พบปัญหาว่า วิธีการต้นไม้ตัดสินใจส่วนใหญ่กำหนดการกระจายของตัวแทนข้อมูลที่แตกต่างกันมีค่าเท่ากัน ซึ่งความเป็นจริงแล้วไม่ควรที่จะทำแบบนั้น ตัวอย่างเช่น ถ้ามีกลุ่มที่มีข้อมูลจำนวนมากกับกลุ่มที่มีข้อมูลจำนวนน้อยอยู่ 2 กลุ่ม เมื่อคำนวณความแม่นยำจะพบว่า กลุ่มที่มีข้อมูลจำนวนมากจะสามารถทำนายได้อย่างถูกต้อง แต่กลุ่มที่มีข้อมูลจำนวนน้อยจะเกิดความผิดพลาดขึ้น ซึ่งบางทีกลุ่มข้อมูลจำนวนน้อยอาจจะเป็นส่วนที่สำคัญในการจำแนกกลุ่มประเภท และทำนายข้อมูลในอนาคตได้ เพื่อแก้ปัญหาที่กล่าวมา งานวิจัยจึงได้ใช้วิธีการที่ง่าย และมีประสิทธิภาพ โดยการใช้การถ่วงน้ำหนักบนต้นไม้ตัดสินใจ ซึ่งใช้ความรู้พื้นฐานจาก Cost-Sensitive Learning มาช่วยในการปรับปรุงต้นไม้ตัดสินใจ C4.5

วิธีการการปรับปรุงตัดสินใจนั้นเริ่มต้นกำหนดให้กลุ่มมีทั้งหมด n กลุ่ม ได้แก่ $C_1, C_2,..., C_n$ ข้อมูลเรียนรู้มีทั้งหมด m แถว ซึ่ง m_i คือจำนวนข้อมูลเรียนรู้ที่อยู่ในกลุ่ม C_i และจะได้ว่า $\sum_i^n m_i = m$ จากนั้นกำหนดให้ $W = \{w_1, w_2,..., w_m\}$ ซึ่งเป็นเซตของน้ำหนักการกระจายของกลุ่มข้อมูลในแต่ละแถวข้อมูลเรียนรู้ เมื่อกำหนดให้ข้อมูลแถวที่ k อยู่ในกลุ่มของ C_r และข้อมูลแถวที่ t อยู่ในกลุ่มของ C_s โดยที่ $k, t = 1,2,\ldots,m$ และ $r, s = 1,2,\ldots,n$ ดังนั้นค่าน้ำหนักของข้อมูลจะมีเงื่อนไขในการกำหนด 2 เงื่อนไข ได้แก่

(ก): $0 \leq w_j \leq 1$

(ข): if $m_r \leq m_s$, then $w_k \leq w_t$

โดยที่เงื่อนไข 2 ข้อดังกล่าวจะถูกนำมาใช้ในการปรับปรุงการคำนวณการเลือกคุณลักษณะสร้างต้นไม้ตัดสินใจ จากนั้นกำหนดให้ข้อมูลแถวที่ j อยู่ในกลุ่มของ C_i ดังนั้นค่าน้ำหนัก w_j ของข้อมูลแถวที่ j สามารถเขียนในรูปสมการที่ 2.7

$$w_j = \frac{m_i}{m} \tag{2.7}$$

หลังจากนั้นทำการปรับปรุงค่า information gain เป็น weight information gain (WIG) ซึ่งเป็นค่าเกนถ่วงน้ำหนักการกระจายของกลุ่มข้อมูล สามารถเขียนได้ในรูปสมการที่ 2.8

$$\mathrm{WIG(A,T)=WIE(T)} - \sum_{i}^{k} \left(\frac{\sum_{x \in S_i} w_x}{\sum_{j=1}^{m} w_j} \right) \times \mathrm{WIE}(T_i) \tag{2.8}$$

โดยที่ WIG(A,T) เป็นค่าเกนถ่วงน้ำหนักการกระจายของกลุ่มข้อมูลของคุณลักษณะ A

k เป็นค่าที่เป็นไปได้ของคุณลักษณะ A

S_i เป็นเซตของข้อมูลของคุณลักษณะ A ที่มีค่า i (2.9)

จากนั้นค่า weight information entropy (WIE) สามารถเขียนได้ในรูปสมการที่ 2.9

$$\mathrm{WIE(A,T)} = - \sum_{i}^{n} \left(\frac{\sum_{x \in c_i} w_x}{\sum_{j=1}^{m} w_j} \right) \times \log_2 \left(\frac{\sum_{x \in c_i} w_x}{\sum_{j=1}^{m} w_j} \right)$$

สุดท้ายค่ามาตรฐานอัตราส่วนเกน (Gain Ratio) ถูกปรับปรุงค่าด้วย weight gain ratio (WGR) สามารถเขียนได้ในรูปสมการที่ 2.10

$$\mathrm{WGR(A,T)} = \frac{WIG(A,T)}{WSI(A,T)} \tag{2.10}$$

โดยที่ WSI(A,T) หมายถึง weight splitting information (WSI) ซึ่งปรับปรุงมาจากค่าสารสนเทศของการแบ่งแยก สามารถเขียนได้ในรูปสมการที่ 2.11

$$\mathrm{WSI(A,T)} = - \sum_{i}^{k} \left(\frac{\sum_{x \in S_i} w_x}{\sum_{j=1}^{m} w_j} \right) \times \log_2 \left(\frac{\sum_{x \in S_i} w_x}{\sum_{j=1}^{m} w_j} \right) \tag{2.11}$$

ค่า k คือค่าที่เป็นไปได้ของคุณลักษณะ A ซึ่งการคำนวณหาคุณลักษณะที่เหมาะสมนั้นจะใช้ขั้นตอนวิธีเดิม แต่จะเลือกคุณลักษณะที่มีค่า WGR สูงที่สุดแทนที่จะใช้ค่ามาตรฐานอัตราส่วนเกน

จากผลการทดลองการปรับปรุงวิธีการ C4.5 นั้นพบว่า วิธีการที่ได้จากการปรับปรุงนั้นให้ค่าความแม่นยำในการจำแนกประเภทสูงกว่าเดิม

งานวิจัยเกี่ยวกับการศึกษาเปรียบเทียบวิธีการต้นไม้ตัดสินใจ เป็นงานวิจัยที่สนใจว่า ข้อมูลที่มีอยู่นั้น โมเดลจำแนกประเภทใดจะสามารถให้ค่าของมาตรวัดต่างๆ ที่ดีกว่ากัน โดยปกตินั้นข้อมูลที่ถูกนำมาใช้ในงานวิจัยแบบนี้มักจะเป็นข้อมูลเฉพาะทาง เช่น ข้อมูลทางการแพทย์ ข้อมูลทางอิเล็กทรอนิกส์ ข้อมูลทางการทหารหรืออื่นๆ ตัวอย่างของงานวิจัย ได้แก่

การศึกษาเปรียบเทียบกระบวนการจำแนกประเภทข้อมูลทางการแพทย์ด้วยต้นไม้ตัดสินใจ และขั้นตอนวิธีแบ็กกิ้ง (Bagging) (My Chau Tu, Dongil Shian and Dongkyoo Shin, 2009) งานวิจัยนี้เกิดขึ้นเพราะความต้องการที่จะพัฒนาระบบช่วยตัดสินใจทางการแพทย์ สำหรับให้แพทย์ใช้งานในการหาทางรักษาผู้ป่วยที่เป็นโรคหัวใจ ดังนั้นจึงต้องมีโมเดลจำแนกประเภทผู้ป่วยที่ให้ความแม่นยำที่สุด ในงานวิจัยนั้นได้เลือกวิธีการต้นไม้ตัดสินใจ C4.5 วิธีการต้นไม้ตัดสินใจ C4.5 แบบ bagging และ Naïve Bayes แบบ bagging ให้ค่ามาตรวัดต่างๆ ดีที่สุด ซึ่งได้แก่ precision, recall, f-measure, TPR และ FPR

งานวิจัยเกี่ยวกับการนำวิธีการต้นไม้ตัดสินใจสร้างคุณค่าให้กับธุรกิจ เป็นงานวิจัยที่นำวิธีการต้นไม้ตัดสินใจมาประยุกต์ใช้ในทางธุรกิจ ไม่ว่าจะเป็นการวิเคราะห์การขาย การประเมินลูกค้า การวิเคราะห์ตลาดและอื่นๆ ซึ่งข้อมูลที่ได้จากงานวิจัยนั้นจะถูกนำไปใช้กับระบบงานจริงเพื่อสร้างความเจริญเติบโตให้แก่บริษัท

2.6.2 งานวิจัยที่เกี่ยวข้องกับการปรับปรุงการจัดกลุ่ม

งานวิจัยที่เกี่ยวข้องกับการจัดกลุ่ม เป็นอีกงานหนึ่งที่ได้รับการศึกษาค้นคว้าและมีผู้ให้ความสนใจเป็นจำนวนมาก ทั้งในเรื่องของการพัฒนาเทคนิคใหม่เพื่อปรับปรุงประสิทธิภาพของวิธีการจัดกลุ่มที่มีอยู่ (Jian Zhu and Hanshi Wans, 2010; Xiaoping Qin, Shijne Zheng, Tingting He, Ning Zou and Ying Huans, 2010) และการศึกษาเปรียบเทียบการจัดกลุ่ม (Pawesh kumar and Siri Krishan Wasan, 2010)

งานวิจัยเกี่ยวกับการพัฒนาเทคนิคใหม่เพื่อปรับปรุงประสิทธิภาพของการจัดกลุ่มที่มีอยู่ เป็นงานวิจัยที่สนใจเกี่ยวกับการเพิ่มประสิทธิภาพของวิธีการจัดกลุ่มที่มีอยู่ ให้ทำงานได้ดีขึ้น หรือลดเวลาในการจัดกลุ่มให้น้อยลง และความแม่นยำในการทำงานสูญเสียไปเล็กน้อย ซึ่งส่งผลดีในงานที่เวลาเป็นสิ่งสำคัญ ขณะเดียวกันก็ต้องการผลลัพธ์ที่ดีด้วย ตัวอย่างของงานวิจัย ได้แก่

ขั้นตอนการจัดกลุ่ม k-means ที่ได้รับการพัฒนา (Jian Zhu and Hanshi Wang, 2010) จากงานวิจัยได้พบปัญหาของขั้นตอนการจัดกลุ่ม k-means ได้แก่ ปัญหาในการเลือกจุดเริ่มต้นสร้างคลัสเตอร์ไม่มีอะไรบ่งชี้ได้ว่า ค่า k เท่าไหร่จึงจะได้ผลลัพธ์ดีที่สุด ขั้นตอนวิธีมีความอ่อนไหวกับค่าแปลกแยกสุดท้ายบางครั้งผลลัพธ์ที่ได้จากการจัดกลุ่มไม่ดีนัก ในงานวิจัยเลือกที่จะพัฒนาการเลือกค่า k เพื่อให้ได้ค่าที่เหมาะสมที่สุดโดยใช้ขั้นตอนวิธีเชิงพันธุกรรม (genetic algorithm) จากการทดลองพบว่า การทำงานทำได้ดีเพียงบางชุดข้อมูลเท่านั้น บางชุดข้อมูลยังมีความผิดพลาดไม่ชัดเจน แต่อย่างไรก็ตามก็ทำให้จำนวนจุดศูนย์กลางถูกคำนวณอย่างยุติธรรม มีความหมายชัดเจน ซึ่งเป็นแนวทางที่ดีที่จะพัฒนาต่อไป

ขั้นตอนวิธีการ k-means ที่มีประสิทธิภาพและการประยุกต์ในระบบ CRM (Xiaoping Qin, Shijue Zheng, Tingting He, Ming Zou and Yins Huang, 2010) จากงานวิจัยพบว่า เมื่อต้องการจัดกลุ่มลูกค้าที่มีข้อมูลจำนวนมหาศาลทำให้การทำงานของขั้นตอนวิธีทำงานได้ช้าลง รวมถึงประสิทธิภาพก็ต่ำลง แต่ข้อมูลลูกค้าไม่สามารถที่จะลดบั่นทอนลงไปได้ ดังนั้นจึงได้คิดปรับปรุงการทำงานของการจัดกลุ่ม k-means ให้สามารถจัดการกับข้อมูลมหาศาลได้ดีขึ้น โดยนำทฤษฎีอสมการสามเหลี่ยม (Triangle inequality) มาใช้งาน

จากการทดลองพบว่า การจัดกลุ่มสามารถทำงานได้รวดเร็วขึ้น และยิ่งเพิ่มทวีอัตราการลดลงเมื่อข้อมูลมีมากขึ้น ทำให้ปัญหาการจัดกลุ่มลูกค้าลดลง และนำมาใช้งานในระบบ CRM ได้อย่างมีประสิทธิภาพ

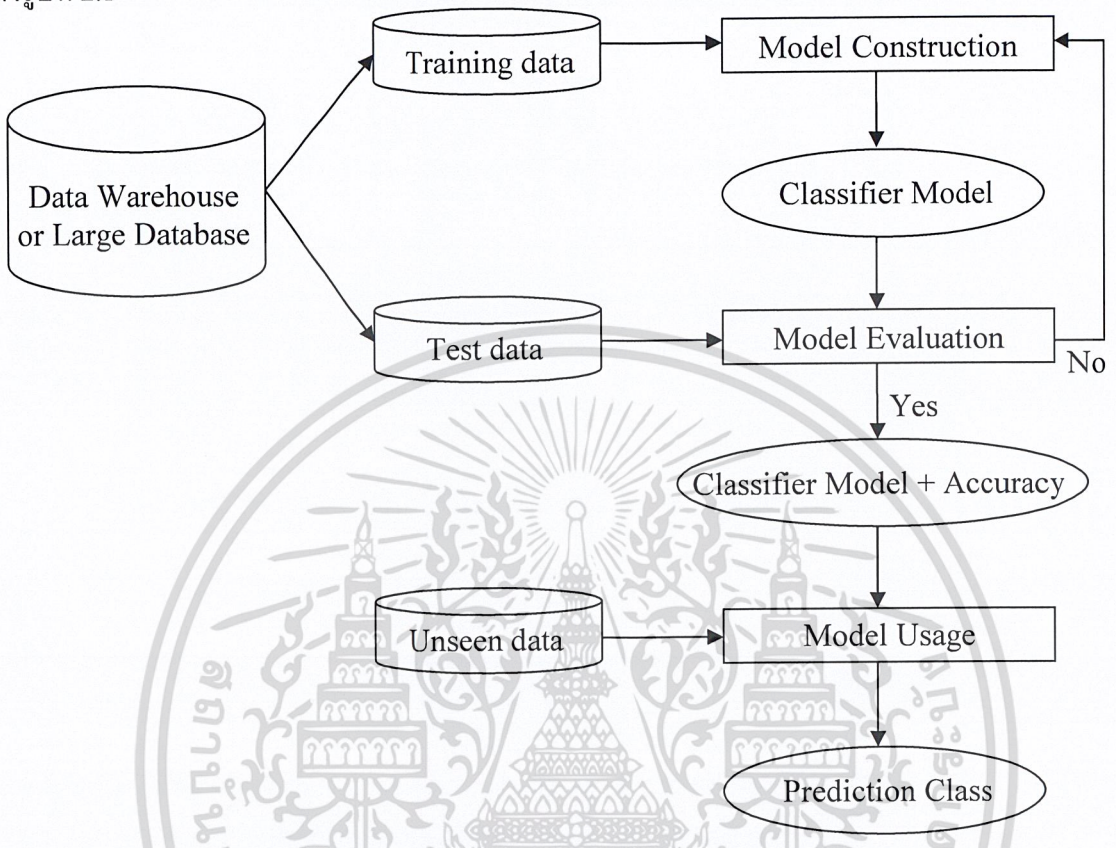
งานวิจัยเกี่ยวกับการศึกษาเปรียบเทียบการจัดกลุ่ม เป็นงานวิจัยที่สนใจว่า การทำงานของขั้นตอนวิธีการจัดกลุ่มที่มีการพัฒนาขึ้นแล้ว หากเทียบกับวิธีการแรกเริ่มจะสามารถให้ผลที่ดีขึ้นจริงหรือไม่ โดยใช้เกณฑ์ชุดข้อมูลเดียวกันในการทดสอบ ตัวอย่างของงานวิจัย ได้แก่

การวิเคราะห์เปรียบเทียบขั้นตอนวิธีบนพื้นฐาน k-means (Parresh Kumar and Siri Krishan Wasan, 2010) จากงานวิจัยจะทำการวิเคราะห์เปรียบเทียบขั้นตอนวิธีที่ได้รับการพัฒนาจาก k-means (MacQueen, 1967) ซึ่งได้แก่ x-means (Dan Relleg and Andre Moore, 2000) efficient k-means (Zhang et al.,2003), global k-means (Likas et al., 2003) และ k-means++ (David Arthur et Al., 2007) ชุดข้อมูล 2 ชุด ได้แก่ Colon และ Lenkemla จากการทดลองพบว่า ความแม่นยำในการจัดกลุ่มชุดข้อมูล Colon ทั้ง 5 วิธีทำได้ไม่ดีนัก โดยที่ Global k-means, x-means, k-means++ ให้ความแม่นยำสูงที่สุด ในการจัดกลุ่มชุดข้อมูล Lenkemia พบว่า ทั้ง 5 วิธีทำได้ดี โดยเมื่อใช้ 50-gene-lenkemia พบว่า k-means++ ให้ความแม่นยำสูงที่สุด และ k-means ให้ความแม่นยำรองลงมา เมื่อใช้ 3859-gene-lenkemia พบว่า global k-means และ k-means++ ให้ความแม่นยำสูงที่สุด และ k-means ให้ความแม่นยำต่ำที่สุด

เนื้อหาในส่วนต่อไปของงานวิจัยนี้ จะให้รายละเอียดของวิธีดำเนินการวิจัยโดยนำเอาความรู้และงานวิจัยต่างๆ ที่ได้กล่าวไปมาปรับปรุงและพัฒนาให้สามารถสร้างโมเดลจำแนกประเภทที่มีประสิทธิภาพสูง โดยการใช้การทำงานของต้นไม้ตัดสินใจ และการจัดกลุ่มเข้าด้วยกัน ทำให้มีความแม่นยำในการจำแนกกลุ่มของข้อมูลใหม่ได้อย่างถูกต้อง และสามารถทำงานบนฐานข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ

2.1.1 กระบวนการสร้างตัวโมเดลจำแนกประเภทข้อมูล

แบ่งออกเป็น 3 ขั้นตอน ซึ่งภาพรวมของกระบวนการสร้างโมเดลจำแนกประเภทข้อมูลแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 กระบวนการสร้างโมเดลจำแนกประเภทข้อมูล

กระบวนการของแต่ละขั้นตอนมีดังนี้

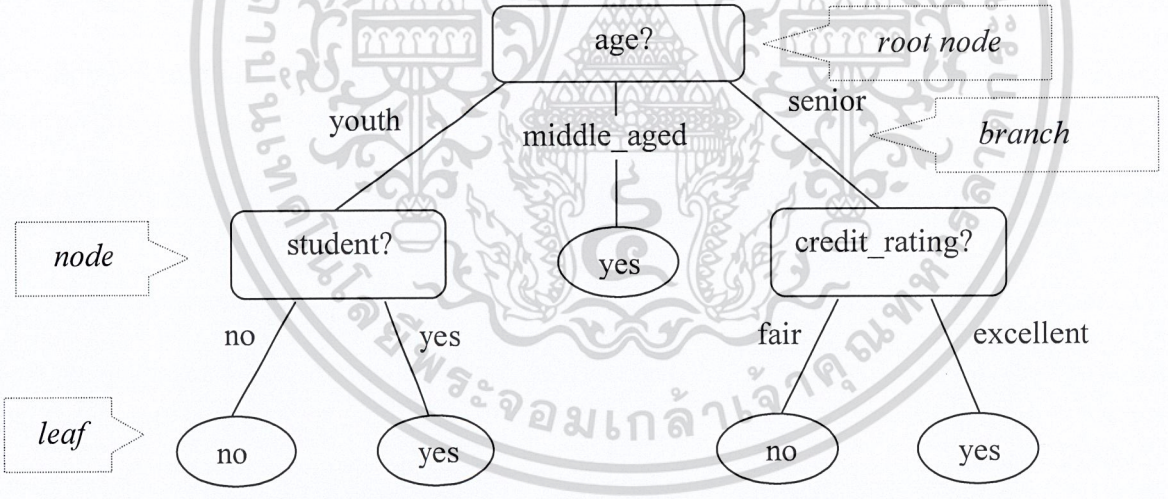
1) Model Construction (Learning) เป็นขั้นตอนการสร้างโมเดลจำแนกประเภท โดยอาศัยการเรียนรู้จากข้อมูลที่ได้กำหนด class ไว้เรียบร้อยแล้วหรือเรียกว่าข้อมูลเรียนรู้ (Training data) ซึ่งโมเดลจำแนกประเภทที่ได้จะแสดงด้วยวิธีการพื้นฐานทางเหมืองข้อมูล (Data mining) ยกตัวอย่างเช่น ต้นไม้ตัดสินใจ (Decision Tree) โมเดลจำแนกประเภทที่ได้จะมีลักษณะคล้ายต้นไม้จริงกลับหัวที่มีโหนดรากอยู่ด้านบนสุดและโหนดใบอยู่ล่างสุดของต้นไม้ แต่ละโหนดบนต้นไม้จะมีคุณลักษณะ (attribute) เป็นตัวเลือกทดสอบ ซึ่งจะมีกิ่งซึ่งเป็นค่าที่เป็นไปได้ของคุณลักษณะ (attribute value) ที่ถูกเลือกทดสอบไว้และมีโหนดใบแสดง class ที่กำหนดไว้

2) Model Evaluation (Accuracy) เป็นขั้นตอนตรวจสอบความถูกต้อง โดยอาศัยข้อมูลที่ใช้สำหรับทดสอบเรียกว่าข้อมูลทดสอบ (Testing data) ซึ่งกลุ่มที่แท้จริงของข้อมูลที่ใช้ทดสอบจะถูกนำมาเปรียบเทียบกับกลุ่มที่หามาได้จากโมเดลจำแนกประเภท เพื่อทดสอบว่าโมเดลจำแนกประเภทนี้สามารถจัดกลุ่มประเภทข้อมูลได้อย่างถูกต้องมากน้อยเพียงใด และมีการปรับปรุงโมเดลจำแนกประเภทจนกว่าจะได้ค่าความถูกต้องในระดับที่ยอมรับได้

3) Model Usage (Classification) เป็นขั้นตอนการนำโมเดลจำแนกประเภทที่สร้างขึ้นมาใช้กับข้อมูลที่ไม่เคยเห็นมาก่อน (unseen data) เพื่อทำนายและกำหนดกลุ่มให้กับข้อมูลนั้น

2.2 ต้นไม้ตัดสินใจ (Decision Tree)

ต้นไม้ตัดสินใจ (Decision Tree) เป็นโครงสร้างข้อมูลชนิดเป็นลำดับชั้น (hierarchy) ใช้สนับสนุนการตัดสินใจ โดยจะมีลักษณะคล้ายต้นไม้จริงกลับหัวที่มีโหนดรากอยู่ด้านบนสุดและโหนดใบอยู่ล่างสุดของต้นไม้ โดยที่ภายในต้นไม้จะประกอบไปด้วยโหนด (node) ซึ่งแต่ละโหนดจะมีคุณลักษณะ (attribute) เป็นตัวทดสอบ กิ่งของต้นไม้ (branch) แสดงถึงค่าที่เป็นไปได้ของคุณลักษณะที่ถูกเลือกทดสอบ และใบ (leaf) ซึ่งเป็นสิ่งที่อยู่ล่างสุดของต้นไม้ตัดสินใจแสดงถึงกลุ่มของข้อมูล (class) หรือนั่นก็คือผลลัพธ์ที่ได้จากการทำนาย โหนดที่อยู่บนสุดของต้นไม้เรียกว่าโหนดราก (root node) ดังแสดงโครงสร้างของต้นไม้ตัดสินใจตัดสินใจดังรูปที่ 2.2 ซึ่งเป็นต้นไม้ที่ใช้ในการตัดสินใจว่าจะเลือกซื้อคอมพิวเตอร์หรือไม่ (Quinlan, 1986) มีคุณลักษณะที่พิจารณาคืออายุ (age) นักศึกษา (student) และอัตราเครดิต (credit_rating) โดยที่โหนดสี่เหลี่ยมมุมโค้งจะเป็นการทดสอบคุณลักษณะของข้อมูล ท้ายสุดจะได้ผลลัพธ์ของการทำนายว่าจะซื้อคอมพิวเตอร์ (yes) หรือไม่ซื้อคอมพิวเตอร์ (no) จากการทดสอบตามเส้นทางของต้นไม้ตัดสินใจตั้งแต่โหนดรากไปจนถึงใบ



รูปที่ 2.2 ต้นไม้ตัดสินใจที่ใช้ในการตัดสินใจการเลือกซื้อคอมพิวเตอร์ (Han and Kamber, 2006, p.291)

2.2.1 การสร้างต้นไม้ตัดสินใจ

การสร้างต้นไม้ตัดสินใจจะสร้างในลักษณะจากบนลงล่าง (top-down) นั้นก็คือเริ่มจากการหาคุณลักษณะที่เหมาะสมที่สุดเพื่อนำมาเป็นรากของต้นไม้แล้วจึงแตกกิ่งไปจนถึงใบ โดยขั้นตอนการสร้างต้นไม้ตัดสินใจจะมีดังนี้ (Han and Kamber, 2006, p.293)

ขั้นตอนสุดท้าย ทำการสร้างการจัดกลุ่มแบบ k-means จากข้อมูลที่ได้ผ่านการคัดเลือกคุณลักษณะแล้ว โดยเริ่มต้นจากเลือกตัวอย่าง k ชุดเป็นจุดศูนย์กลางคลัสเตอร์เริ่มต้น แล้วทำการวนซ้ำกำหนดกลุ่มให้ข้อมูลเรียนรู้ด้วยการวัดความคล้ายกับจุดศูนย์กลางคลัสเตอร์ทั้งหมด แล้วเลือกคลัสเตอร์ที่มีความคล้ายกับข้อมูลนั้นมากที่สุด จากนั้นเปลี่ยนแปลงจุดศูนย์กลางคลัสเตอร์ด้วยการคำนวณค่าเฉลี่ยของข้อมูลทั้งหมดที่อยู่ในคลัสเตอร์ การวนซ้ำจะสิ้นสุดลงเมื่อจุดศูนย์กลางคลัสเตอร์ทั้งหมดไม่มีการเปลี่ยนแปลงค่าในการวนซ้ำแล้ว

Algorithm: Generate_decision_tree_C4.5. Generate a decision tree based on C4.5 algorithm from the training tuples of data partition *D*.

Input: - *D*, data partition;

- *A*, *attribute_list*;

Output: A decision tree based on C4.5, *M*.

Method:

(1) Create a node *N*.
(2) **if** *tuples* in *D* are all of the same class, *C* **then**
(3) return *N* as a leaf node labeled with the class *C*.
(4) **if** *A* is empty **then**
(5) return *N* as a leaf node labeled with the majority class in *D*. // majority voting
(6) assign *weight* to each data in *D*.
(7) a compute *node N's* weight information entropy (*WIE*).
(8) **for each** attribute *k* of *attribute_list A*
(9) **if** A_k is discrete-valued **then**
(10) count number of outcome A_k with outcome of class in *D*.
(11) compute weight information gain (*WIG*) of A_k.
(12) compute weight splitting information (*WSI*) and weight gain radio (*WGR*) of A_k.
(13) **else** sort data in *D* by A_k and find best split point for A_k.
(14) compute weight information gain (*WIG*) of A_k.
(15) compute weight splitting information (*WSI*) and weight gain radio (*WGR*) of A_k.
(16) **end if**
(17) **end for**
(18) choose *splitting_criterion* of attribute this is the most weight gain radio (*WGR*).

รูปที่ 3.3 (ก) ขั้นตอนวิธีการสร้างต้นไม้ตัดสินใจ C4.5

(19) *label* node *N* with *splitting_criterion*.

(20) *attribute_list* ← *attribute_list – splitting_attribute* // remove splitting_ attribute

(21) **for each** outcome *j* of *splitting_criterion*

// partition the tuples and grow subtrees for each partition.

(22) let D_j be the set of data tuples in *D* satisfying outcome *j*. // a partition

(23) **if** D_j is empty **then**

(24) attach a leaf labled with the majority class in *D* to node *N*.

(25) **else** attach the node return by **Generate_decision_tree_C4.5**(D_j, *A*) to node N.

(26) **end for**

(27) return *N*;

รูปที่ 3.3 (ข) ขั้นตอนวิธีการสร้างต้นไม้ตัดสินใจ C4.5

จากรูปที่ 3.3 ขั้นตอนวิธีการสร้างต้นไม้ตัดสินใจ C4.5 ขั้นตอนแรก จะสร้างโหนด *N* จากนั้นพิจารณาว่าถ้าข้อมูลทั้งหมดอยู่ในกลุ่มเดียวกันแล้ว ให้โหนด *N* เป็นโหนดใบและกำหนดค่าด้วยกลุ่มของข้อมูลนั้น และพิจารณาว่าถ้าข้อมูลไม่มีคุณลักษณะใดที่เหมาะสมในการแบ่งกลุ่ม ให้โหนด N เป็นโหนดใบและกำหนดค่าด้วยกลุ่มที่มีข้อมูลสนับสนุนมากที่สุด

ขั้นตอนต่อมา กำหนดน้ำหนักในแต่ละข้อมูลใน *D* โดยใช้สมการที่ 2.8 จากนั้นคำนวณค่าความรู้จากทฤษฎีสารสนเทศถ่วงน้ำหนัก (*WIE*) โดยใช้สมการที่ 2.10

ขั้นตอนถัดไป พิจารณาแต่ละคุณลักษณะว่ามีค่าไม่ต่อเนื่อง (discrete valued) หรือค่าต่อเนื่อง (continues valued) ถ้าเป็นค่าไม่ต่อเนื่องให้นับจำนวนโดยพิจารณาตามค่าที่เป็นไปได้ของคุณลักษณะและกลุ่มของข้อมูล แต่ถ้าเป็นค่าต่อเนื่องให้เรียงข้อมูลแล้วคำนวณค่าจุดแบ่งที่ดีที่สุดออกมา ต่อมาคำนวณค่ามาตรฐานเกนถ่วงน้ำหนัก (*WIG*) ค่าสารสนเทศของการแบ่งแยกถ่วงน้ำหนัก (*WSI*) และค่ามาตรฐานอัตราส่วนเกนถ่วงน้ำหนัก (*WGR*) โดยใช้สมการที่ 2.9, 2.12 และ 2.11 ตามลำดับ

จากนั้นเลือกคุณลักษณะที่มีค่ามาตรฐานอัตราส่วนเกนถ่วงน้ำหนักสูงที่สุดและกำหนดค่าให้โหนด *N* เป็นตัวทดสอบการตัดสินใจด้วยคุณลักษณะที่ได้มา

ขั้นตอนสุดท้าย ทำการวนซ้ำเพื่อแบ่งข้อมูลและแตกกิ่งของต้นไม้ แล้วพิจารณาข้อมูลแต่ละกิ่ง หากพบว่าข้อมูลทั้งหมดอยู่ในกลุ่มเดียวกัน ให้ต่อกิ่งด้วยโหนดใบ และกำหนดค่าด้วยกลุ่มของข้อมูลนั้น แต่ถ้าพบว่าข้อมูลมีหลากหลายกลุ่มปะปนกัน ให้ทำการวนซ้ำการหาตัวทดสอบการตัดสินใจที่เหมาะสมต่อไปจนกว่าเงื่อนไขข้อใดข้อหนึ่งต่อไปนี้เป็นจริง

1) ข้อมูลทั้งหมดในโหนดอยู่ในกลุ่มเดียวกัน

2) ไม่มีคุณลักษณะใดที่เหมาะสมในการแบ่งกลุ่ม

Algorithm: Attribute_selection_by_treebagging. Attribute selection in D with a bagging algorithm with decision tree based on C4.5

Input: - D, data partition;

- A, *attribute_list*;

- M^*, decision tree base on C4.5 model set;

- n, number of model;

Output: - Data partition with selection, D^*.

- *weight_attribute_list*, WA.

Method:

(1) $D^* = D$

(2) let WA be the set of weight belong to an attribute and initialized to zero.

(3) **for** $i = 1$ to n **do**

(4) let *temp_weight* be the set of temporary weight belong to an attribute and initialized to zero.

(5) **for each** level j of M_i // start from level 0 to level $height\_M_i$-1

(6) let k be attribute position in A.

(7) **if** $temp\_weight_k == 0$ **then**

(8) $temp\_weight_k = \frac{height\_M_i - j+1}{height\_M_i+1}$;

(9) **end if**

(10) **end for**

(11) $WA = WA + temp\_weight$.

(12) **end for**

(13) **for each** attribute p of attribute_list A

(14) $WA_p = \frac{WA_p}{n}$;

(15) **if** $WA_p = 0$ then

(16) delete A_p and column p for each data in D^*.

(17) **end if**

(18) **end for**

(19) $WA = \frac{WA - \mathbf{MIN}(WA)}{\mathbf{MAX}(WA) - \mathbf{MIN}(WA)}$ // min-max normalize WA to [0, 1].

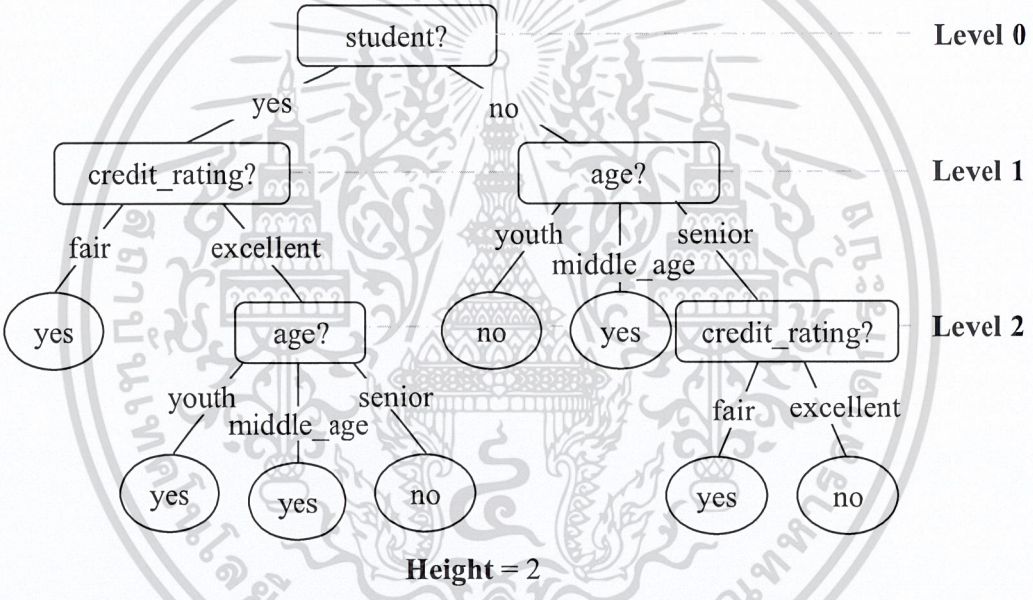
(20) return D^* and WA;

รูปที่ 3.4 ขั้นตอนวิธีการคัดเลือกคุณลักษณะจากโมเดลทั้งหมดที่ได้

จากรูปที่ 3.4 ขั้นตอนวิธีการคัดเลือกคุณลักษณะ ขั้นตอนแรกจะสร้างเซตของน้ำหนักแต่ละคุณลักษณะขึ้นมา (*WA*) กำหนดค่าเริ่มต้นให้เป็น 0 ทั้งหมด

ขั้นตอนต่อไป พิจารณาโมเดลทั้งหมด สร้างเซตของน้ำหนักแต่ละคุณลักษณะชั่วคราวขึ้นมา (*temp_weight*) ซึ่งจะเก็บค่าน้ำหนักของคุณลักษณะบนโมเดลที่กำลังสนใจเท่านั้น และเนื่องจากเราจะให้น้ำหนักเป็นค่าที่เกิดขึ้นมากที่สุดบนต้นไม้ตัดสินใจ เช่นกรณีที่พบว่าคุณลักษณะอยู่ในระดับที่ 1 และระดับที่ 2 ของต้นไม้ตัดสินใจก็จะให้น้ำหนักเป็นระดับที่ 1 จากนั้นเพิ่มค่าน้ำหนักของ *temp_weight* ให้กับ *WA* จนครบทุกโมเดล

ตัวอย่างการคำนวณค่าน้ำหนักของแต่ละคุณลักษณะ สมมุติให้การทำงานมีจำนวนโมเดล $n = 2$ และมีคุณลักษณะทั้งหมด $A = \{age, income, student, credit\_rating\}$ เริ่มต้นพิจารณาโมเดลต้นไม้ตัดสินใจที่ 1 (M_1) แสดงได้ดังรูปที่ 3.5



รูปที่ 3.5 โมเดลต้นไม้ตัดสินใจที่ 1 (M_1)

จากรูปที่ 3.5 จะเห็นได้ว่าโมเดลมีความสูงเท่ากับ 2 และมีระดับทั้งหมด 3 ระดับด้วยกันคือ ระดับที่ 0, 1 และ 2 การคำนวณค่าน้ำหนักของแต่ละคุณลักษณะจะพิจารณาเริ่มต้นที่ระดับที่ 0 พบว่ามีคุณลักษณะ *student* เป็นตัวทดสอบการตัดสินใจ ดังนั้นจะให้น้ำหนักของคุณลักษณะ *student* โดยใช้สูตรที่ 3.1

$$w_{k,i} = \frac{height\_M_i - j + 1}{height\_M_i + 1} \tag{3.1}$$

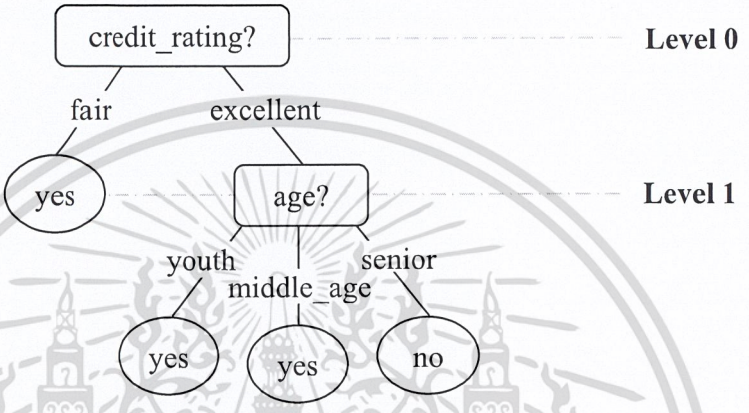
โดยที่ $w_{k,i}$ เป็นน้ำหนักของคุณลักษณะ k บนโมเดล i

$height\_M_i$ เป็นความสูงของโมเดล i

j เป็นระดับของคุณลักษณะ k

ค่าน้ำหนักของ *student* ที่ได้คือ $\frac{2-0+1}{2+1} = 1$ จากนั้นพิจารณาระดับที่ 1 พบว่ามีคุณลักษณะ *credit_rating* และ *age* เป็นตัวทดสอบการตัดสินใจ ดังนั้นจะให้น้ำหนักของคุณลักษณะทั้งสอง โดยใช้สูตรที่ 3.1 พบว่าค่าน้ำหนักที่ได้คือ $\frac{2-1+1}{2+1} = 0.67$ สุดท้ายพิจารณาระดับที่ 2 พบว่ามีคุณลักษณะ *age* และ *credit_rating* เป็นตัวทดสอบการตัดสินใจ แต่พบว่าคุณลักษณะทั้งสองถูกพบในระดับที่ 1 แล้ว จึงไม่ต้องคำนวณค่าน้ำหนักเปลี่ยนแปลงจากเดิม

พิจารณาโมเดลต้นไม้ตัดสินใจที่ 2 (M_2) แสดงได้ดังรูปที่ 3.6



รูปที่ 3.6 โมเดลต้นไม้ตัดสินใจที่ 2 (M_2)

จากรูปที่ 3.6 จะเห็นได้ว่าโมเดลมีความสูงเท่ากับ 1 และมีระดับทั้งหมด 2 ระดับด้วยกันคือระดับที่ 0 และ 1 การคำนวณค่าน้ำหนักของแต่ละคุณลักษณะจะพิจารณาเริ่มต้นที่ระดับที่ 0 พบว่ามีคุณลักษณะ *credit_rating* เป็นตัวทดสอบการตัดสินใจ ดังนั้นจะให้น้ำหนักของคุณลักษณะ *credit_rating* โดยใช้สูตรที่ 3.1 พบว่าค่าน้ำหนักที่ได้คือ $\frac{1-0+1}{1+1} = 1$ สุดท้ายพิจารณาระดับที่ 1 พบว่ามีคุณลักษณะ *age* เป็นตัวทดสอบการตัดสินใจ ดังนั้นจะให้น้ำหนักของคุณลักษณะ *age* โดยใช้สูตรที่ 3.1 พบว่าค่าน้ำหนักที่ได้คือ $\frac{1-1+1}{1+1} = 0.5$

เมื่อได้พิจารณาโมเดลต้นไม้ตัดสินใจทุกโมเดลแล้ว ให้คำนวณน้ำหนักรวมของแต่ละคุณลักษณะ โดยใช้สูตรที่ 3.2

$$w_k = \frac{w_{k,1} + w_{k,2} + \cdots + w_{k,n}}{n} \tag{3.2}$$

โดยที่ w_k เป็นน้ำหนักรวมของคุณลักษณะ k

n เป็นจำนวนโมเดล

ค่าน้ำหนักรวมของแต่ละคุณลักษณะที่มีคือ $w_{age} = \frac{0.67+0.5}{2} = 0.59, w_{income} = \frac{0+0}{2} = 0,$ $w_{student} = \frac{1+0}{2} = 0.5$ และ $w_{credit\_rating} = \frac{0.67+1}{2} = 0.84$

ขั้นตอนสุดท้าย พิจารณาแต่ละคุณลักษณะ ให้คำนวณปรับค่าน้ำหนักให้อยู่ในช่วง 0 ถึง 1 และตรวจสอบว่าค่าน้ำหนักที่ได้นั้น มีค่าเท่ากับ 0 หรือไม่ หากพบว่าเท่ากับ 0 แสดงว่าคุณลักษณะนั้นไม่มีความสำคัญในการจำแนกประเภท ดังนั้นจึงทำการลบคุณลักษณะนั้นออก และลบข้อมูลคอลัมน์ที่มีค่าคุณลักษณะนั้นออกจากข้อมูลเรียนรู้ที่มี จากนั้นปรับค่าน้ำหนักอีกครั้ง โดยใช้การปรับค่าแบบ Min-Max Normalization กำหนดค่าต่ำสุดเป็น 0 และค่าสูงสุดเป็น 1 ทำให้ค่าน้ำหนักของคุณลักษณะมีการกระจายตัวมากกว่าเดิม เมื่อดำเนินการเรียบร้อยแล้วให้คืนค่าชุดข้อมูลเรียนรู้และเซตของน้ำหนักแต่ละคุณลักษณะที่ได้กลับไปให้การทำงานหลัก

การทำนายด้วยขั้นตอนวิธี Tree Bagging and Weighted Clustering แสดงเป็นขั้นตอนดังนี้

Algorithm: Predict_TBWC. Use a set of *k* centroids and a set of class labels to predict a new example *X*.

Input: - *X*, a new example;
- *C*, a set of k centroids;
- *L*, a set of class labels;

Output: a class label of a new example *X*.

Method:

(1) compare distance between *X* and all members of a set *C*.

(2) select the nearest cluster.

(3) assign class label to *X* by using class label of a selected cluster

(4) return a class label of the new example *X*.

รูปที่ 3.7 ขั้นตอนวิธีการทำนายด้วยขั้นตอนวิธี Tree Bagging and Weighted Clustering

ขั้นตอนแรก เปรียบเทียบระหว่างตัวอย่างใหม่กับจุดศูนย์กลางคลัสเตอร์ทั้งหมดแล้วเลือกคลัสเตอร์ที่มีความคล้ายกับตัวอย่างนั้นมากที่สุด จากนั้นกำหนดชื่อกลุ่มของตัวอย่างด้วยชื่อกลุ่มของคลัสเตอร์ที่ถูกเลือก ขั้นตอนสุดท้ายคืนค่าชื่อกลุ่มของตัวอย่างใหม่ที่ได้กลับไปให้การทำงานหลัก

บทที่ 4

ผลการทดลอง

4.1 แหล่งที่มาและรายละเอียดชุดข้อมูลเรียนรู้

ข้อมูลที่ใช้ในการวิจัยเพื่อใช้ทดสอบประสิทธิภาพของขั้นตอนวิธีที่ได้พัฒนาขึ้น จะคัดเลือกชุดข้อมูลจากแหล่งข้อมูลของมหาวิทยาลัยแห่งรัฐแคลิฟอร์เนีย เมืองเออร์ไวย์ ประเทศสหรัฐอเมริกา (http://archive.ics.uci.edu/ml/) (Blake and Merz, 1998) ซึ่งได้รวบรวมชุดข้อมูลสำหรับใช้เป็นเกณฑ์สำหรับทดสอบประสิทธิภาพการทำงานของอัลกอริทึมต่างๆ ของการทำเหมืองข้อมูล โดยได้ทำการคัดเลือกข้อมูลจำนวนทั้งหมด 4 ชุดข้อมูล และในการคัดเลือกชุดข้อมูล จะพิจารณาลักษณะของข้อมูลดังต่อไปนี้

1) เป็นชุดข้อมูลที่มีจำนวนมากกว่า 1,000 ตัวอย่าง ยกเว้นชุดข้อมูล Libras Movement เพื่อเปรียบเทียบผลการทดลองว่า หากจำนวนตัวอย่างมีจำนวนน้อยจะให้ผลลัพธ์ที่มีประสิทธิภาพหรือไม่

2) เป็นชุดข้อมูลที่มีจำนวนคุณลักษณะมากกว่า 90 รายการ ยกเว้นชุดข้อมูล Cardiocography 1 และ Cardiocography 2 เพื่อเปรียบเทียบผลการทดลองว่า หากจำนวนคุณลักษณะมีจำนวนน้อยจะให้ผลลัพธ์ที่มีประสิทธิภาพหรือไม่

รายชื่อชุดข้อมูล จำนวนตัวอย่าง จำนวนคุณลักษณะจากการจำแนกคุณลักษณะต่างๆ และจำนวนกลุ่มในชุดข้อมูลมีรายละเอียดดังตารางที่ 4.1

ตารางที่ 4.1 รายละเอียดของชุดข้อมูลโดยสรุปที่ใช้ในงานวิจัย

ชื่อชุดข้อมูล	จำนวนตัวอย่าง	คุณลักษณะ		จำนวนกลุ่ม
		ทั้งหมด	รูปแบบ	
1. Cardiocography 1	2,126	23	real	3
2. Cardiocography 2	2,126	23	real	10
3. Internet Advertisement	3,279	1,558	categorical, integer, real	2
4. Libras Movement	360	91	real	15
5. Multiple Features	2,000	649	integer, real	10

ข้อมูลแต่ละชุดจะมีรายละเอียดและการจำแนกกลุ่มของข้อมูลดังต่อไปนี้

1) Cardiocography 1 เป็นข้อมูลที่ประกอบด้วยการตรวจวัดอัตราการเต้นหัวใจของทารกในครรภ์ (FHR) และอัตราการหดตัวของมดลูก (UC)

มีจำนวนข้อมูล 2,126 ตัวอย่าง จำแนกออกเป็น 3 กลุ่ม คือ

- normal (จำนวน 1,655 ตัวอย่าง)
- suspect (จำนวน 295 ตัวอย่าง)
- pathologic (จำนวน 176 ตัวอย่าง)

2) Cardiocography 2 เป็นข้อมูลที่ประกอบด้วยการตรวจวัดอัตราการเต้นหัวใจของทารกในครรภ์ (FHR) และอัตราการหดตัวของมดลูก (UC)

มีจำนวนข้อมูล 2,126 ตัวอย่าง จำแนกออกเป็น 10 กลุ่ม คือ

- FHR pattern class code is 1 (จำนวน 384 ตัวอย่าง)
- FHR pattern class code is 2 (จำนวน 379 ตัวอย่าง)
- FHR pattern class code is 3 (จำนวน 53 ตัวอย่าง)
- FHR pattern class code is 4 (จำนวน 81 ตัวอย่าง)
- FHR pattern class code is 5 (จำนวน 72 ตัวอย่าง)
- FHR pattern class code is 6 (จำนวน 332 ตัวอย่าง)
- FHR pattern class code is 7 (จำนวน 252 ตัวอย่าง)
- FHR pattern class code is 8 (จำนวน 107 ตัวอย่าง)
- FHR pattern class code is 9 (จำนวน 69 ตัวอย่าง)
- FHR pattern class code is 10 (จำนวน 197 ตัวอย่าง)

3) Internet Advertisements เป็นข้อมูลของปัญหาการจำแนกรูปภาพบนหน้าเว็บเพจ

มีจำนวนข้อมูล 3,279 ตัวอย่าง จำแนกเป็น 2 กลุ่ม คือ

- advertisement (จำนวน 2,820 ตัวอย่าง)
- not advertisement (จำนวน 459 ตัวอย่าง)

4) Libras Movement เป็นข้อมูลของปัญหาการจำแนกประเภทการเคลื่อนไหวของมือ มีจำนวนข้อมูล 360 ตัวอย่าง จำแนกออกเป็น 15 กลุ่ม คือ

- curved swing (จำนวน 24 ตัวอย่าง)
- horizontal swing (จำนวน 24 ตัวอย่าง)
- vertical swing (จำนวน 24 ตัวอย่าง)
- anti-clockwise arc (จำนวน 24 ตัวอย่าง)
- clockwise arc (จำนวน 24 ตัวอย่าง)
- circle (จำนวน 24 ตัวอย่าง)
- horizontal straight-line (จำนวน 24 ตัวอย่าง)
- vertical straight-line (จำนวน 24 ตัวอย่าง)
- horizontal zigzag (จำนวน 24 ตัวอย่าง)
- vertical zigzag (จำนวน 24 ตัวอย่าง)
- horizontal wavy (จำนวน 24 ตัวอย่าง)
- vertical wavy (จำนวน 24 ตัวอย่าง)
- face-up curve (จำนวน 24 ตัวอย่าง)
- face-down curve (จำนวน 24 ตัวอย่าง)
- tremble (จำนวน 24 ตัวอย่าง)

5) Multiple Features เป็นข้อมูลของปัญหาการจำแนกตัวเลขจากลายนิ้วมืออิเล็กทรอนิกส์ มีจำนวนข้อมูล 2,000 ตัวอย่าง จำแนกออกเป็น 10 กลุ่ม คือ

- '0' (จำนวน 200 ตัวอย่าง)
- '1' (จำนวน 200 ตัวอย่าง)
- '2' (จำนวน 200 ตัวอย่าง)
- '3' (จำนวน 200 ตัวอย่าง)
- '4' (จำนวน 200 ตัวอย่าง)
- '5' (จำนวน 200 ตัวอย่าง)
- '6' (จำนวน 200 ตัวอย่าง)
- '7' (จำนวน 200 ตัวอย่าง)
- '8' (จำนวน 200 ตัวอย่าง)
- '9' (จำนวน 200 ตัวอย่าง)

4.2 ผลการทดลอง

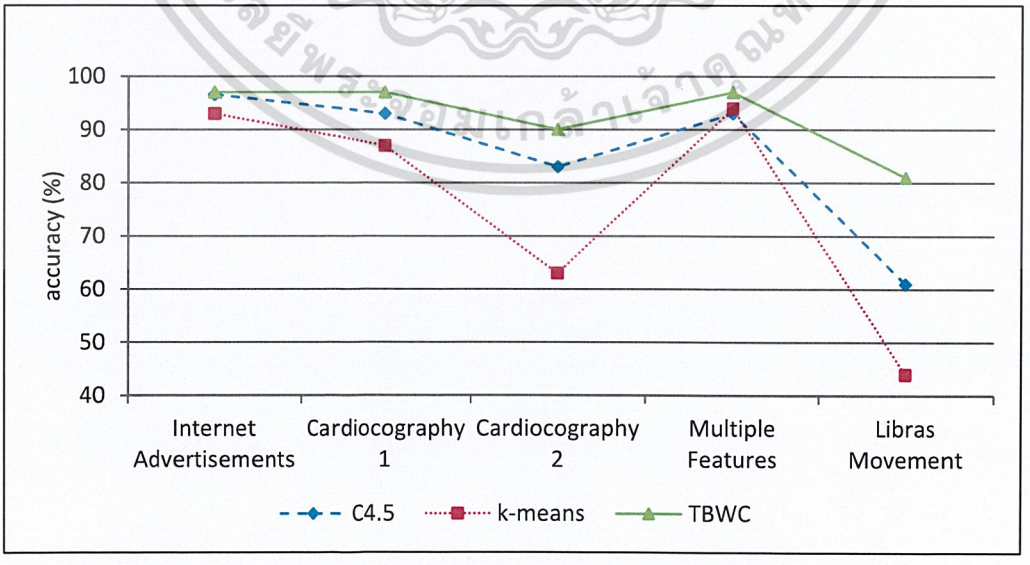
การทดลองวัดประสิทธิภาพของการทำนายด้วยขั้นตอนวิธี TBWC ได้มีการพัฒนาขั้นตอนวิธีด้วยโปรแกรม MATLAB Version 7.10.0.499 (R2010a) 32-bit โดยใช้ Toolbox ที่สำคัญ 2 ชนิดคือ Statistics Toolbox และ MATLABArsenal Toolbox

การวิเคราะห์เปรียบเทียบประสิทธิภาพของการจำแนกประเภทนั้น จะใช้ขั้นตอนวิธีการจำแนกประเภทที่สำคัญ 2 วิธี ได้แก่ ขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means ซึ่งนำไปเปรียบเทียบกับขั้นตอนวิธีการจำแนกประเภทที่พัฒนาขึ้นมาโดยใช้ชื่อว่าขั้นตอนวิธี TBWC ทดสอบกับชุดข้อมูลเรียนรู้ที่เลือกมาจำนวน 5 ชุด ผลการทดลองสรุปได้ดังตารางที่ 4.2 ซึ่งมีการใช้พารามิเตอร์ต่างๆ ดังตารางที่ 4.4 สามารถลดทอนคุณลักษณะดังตารางที่ 4.5 และเวลาที่ใช้ในการเรียนรู้และทำนายชุดข้อมูลดังตารางที่ 4.6

ตารางที่ 4.2 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลเรียนรู้

ชื่อชุดข้อมูล	C4.5	k-means	TBWC
Cardiocography 1	92.56%	86.97%	96.47%
Cardiocography 2	82.83%	62.65%	90.35%
Internet Advertisements	96.19%	93.11%	96.65%
Libras Movement	60.56%	43.89%	80.56%
Multiple Features	92.60%	93.45%	96.55%

จากข้อมูลในตารางที่ 4.2 เมื่อนำข้อมูลที่ได้มาสร้างเป็นกราฟแสดงประสิทธิภาพของขั้นตอนวิธีจำแนกประเภท สามารถแสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 กราฟแสดงประสิทธิภาพในการจำแนกประเภทของขั้นตอนวิธีต่างๆ

จากกราฟในรูปที่ 4.1 แสดงประสิทธิภาพของขั้นตอนวิธีที่พัฒนาขึ้น (TBWC) เมื่อเปรียบเทียบกับขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means โดยใช้ชุดข้อมูลเรียนรู้ Cardiocography 1, Cardiocography 2, Internet Advertisement, Libras Movement และ Multiple Features ซึ่งจากกราฟทำให้สรุปได้ว่า ขั้นตอนวิธี TBWC ให้ค่าความแม่นยำมากกว่าขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means ในทุกชุดข้อมูลเรียนรู้ โดยผลการเปรียบเทียบค่าความแม่นยำระหว่างขั้นตอนวิธีที่พัฒนาขึ้น (TBWC) เทียบกับขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means สรุปได้ดังตารางที่ 4.3

ตารางที่ 4.3 ผลการเปรียบเทียบค่าความแม่นยำระหว่างขั้นตอนวิธีต่างๆ

ชื่อชุดข้อมูล	ผลการเปรียบเทียบ TBWC เทียบกับ C4.5	ผลการเปรียบเทียบ TBWC เทียบกับ k-means	จำนวนกลุ่ม
Cardiocography 1	+ 3.91%	+ 9.50%	3
Cardiocography 2	+ 7.52%	+ 27.70%	10
Internet Advertisements	+ 0.46%	+ 3.54%	2
Libras Movement	+ 20.00%	+ 36.67%	15
Multiple Features	+ 3.95%	+ 3.10%	10

จากตารางที่ 4.3 แสดงให้เห็นผลการเปรียบเทียบค่าความแม่นยำระหว่างขั้นตอนวิธีต่างๆ โดยที่ "+" คือสัญลักษณ์แสดงว่าขั้นตอนวิธี TBWC ให้ค่าความแม่นยำมากกว่าในการทำนายเมื่อเทียบกับขั้นตอนวิธี C4.5 หรือการจัดกลุ่ม k-means ตัวอย่างเช่นในชุดข้อมูลเรียนรู้ Cardiocography 1 ผลการเปรียบเทียบขั้นตอนวิธี TBWC เทียบกับขั้นตอนวิธี C4.5 คือ "+ 3.91%" ซึ่งหมายถึง ขั้นตอนวิธี TBWC ให้ค่าความแม่นยำในการทำนายมากกว่าขั้นตอนวิธี C4.5 3.91 เปอร์เซ็นต์

จากผลการเปรียบเทียบในตารางที่ 4.3 แสดงให้เห็นว่าขั้นตอนวิธีที่พัฒนาขึ้น (TBWC) สามารถปรับปรุงประสิทธิภาพการทำนายได้มากที่สุดเมื่อเทียบกับขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means ในชุดข้อมูล Libras Movement และการปรับปรุงประสิทธิภาพการทำนายมีการแปรผันตรงตามจำนวนกลุ่ม กล่าวคือเมื่อมีจำนวนกลุ่มมาก การปรับปรุงประสิทธิภาพการทำนายจะทำได้อย่างดี

ตารางที่ 4.4 พารามิเตอร์ที่ใช้ในขั้นตอนวิธีการจำแนกประเภทข้อมูล

ชื่อชุดข้อมูล	จำนวนโมเดล (n)	จำนวนคลัสเตอร์ (k)
Cardiocography 1	40	100
Cardiocography 2	20	100
Internet Advertisements	10	100
Libras Movement	5	20
Multiple Features	10	80

จากตารางที่ 4.4 เป็นพารามิเตอร์ที่ใช้ในขั้นตอนวิธีการจำแนกประเภทข้อมูลแต่ละชุดข้อมูลเรียนรู้ โดยที่ค่าจำนวนโมเดล (n) คือ จำนวนของโมเดลในการสร้างโมเดลจำแนกประเภทด้วยขั้นตอนวิธี Bagging ด้วยต้นไม้ตัดสินใจ C4.5 และค่าจำนวนคลัสเตอร์ (k) คือ จำนวนของคลัสเตอร์ในการจัดกลุ่มข้อมูลด้วยขั้นตอนวิธีการจัดกลุ่ม k-means ซึ่งการเลือกค่าพารามิเตอร์ต่างๆ นั้นแสดงไว้ในภาคผนวก ก โดยการเลือกจะใช้ค่าพารามิเตอร์ที่นำมาสร้างโมเดลแล้วให้ค่าความแม่นยำมากที่สุด และมีค่าจำนวนของโมเดลต่ำที่สุด

ตารางที่ 4.5 เปอร์เซ็นต์การลดทอนคุณลักษณะในขั้นตอนวิธีที่พัฒนาขึ้น (TBWC)

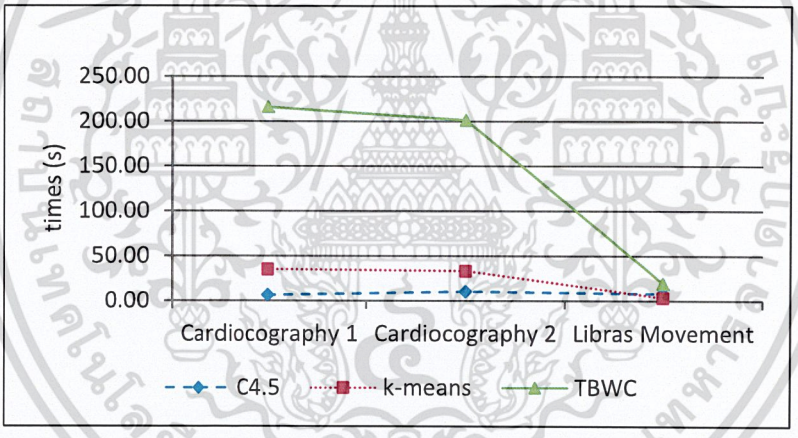
ชื่อชุดข้อมูล	จำนวนคุณลักษณะทั้งหมด	จำนวนคุณลักษณะที่คัดเลือกมาใช้งาน	เปอร์เซ็นต์การลดทอนคุณลักษณะ
Cardiocography 1	23	22	4.35%
Cardiocography 2	23	22	4.35%
Internet Advertisements	1,558	626	59.82%
Libras Movement	91	86	5.49%
Multiple Features	649	454	30.05%

จากตารางที่ 4.5 แสดงเปอร์เซ็นต์การลดทอนคุณลักษณะในการจำแนกประเภทด้วยขั้นตอนวิธีที่พัฒนาขึ้น (TBWC) ทำให้สรุปได้ว่า ขั้นตอนวิธีที่พัฒนาขึ้นนั้นสามารถลดทอนคุณลักษณะได้มากที่สุดถึง 59.82% ในชุดข้อมูล Internet Advertisements และให้ผลลัพธ์มากกว่าขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means แต่สำหรับชุดข้อมูล Cardiocography 1 และ Cardiocography 2 นั้นสามารถลดทอนคุณลักษณะได้เพียงเล็กน้อย เนื่องจาก ชุดข้อมูลเรียนรู้มีจำนวนคุณลักษณะทั้งหมดที่ไม่มาก เมื่อเทียบกับจำนวนตัวอย่าง และส่วนชุดข้อมูล Libras Movement ที่สามารถลดทอนได้เพียงเล็กน้อยเช่นกัน เนื่องจากชุดข้อมูลมีจำนวนกลุ่มจำนวนมาก เมื่อเทียบกับจำนวนตัวอย่าง ทำให้คุณลักษณะเกือบทั้งหมดเป็นค่าที่สำคัญในการจำแนกประเภท ไม่สามารถที่จะลดทอนออกไปได้

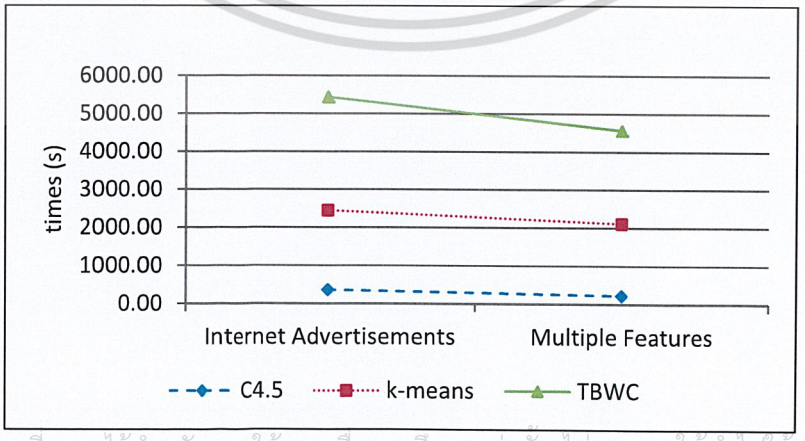
ตารางที่ 4.6 เวลาที่ใช้ในการเรียนรู้และทำนายชุดข้อมูลของชุดข้อมูลเรียนรู้

ชื่อชุดข้อมูล	C4.5			k-means			TBWC		
	เวลาที่ใช้เรียนรู้ (วินาที)	เวลาที่ใช้ทำนาย (วินาที)	เวลารวม (วินาที)	เวลาที่ใช้เรียนรู้ (วินาที)	เวลาที่ใช้ทำนาย (วินาที)	เวลารวม (วินาที)	เวลาที่ใช้เรียนรู้ (วินาที)	เวลาที่ใช้ทำนาย (วินาที)	เวลารวม (วินาที)
Cardiocography 1	6.33	0.28	6.61	35.23	0.04	35.27	215.93	0.03	215.96
Cardiocography 2	10.21	0.17	10.38	33.31	0.04	33.35	201.28	0.03	201.31
Internet Advertisements	355.00	2.33	357.33	2446.63	3.11	2449.74	5435.26	2.46	5437.72
Libras Movement	7.86	0.08	7.94	3.37	0.01	3.38	19.06	0.01	19.07
Multiple Features	219.13	2.59	221.72	2110.04	0.65	2110.69	4560.52	0.57	4561.09

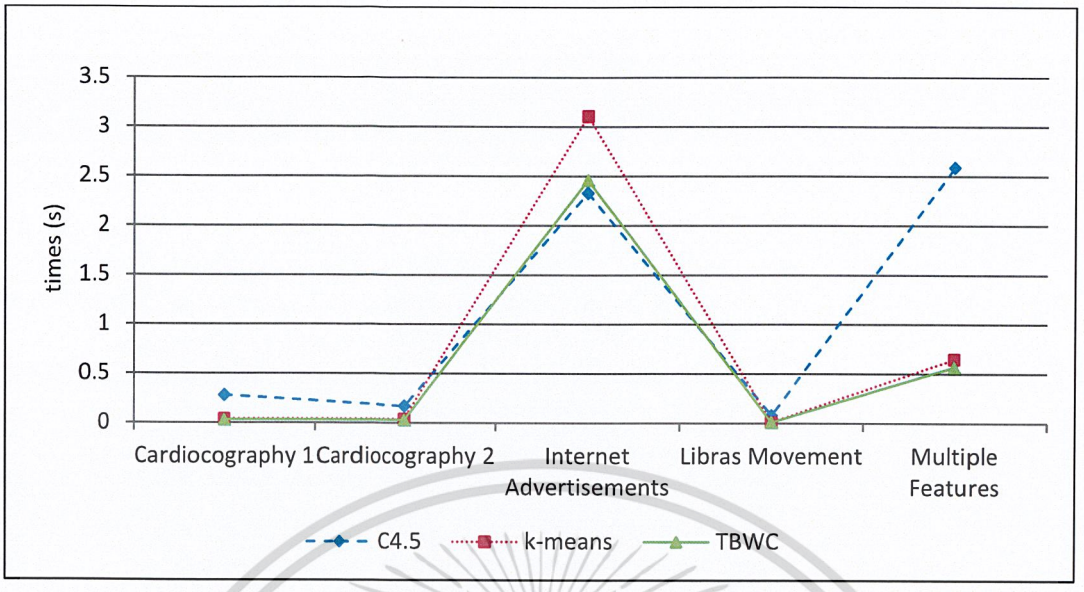
จากข้อมูลในตารางที่ 4.6 เมื่อนำข้อมูลที่ได้มาสร้างเป็นกราฟแสดงเวลาที่ใช้ในการเรียนรู้ สามารถแสดงได้ดังรูปที่ 4.2 กับ 4.3 และสร้างเป็นกราฟแสดงเวลาที่ใช้ในการทำนาย สามารถแสดงได้ดังรูปที่ 4.4



รูปที่ 4.2 กราฟแสดงเวลาที่ใช้ในการเรียนรู้ชุดข้อมูล Cardiocography 1, Cardiocography 2 และ Libras Movement



รูปที่ 4.3 กราฟแสดงเวลาที่ใช้ในการเรียนรู้ชุดข้อมูล Internet Advertisement และ Multiple Features



รูปที่ 4.4 กราฟแสดงเวลาที่ใช้ในการทำนายชุดข้อมูลเรียนรู้

จากกราฟในรูปที่ 4.2 และ 4.3 แสดงเวลาที่ใช้ในการเรียนรู้ชุดข้อมูลเรียนรู้ ซึ่งจากกราฟทำให้สรุปได้ว่าขั้นตอนวิธี TBWC ใช้เวลาในการเรียนรู้มากกว่าขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means ในทุกชุดข้อมูลเรียนรู้ แต่เมื่อพิจารณากราฟในรูปที่ 4.4 แสดงเวลาที่ใช้ในการทำนายชุดข้อมูลเรียนรู้ กลับสรุปได้ว่าขั้นตอนวิธี TBWC ใช้เวลาในการทำนายน้อยกว่าขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means ในทุกชุดข้อมูลเรียนรู้ ยกเว้นชุดข้อมูล Internet Advertisement และ Libras Movement

จากข้อมูลทดลองทั้งหมด ทำให้สรุปได้ว่าขั้นตอนวิธีที่พัฒนาขึ้น (TBWC) สามารถให้ค่าความแม่นยำมากกว่าและใช้เวลาใช้ในการทำนายต่ำกว่าขั้นตอนต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means ในหลายชุดข้อมูลเรียนรู้ โดยการปรับปรุงประสิทธิภาพการทำนายมีการแปรผันตรงตามจำนวนกลุ่มของข้อมูล กล่าวคือ เมื่อมีจำนวนกลุ่มมาก การปรับปรุงประสิทธิภาพการทำนายก็จะทำได้ดีขึ้น

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้มีจุดมุ่งหมายเพื่อที่จะพัฒนาขั้นตอนวิธีการจำแนกประเภทข้อมูลได้อย่างมีประสิทธิภาพ โดยขั้นตอนการดำเนินงานวิจัยเริ่มจากการศึกษาค้นคว้าขั้นตอนวิธีต้นไม้ตัดสินใจต่างๆ ที่น่าสนใจ ได้แก่ ขั้นตอนวิธีต้นไม้ตัดสินใจ ID3 และขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และศึกษาขั้นตอนวิธีการจัดกลุ่มต่างๆ ที่น่าสนใจ ได้แก่ ขั้นตอนวิธีการจัดกลุ่ม k-means ขั้นตอนวิธีการจัดกลุ่ม hierarchical เพื่อเปรียบเทียบข้อดีและข้อเสียของแต่ละวิธี และนำเอาลักษณะเด่นของขั้นตอนวิธีต่างๆ มาประยุกต์ใช้เพื่อพัฒนาขั้นตอนวิธีการจำแนกประเภทที่มีประสิทธิภาพ โดยทดสอบกับชุดข้อมูลเรียนรู้จำนวน 5 ชุดข้อมูล เกณฑ์ที่ใช้ในการเปรียบเทียบประสิทธิภาพพิจารณาจากความแม่นยำในการจำแนกชุดข้อมูลทดสอบ ซึ่งใช้โมเดลที่สร้างจากชุดข้อมูลเรียนรู้ที่มีการใช้หลักการ k-fold cross-validation

การพัฒนาขั้นตอนวิธี Tree Bagging and Weighted Clustering Algorithm ในงานวิจัยนี้ ได้นำขั้นตอนวิธี Bagging โดยทำการสร้างโมเดลด้วยเทคนิคต้นไม้ตัดสินใจ C4.5 มาใช้ในการคัดเลือกคุณลักษณะ (Attribute Selection) แล้วจากนั้นจะใช้การจัดกลุ่ม k-means ในการจำแนกประเภทและทำนายข้อมูล

การเลือกพารามิเตอร์ที่ใช้ในขั้นตอนวิธีการจำแนกประเภทข้อมูลนั้นจะเลือกโดยใช้ค่าพารามิเตอร์ที่นำมาสร้างโมเดลด้วยชุดข้อมูลยืนยัน (Validate data) แล้วให้ค่าความแม่นยำสูงที่สุด และมีจำนวนโมเดลต่ำที่สุด ในแต่ละชุดข้อมูลเรียนรู้

จากผลการทดลองสามารถสรุปประเด็นต่างๆ ที่เกิดขึ้นได้ดังนี้

1) เมื่อทดสอบเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการจำแนกประเภทด้วยขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 การจัดกลุ่ม k-means และขั้นตอนวิธี TBWC พบว่า ขั้นตอนวิธี TBWC สามารถให้ค่าความแม่นยำสูงกว่าขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means ในทุกชุดข้อมูลเรียนรู้

2) เมื่อพิจารณาผลการเปรียบเทียบค่าความแม่นยำระว่างขั้นตอนวิธีต่างๆ พบว่า การปรับปรุงประสิทธิภาพการทำนายมีการแปรผันตรงตามจำนวนกลุ่ม กล่าวคือเมื่อมีจำนวนกลุ่มมาก การปรับปรุงประสิทธิภาพการทำนายจะทำได้อย่างดี

3) เมื่อพิจารณาตารางเปอร์เซ็นต์การลดทอนคุณลักษณะในขั้นตอนวิธี TBWC พบว่า ขั้นตอนวิธี TBWC สามารถลดทอนคุณลักษณะได้สูงสุดถึง 59.82% ในชุดข้อมูล Internet Advertisement ทั้งยังสามารถให้ผลลัพธ์สูงกว่าขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และขั้นตอนวิธีการจัดกลุ่ม k-means

4) เมื่อพิจารณาตารางเปอร์เซ็นต์การลดทอนคุณลักษณะในขั้นตอนวิธี TBWC พบว่า ขั้นตอนวิธี TBWC สามารถลดทอนคุณลักษณะได้เพียงเล็กน้อย ในชุดข้อมูล Cardiocography 1, Cardiocography 2 และ Libras Movement กล่าวคือสามารถลดทอนได้ 4.35% 4.35% และ 5.49% ตามลำดับ

5) เมื่อพิจารณาเวลาที่ใช้ในการเรียนรู้และทำนายชุดข้อมูลเรียนรู้ พบว่า เวลาที่ใช้ในการเรียนรู้ของขั้นตอนวิธี TBWC มีการใช้เวลาในการเรียนรู้มากกว่าขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means แต่ในส่วนของเวลาที่ใช้ในการทำนายของขั้นตอนวิธี TBWC มีการใช้เวลาในการทำนายน้อยกว่าขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means ในทุกชุดข้อมูลเรียนรู้ ยกเว้นชุดข้อมูล Internet Advertisement ช้ากว่าขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และเท่ากับการจัดกลุ่ม k-means ในชุดข้อมูล Libras Movement

จากผลการทดลองจึงแสดงให้เห็นว่า การนำขั้นตอนวิธี Bagging โดยทำการสร้างโมเดลด้วยเทคนิคต้นไม้ตัดสินใจ C4.5 มาใช้ในการคัดเลือกคุณลักษณะ และการใช้การถ่วงน้ำหนักให้กับคุณลักษณะในการจัดกลุ่ม k-means สามารถทำให้การจำแนกประเภทข้อมูลมีประสิทธิภาพที่ดีขึ้น

5.2 ข้อเสนอแนะ

ขั้นตอนวิธี TBWC มีปัญหาที่ต้องปรับปรุงแก้ไขเพิ่มเติมดังนี้

1) การปรับปรุงประสิทธิภาพการทำนายมีการแปรผันตรงตามจำนวนกลุ่ม ดังนั้นเมื่อทำงานร่วมกับชุดข้อมูลเรียนที่มีจำนวนกลุ่มน้อย ผลการเปรียบเทียบประสิทธิภาพการทำนายจะเพิ่มขึ้นเพียงเล็กน้อยเท่านั้น เนื่องจากเมื่อข้อมูลมีจำนวนกลุ่มน้อย ขั้นตอนวิธี TBWC จะพบปัญหาในการเลือกกลุ่มคลัสเตอร์ที่เหมาะสม

2) ใช้เวลาจำนวนมากในการเรียนรู้ชุดข้อมูลเรียนรู้ เมื่อเปรียบเทียบกับขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means เนื่องจากการเรียนรู้ของขั้นตอนวิธีนั้นจำเป็นต้องสร้างทั้งโมเดล Bagging ด้วยเทคนิคต้นไม้ตัดสินใจ C4.5 และคำนวณค่าจุดศูนย์กลางของคลัสเตอร์ (centroid)

3) การเลือกจุดศูนย์กลาง ของ Weighted Clustering ใช้หลักการเลือกจุดศูนย์กลางที่ใกล้ที่สุดและพบเป็นอันดับแรก ดังนั้นจึงมีโอกาสที่จะพบจุดศูนย์กลางที่ใกล้ที่สุดมากกว่า 1 จุด แต่คำตอบที่ถูกต้องเป็นจุดศูนย์กลางที่พบเป็นอันดับหลัง

เอกสารอ้างอิง

ดร.ปริญญา สงวนสัตย์, *"คู่มือ MATLAB ฉบับสมบูรณ์"*, ไอดีซี, 2010.

สุธรรม ศรีเกษม, *"MATLAB เพื่อการแก้ปัญหาทางวิศวกรรม"*, มหาวิทยาลัยรังสิต, 1997.

จารุทัศน์ วงษ์สันต์, *"MATLAB สำหรับแก้ปัญหาเชิงวิทยาศาสตร์และวิศวกรรม"*, ฟิสิกส์เซ็นเตอร์, 2001.

Shekhar R. Gaddam, Vir V. Phoha and Kiran S. Balagani, *"K-Means+ID3: A Novel Method for Supervised Anomaly Detection by Cascading K-Means Clustering and ID3 Decision Tree Learning Methods"*, IEEE Transactions on Knowledge and Data Engineering, pp. 345-354, March 2007.

Weizhao Guo, Jian Yin, Zhimin Yang, Xiaobo Yang and Li Huang, *"Exploring an Improved Decision Tree Based Weights"*, Fifth International Conference on Natural Computation, ICNC '09, Vol. 1, pp. 139-143, August 2009.

My Chau Tu, Dongil Shin and Dongkyoo Shin, *"A Comparative Study of Medical Data Classification Methods Based on Decision Tree and Bagging Algorithms"*, Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC '09, pp. 183-187, December 2009.

Jian Zhu and Hanshi Wang, *"An improved K-means clustering algorithm"*, The 2nd IEEE International Conference on Information Management and Engineering (ICIME), pp. 190-192, April 2010.

Xiaoping Qin, Shijue Zheng, Tingting He, Ming Zou and Ying Huang, *"Optimizated K-means algorithm and application in CRM system"*, International Symposium on Computer Communication Control and Automation (3CA), pp. 519-522, May 2010.

Parvesh Kumar and Siri Krishan Wasan, *"Comparative Analysis of k-mean Based Algorithms"*, IJCSNS International Journal of Computer Science and Network Security, VOL.10, No.4, April 2010.

Chen Jin, Luo De-lin and Mu Fen-xiang, *"An Improved ID3 Decision Tree Algorithm"*, Computer Science & Education, ICCSE '09. 4th International, pp. 127-130, 2009.

Boris Mirkin, *"Clustering for Data Mining"*, Taylor & Francis Group, 2005.

Jiawei Han and Micheline Kamber, *"Data mining: concepts and techniques, Second Edition"*, Morgan Kaufmann, 2005.

Oded Maimon and Lior Rokach, *"The Data Mining and Knowledge Discovery Handbook"*, Springer, 2005.

Daniel T. Larose, *"Discovering knowledge in data: an introduction to Data Mining"*, Wiley, 2005.

Michael W. Berry and Murray Browne, *"Lecture Note in Data Mining"*, World Scientific Publishing, 2006.

Lior Rokach and Oded Maimom, *"Data Mining With Decision Tree Theory and Applications"*, World Scientific Publishing. 2008.

Xingquan Zhu and lan Devidson, *"Knowledge discovery and data mining"*, Information science reference, 2007.

The MathWorks, *"Getting Started with MATLAB"*, The MathWorks Inc, 2004.

The MathWorks, *"MATLAB Mathematics"*, The MathWorks Inc, 2004.

The MathWorks, *"MATLAB Programming"*, The MathWorks Inc, 2004.

The MathWorks, *"Statistics Toolbox™ User's Guide"*, The MathWorks Inc, 2010.

David G. Stork and Elad Yom-Tov, *"Appendix to the Computer Manual in MATLAB to accompany Pattern Classification, Second Edition"*, Paperback, 2004.

[Online].Available : www.mat.univie.ac.at/~neum/statdat.html

[Online].Available : archive.ics.uci.edu/ml/datasets.html

[Online].Available : www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html

[Online].Available : teacher.en.rmutt.ac.th/ktw/04-720-101/intro_matlab.html

[Online].Available : www.mathworks.com/products/statistics/

[Online].Available : www.mathtools.net/MATLAB/Neural_Networks/index.html

[Online].Available : www.regim.org/download_9_Classification-Toolbox.html

[Online].Available : www.tech.plym.ac.uk/spmc/links/matlab/matlab_toolbox.html

[Online].Available : www.dictall.com/

[Online].Available : www.tech.plym.ac.uk/spmc/links/matlab/matlab_trees.html

[Online].Available : www.cs.cmu.edu/~juny/MILL/index.html

ภาคผนวก ก.

ข้อมูลผลการทดลอง

ก.1 ข้อมูลผลการทดลองเพื่อเลือกพารามิเตอร์ต่างๆ ด้วยชุดข้อมูลยืนยัน (Validate data)

พารามิเตอร์ที่ใช้ในขั้นตอนวิธีจำแนกประเภทข้อมูลแต่ละชุดข้อมูลเรียนรู้ ประกอบไปด้วยกัน 2 ค่า คือ ค่าจำนวนโมเดล (n) คือจำนวนของโมเดลในการสร้างโมเดลจำแนกประเภทด้วยขั้นตอนวิธี Bagging ด้วยต้นไม้ตัดสินใจ C4.5 และค่าจำนวนคลัสเตอร์ (k) คือจำนวนของคลัสเตอร์ในการจัดกลุ่มข้อมูลด้วยขั้นตอนวิธีการจัดกลุ่ม k-means เงื่อนไขการเลือกพารามิเตอร์ที่ใช้ในขั้นตอนวิธีการจำแนกประเภทข้อมูลนั้นจะเลือกโดยใช้ค่าพารามิเตอร์ที่นำมาสร้างโมเดลด้วยชุดข้อมูลยืนยัน (Validate data) แล้วให้ค่าความแม่นยำสูงที่สุด และมีจำนวนโมเดลต่ำที่สุด ในแต่ละชุดข้อมูลเรียนรู้

ก.1.1 ชุดข้อมูลเรียนรู้ Cardiocography 1

ชุดข้อมูลยืนยันมีจำนวนข้อมูล 213 ตัวอย่าง จำแนกออกเป็น 3 กลุ่ม คือ

- normal (จำนวน 71 ตัวอย่าง)
- suspect (จำนวน 71 ตัวอย่าง)
- pathologic (จำนวน 71 ตัวอย่าง)

ตารางที่ ก.1 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลยืนยันของชุดข้อมูล Cardiocography 1

จำนวนคลัสเตอร์ (k) / จำนวนโมเดล (n)	10	20	30	40	50	60	70	80	90	100
10	67.29	75.70	82.24	86.92	85.05	91.59	91.59	97.20	97.20	98.13
20	69.16	74.77	80.37	86.92	85.05	89.72	93.46	93.46	96.26	98.13
30	73.83	78.50	82.24	88.79	88.79	88.79	91.59	95.33	98.13	99.07
40	70.09	79.44	83.18	83.18	85.98	89.72	96.26	96.26	94.39	**100.00**
50	68.22	79.44	76.64	87.85	87.85	90.65	95.33	96.26	98.13	98.13
60	68.22	80.37	85.05	83.18	92.52	91.59	93.46	99.07	96.26	99.07
70	73.83	83.18	80.37	85.98	90.65	92.52	93.46	94.39	98.13	99.07
80	71.96	78.50	78.50	83.18	85.05	92.52	91.59	95.33	95.33	100.00
90	71.96	72.90	81.31	86.92	91.59	91.59	95.33	94.39	96.26	99.07
100	76.64	80.37	87.85	87.85	85.05	92.52	90.65	96.26	94.39	99.07

จากตารางที่ ก.1 เมื่อพิจารณาตามเงื่อนไขการเลือกพารามิเตอร์ที่กล่าวไว้จะได้ค่าจำนวนโมเดล (n) เท่ากับ 40 โมเดล และค่าจำนวนคลัสเตอร์เท่ากับ 100 คลัสเตอร์ ซึ่งให้ค่าความแม่นยำ 100.00% ในชุดข้อมูลยืนยันของชุดข้อมูล Cardiocography 1

ก.1.2 ชุดข้อมูลเรียนรู้ Cardiocography 2

ชุดข้อมูลยืนยันมีจำนวนข้อมูล 210 ตัวอย่าง จำแนกออกเป็น 10 กลุ่ม คือ

- FHR pattern class code is 1 (จำนวน 21 ตัวอย่าง)
- FHR pattern class code is 2 (จำนวน 21 ตัวอย่าง)
- FHR pattern class code is 3 (จำนวน 21 ตัวอย่าง)
- FHR pattern class code is 4 (จำนวน 21 ตัวอย่าง)
- FHR pattern class code is 5 (จำนวน 21 ตัวอย่าง)
- FHR pattern class code is 6 (จำนวน 21 ตัวอย่าง)
- FHR pattern class code is 7 (จำนวน 21 ตัวอย่าง)
- FHR pattern class code is 8 (จำนวน 21 ตัวอย่าง)
- FHR pattern class code is 9 (จำนวน 21 ตัวอย่าง)
- FHR pattern class code is 10 (จำนวน 21 ตัวอย่าง)

ตารางที่ ก.2 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลยืนยันของชุดข้อมูล Cardiocography 2

จำนวนคลัสเตอร์ (k) / จำนวนโมเดล (n)	10	20	30	40	50	60	70	80	90	100
10	58.10	69.52	77.14	84.76	80.00	86.67	94.29	94.29	97.14	99.05
20	50.48	73.33	78.10	77.14	80.95	82.86	91.43	96.19	97.14	**100.00**
30	58.10	72.38	71.43	74.29	81.90	88.57	88.57	95.24	95.24	100.00
40	61.90	60.00	74.29	77.14	87.62	89.52	89.52	96.19	99.05	100.00
50	59.05	63.81	73.33	78.10	80.95	89.52	88.57	96.19	97.14	98.10
60	59.05	62.86	68.57	79.05	80.95	87.62	95.24	94.29	97.14	100.00
70	60.95	69.52	75.24	77.14	86.67	84.76	89.52	91.43	96.19	98.10
80	59.05	71.43	68.57	76.19	80.00	86.67	95.24	95.24	99.05	100.00
90	57.14	63.81	74.29	79.05	84.76	89.52	85.71	91.43	96.19	99.05
100	50.48	71.43	71.43	77.14	80.00	87.62	89.52	93.33	95.24	100.00

จากตารางที่ ก.2 เมื่อพิจารณาตามเงื่อนไขการเลือกพารามิเตอร์ที่กล่าวไว้จะได้ค่าจำนวนโมเดล (n) เท่ากับ 20 โมเดล และค่าจำนวนคลัสเตอร์เท่ากับ 100 คลัสเตอร์ ซึ่งให้ค่าความแม่นยำ 100.00% ในชุดข้อมูลยืนยันของชุดข้อมูล Cardiocography 2

ก.1.3 ชุดข้อมูลเรียนรู้ Internet Advertisements

ชุดข้อมูลยืนยันมีจำนวนข้อมูล 328 ตัวอย่าง จำแนกออกเป็น 2 กลุ่ม คือ

- advertisement (จำนวน 164 ตัวอย่าง)
- not advertisement (จำนวน 164 ตัวอย่าง)

ตารางที่ ก.3 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลยืนยันของชุดข้อมูล Internet Advertisements

จำนวนคลัสเตอร์ (k) / จำนวนโมเดล (n)	10	20	30	40	50	60	70	80	90	100
10	82.32	79.88	90.24	92.07	93.90	95.12	96.95	96.95	95.73	**99.39**
20	70.73	89.02	84.15	87.20	95.73	93.90	93.29	95.12	98.17	99.39
30	69.51	74.39	87.80	90.85	98.17	93.29	98.17	95.73	96.95	98.17
40	82.32	88.41	85.37	90.24	92.68	93.90	96.95	96.95	98.78	96.95
50	86.59	81.10	89.02	93.90	95.12	95.12	94.51	96.95	98.78	97.56
60	76.22	82.93	82.32	89.02	92.68	93.90	93.90	97.56	95.73	98.17
70	83.54	81.10	87.20	95.73	96.34	94.51	95.73	98.17	98.17	96.95
80	84.15	79.27	86.59	90.85	92.07	93.29	95.73	93.29	97.56	99.39
90	77.44	89.02	82.32	93.29	90.85	92.68	93.90	96.34	96.95	99.39
100	78.05	85.37	91.46	90.24	92.68	96.34	94.51	96.34	95.12	98.78

จากตารางที่ ก.3 เมื่อพิจารณาตามเงื่อนไขการเลือกพารามิเตอร์ที่กล่าวไว้จะได้ค่าจำนวนโมเดล (n) เท่ากับ 10 โมเดล และค่าจำนวนคลัสเตอร์เท่ากับ 100 คลัสเตอร์ ซึ่งให้ค่าความแม่นยำ 99.39% ในชุดข้อมูลยืนยันของชุดข้อมูล Internet Advertisements

ก.1.4 ชุดข้อมูลเรียนรู้ Libras Movement

ชุดข้อมูลยืนยันมีจำนวนข้อมูล 45 ตัวอย่าง จำแนกออกเป็น 15 กลุ่ม คือ

- curved swing (จำนวน 3 ตัวอย่าง)
- horizontal swing (จำนวน 3 ตัวอย่าง)
- vertical swing (จำนวน 3 ตัวอย่าง)
- anti-clockwise arc (จำนวน 3 ตัวอย่าง)
- clockwise arc (จำนวน 3 ตัวอย่าง)
- circle (จำนวน 3 ตัวอย่าง)
- horizontal straight-line (จำนวน 3 ตัวอย่าง)
- vertical straight-line (จำนวน 3 ตัวอย่าง)
- horizontal zigzag (จำนวน 3 ตัวอย่าง)
- vertical zigzag (จำนวน 3 ตัวอย่าง)
- horizontal wavy (จำนวน 3 ตัวอย่าง)
- vertical wavy (จำนวน 3 ตัวอย่าง)
- face-up curve (จำนวน 3 ตัวอย่าง)
- face-down curve (จำนวน 3 ตัวอย่าง)
- tremble (จำนวน 3 ตัวอย่าง)

ตารางที่ ก.4 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลยืนยันของชุดข้อมูล Libras Movement

จำนวนคลัสเตอร์ (k) / จำนวนโมเดล (n)	5	10	15	20
5	52.17	78.26	82.61	**100.00**
10	43.48	73.91	86.96	100.00
15	43.48	73.91	91.30	95.65
20	52.17	78.26	86.96	95.65

จากตารางที่ ก.4 เมื่อพิจารณาตามเงื่อนไขการเลือกพารามิเตอร์ที่กล่าวไว้จะได้ค่าจำนวนโมเดล (n) เท่ากับ 5 โมเดล และค่าจำนวนคลัสเตอร์เท่ากับ 20 คลัสเตอร์ ซึ่งให้ค่าความแม่นยำ 100.00% ในชุดข้อมูลยืนยันของชุดข้อมูล Libras Movement

ก.1.5 ชุดข้อมูลเรียนรู้ Multiple Features

ชุดข้อมูลยืนยันมีจำนวนข้อมูล 200 ตัวอย่าง จำแนกออกเป็น 10 กลุ่ม คือ

- '0' (จำนวน 20 ตัวอย่าง)
- '1' (จำนวน 20 ตัวอย่าง)
- '2' (จำนวน 20 ตัวอย่าง)
- '3' (จำนวน 20 ตัวอย่าง)
- '4' (จำนวน 20 ตัวอย่าง)
- '5' (จำนวน 20 ตัวอย่าง)
- '6' (จำนวน 20 ตัวอย่าง)
- '7' (จำนวน 20 ตัวอย่าง)
- '8' (จำนวน 20 ตัวอย่าง)
- '9' (จำนวน 20 ตัวอย่าง)

ตารางที่ ก.5 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลยืนยันของชุดข้อมูล Multiple Features

จำนวนคลัสเตอร์ (k) / จำนวนโมเดล (n)	10	20	30	40	50	60	70	80	90	100
10	86.00	90.00	95.00	94.00	98.00	98.00	99.00	**100.00**	100.00	100.00
20	78.00	95.00	93.00	95.00	95.00	99.00	98.00	98.00	99.00	100.00
30	86.00	94.00	88.00	97.00	97.00	98.00	99.00	99.00	99.00	100.00
40	83.00	92.00	93.00	94.00	95.00	95.00	97.00	97.00	99.00	100.00
50	79.00	85.00	91.00	94.00	95.00	98.00	98.00	100.00	100.00	100.00
60	85.00	95.00	95.00	95.00	95.00	96.00	99.00	100.00	100.00	100.00
70	86.00	95.00	91.00	92.00	97.00	97.00	99.00	98.00	100.00	100.00
80	68.00	89.00	93.00	94.00	96.00	97.00	96.00	98.00	98.00	100.00
90	72.00	94.00	93.00	94.00	97.00	97.00	96.00	98.00	99.00	100.00
100	90.00	86.00	92.00	96.00	96.00	100.00	96.00	99.00	100.00	100.00

จากตารางที่ ก.5 เมื่อพิจารณาตามเงื่อนไขการเลือกพารามิเตอร์ที่กล่าวไว้จะได้ค่าจำนวนโมเดล (n) เท่ากับ 10 โมเดล และค่าจำนวนคลัสเตอร์เท่ากับ 80 คลัสเตอร์ ซึ่งให้ค่าความแม่นยำ 100.00% ในชุดข้อมูลยืนยันของชุดข้อมูล Multiple Features

ก.2 ข้อมูลผลการทดลอง

การวิเคราะห์เปรียบเทียบประสิทธิภาพของการจำแนกประเภทนั้น จะใช้ขั้นตอนวิธีการจำแนกประเภทที่สำคัญ 2 วิธี ได้แก่ ขั้นตอนวิธีต้นไม้ตัดสินใจ C4.5 และการจัดกลุ่ม k-means ซึ่งนำไปเปรียบเทียบกับขั้นตอนวิธีการจำแนกประเภทที่พัฒนาขึ้นมาโดยใช้ชื่อว่าขั้นตอนวิธี Tree Bagging and Weighted Clustering Algorithm (TBWC) ทดสอบกับชุดข้อมูลเรียนรู้ที่เลือกมาจำนวน 5 ชุด และใช้เทคนิค 10-fold cross validation เป็นวิธีการวัดประสิทธิภาพของขั้นตอนวิธีที่ใช้ในการทำนายตัวอย่างของโมเดล

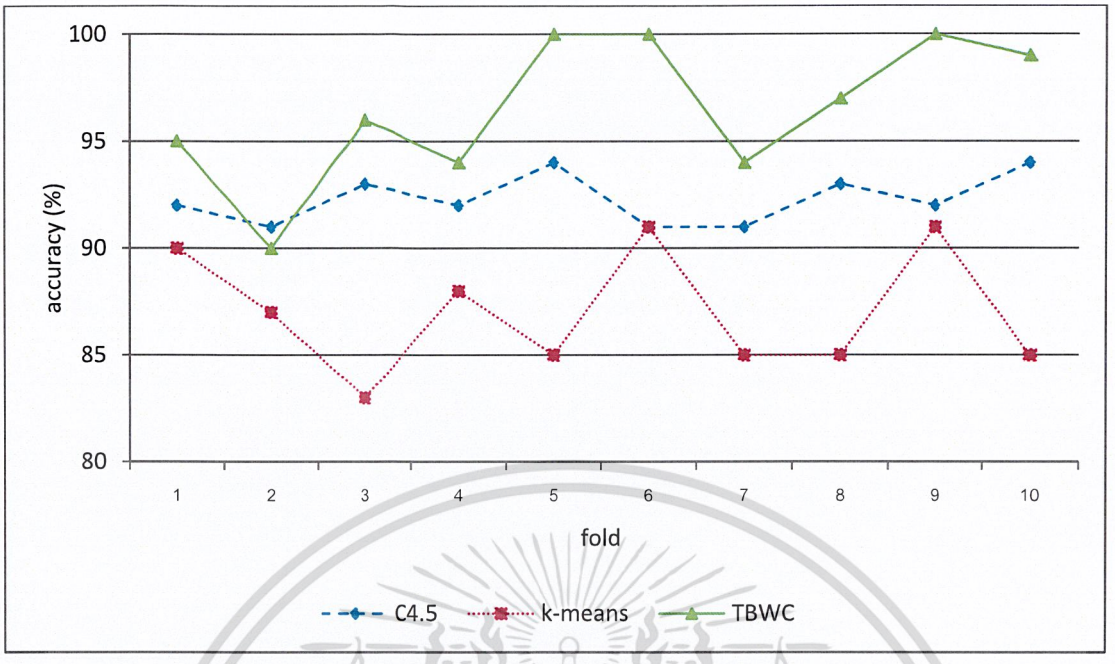
ก.2.1 ชุดข้อมูลเรียนรู้ Cardiocography 1

มีจำนวนข้อมูล 2,126 ตัวอย่าง จำแนกออกเป็น 3 กลุ่ม

ตารางที่ ก.6 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลเรียนรู้ของชุดข้อมูล Cardiocography 1

fold	C4.5	k-means	TBWC
1	91.98%	89.62%	94.81%
2	91.08%	87.32%	90.14%
3	93.40%	83.02%	96.23%
4	92.49%	87.79%	93.90%
5	94.37%	84.98%	99.53%
6	91.04%	90.57%	99.53%
7	91.55%	85.45%	94.84%
8	92.92%	85.38%	97.17%
9	92.49%	90.61%	99.53%
10	94.37%	84.98%	99.06%
Average	92.56%	86.97%	96.47%

จากข้อมูลในตารางที่ ก.6 เมื่อนำข้อมูลที่ได้มาสร้างเป็นกราฟแสดงประสิทธิภาพของขั้นตอนวิธีจำแนกประเภท สามารถแสดงได้ดังรูปที่ ก.1



รูปที่ ก.1 กราฟแสดงประสิทธิภาพในการจำแนกประเภทของขั้นตอนวิธีต่างๆ ของชุดข้อมูล Cardiocography 1

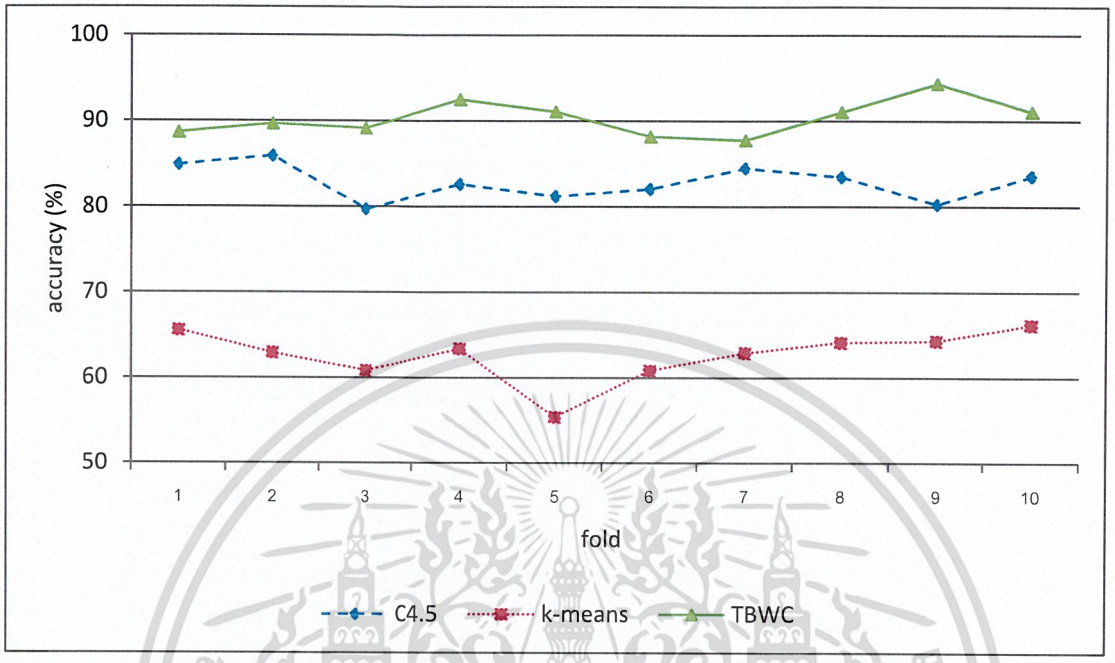
ก.2.2 ชุดข้อมูลเรียนรู้ Cardiocography 2

มีจำนวนข้อมูล 2,126 ตัวอย่าง จำแนกออกเป็น 10 กลุ่ม

ตารางที่ ก.7 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลเรียนรู้ของชุดข้อมูล Cardiocography 2

fold	C4.5	k-means	TBWC
1	84.91%	65.57%	88.68%
2	85.92%	62.91%	89.67%
3	79.72%	60.85%	89.15%
4	82.63%	63.38%	92.49%
5	81.22%	55.40%	91.08%
6	82.08%	60.85%	88.21%
7	84.51%	62.91%	87.79%
8	83.49%	64.15%	91.04%
9	80.28%	64.32%	94.37%
10	83.57%	66.20%	91.08%
Average	82.83%	62.65%	90.35%

จากข้อมูลในตารางที่ ก.7 เมื่อนำข้อมูลที่ได้มาสร้างเป็นกราฟแสดงประสิทธิภาพของขั้นตอนวิธีจำแนกประเภท สามารถแสดงได้ดังรูปที่ ก.2



รูปที่ ก.2 กราฟแสดงประสิทธิภาพในการจำแนกประเภทของขั้นตอนวิธีต่างๆ ของชุดข้อมูล Cardiocography 2

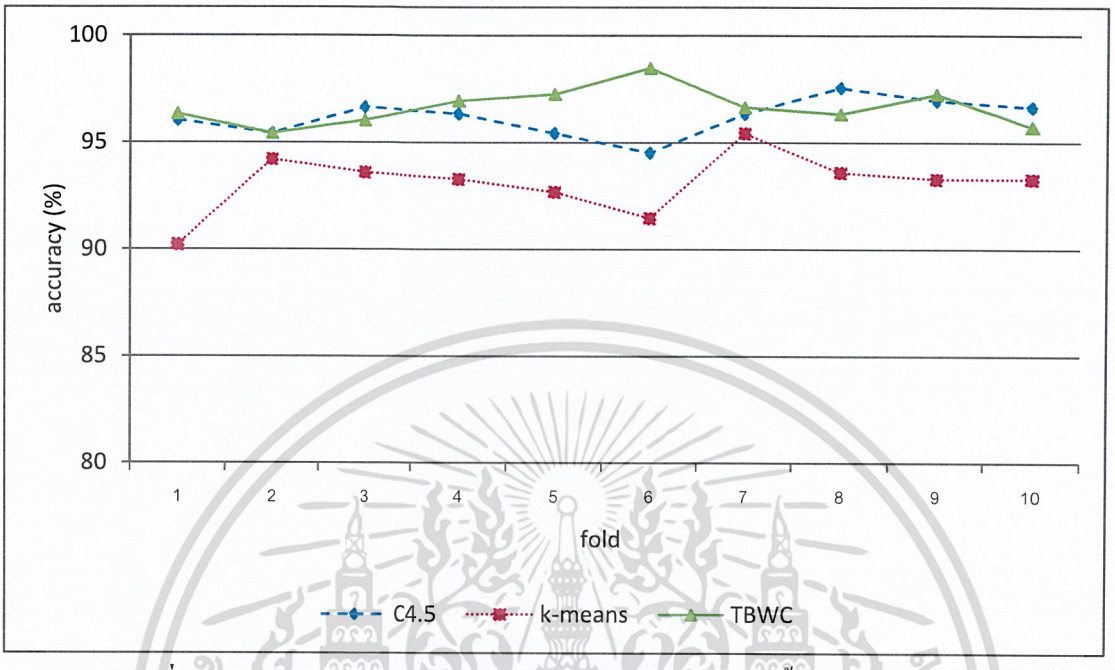
ก.2.3 ชุดข้อมูลเรียนรู้ Internet Advertisements

มีจำนวนข้อมูล 3,279 ตัวอย่าง จำแนกเป็น 2 กลุ่ม

ตารางที่ ก.8 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลเรียนรู้ของชุดข้อมูล Internet Advertisements

fold	C4.5	k-means	TBWC
1	96.04%	90.21%	96.33%
2	95.43%	94.21%	95.43%
3	96.65%	93.60%	96.04%
4	96.34%	93.29%	96.95%
5	95.43%	92.68%	97.26%
6	94.52%	91.46%	98.48%
7	96.34%	95.43%	96.65%
8	97.56%	93.60%	96.34%
9	96.95%	93.29%	97.26%
10	96.65%	93.29%	95.73%
Average	96.19%	93.11%	96.65%

จากข้อมูลในตารางที่ ก.8 เมื่อนำข้อมูลที่ได้มาสร้างเป็นกราฟแสดงประสิทธิภาพของขั้นตอนวิธีจำแนกประเภท สามารถแสดงได้ดังรูปที่ ก.3



รูปที่ ก.3 กราฟแสดงประสิทธิภาพในการจำแนกประเภทของขั้นตอนวิธีต่างๆ ของชุดข้อมูล Internet Advertisements

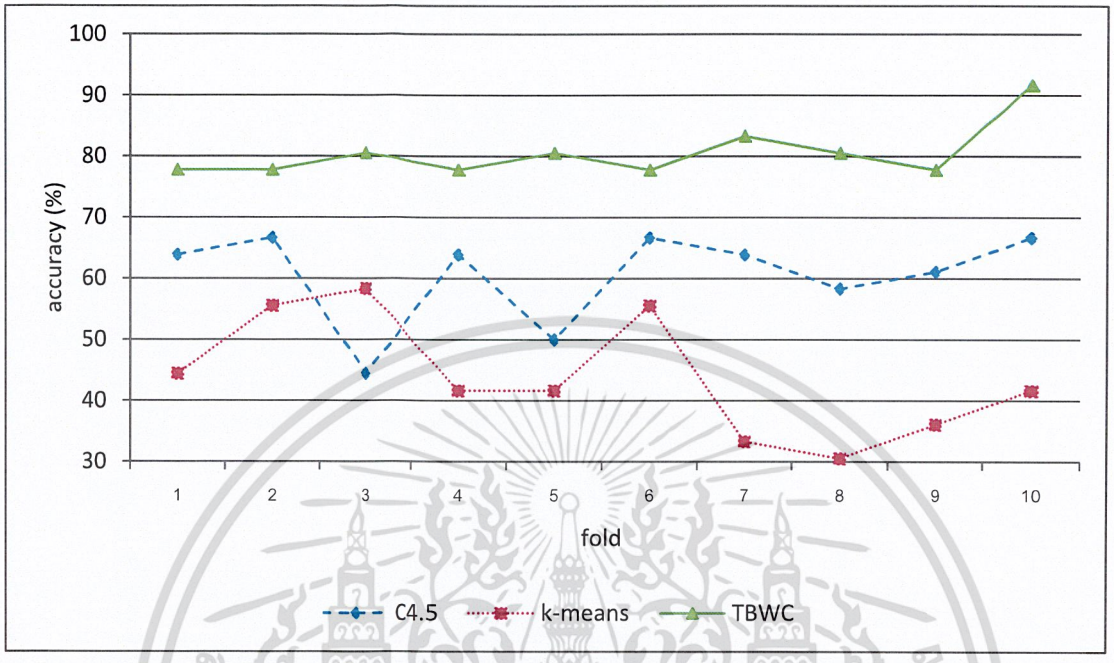
ก.2.4 ชุดข้อมูลเรียนรู้ Libras Movement

มีจำนวนข้อมูล 360 ตัวอย่าง จำแนกออกเป็น 15 กลุ่ม

ตารางที่ ก.9 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลเรียนรู้ของชุดข้อมูล Libras Movement

fold	C4.5	k-means	TBWC
1	63.89%	44.44%	77.78%
2	66.67%	55.56%	77.78%
3	44.44%	58.33%	80.56%
4	63.89%	41.67%	77.78%
5	50.00%	41.67%	80.56%
6	66.67%	55.56%	77.78%
7	63.89%	33.33%	83.33%
8	58.33%	30.56%	80.56%
9	61.11%	36.11%	77.78%
10	66.67%	41.67%	91.67%
Average	60.56%	43.89%	80.56%

จากข้อมูลในตารางที่ ก.9 เมื่อนำข้อมูลที่ได้มาสร้างเป็นกราฟแสดงประสิทธิภาพของขั้นตอนวิธีจำแนกประเภท สามารถแสดงได้ดังรูปที่ ก.4



รูปที่ ก.4 กราฟแสดงประสิทธิภาพในการจำแนกประเภทของขั้นตอนวิธีต่างๆ ของชุดข้อมูล Libras Movement

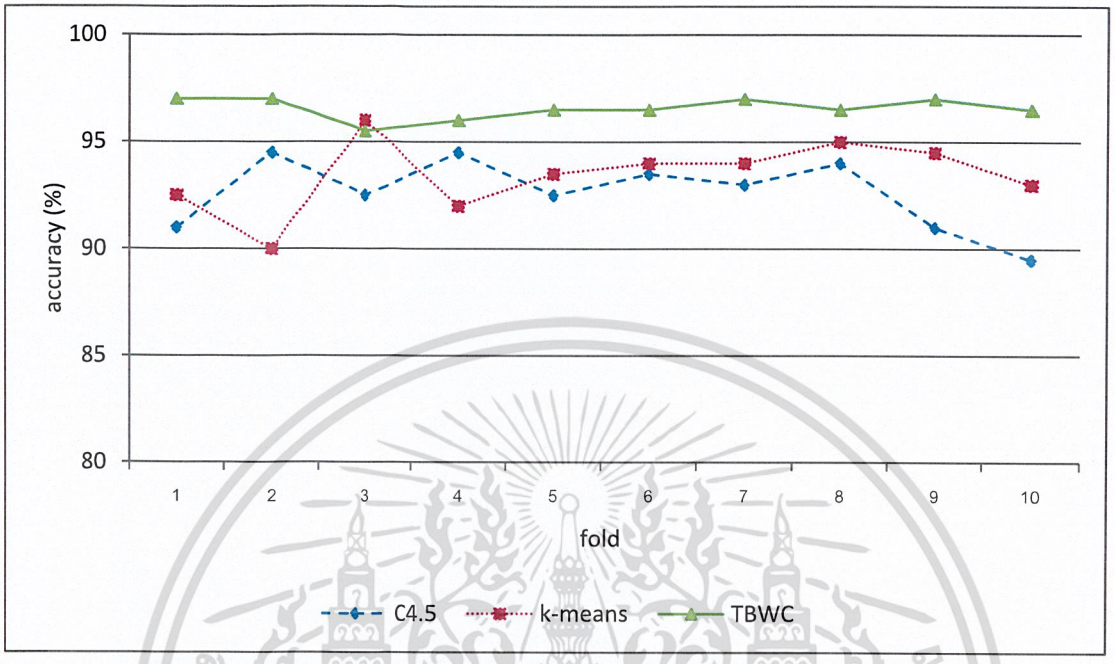
ก.2.5 ชุดข้อมูลเรียนรู้ Multiple Feature

มีจำนวนข้อมูล 2,000 ตัวอย่าง จำแนกออกเป็น 10 กลุ่ม

ตารางที่ ก.10 ความแม่นยำในการจำแนกประเภทของชุดข้อมูลเรียนรู้ของชุดข้อมูล Multiple Features

fold	C4.5	k-means	TBWC
1	91.00%	92.50%	97.00%
2	94.50%	90.00%	97.00%
3	92.50%	96.00%	95.50%
4	94.50%	92.00%	96.00%
5	92.50%	93.50%	96.50%
6	93.50%	94.00%	96.50%
7	93.00%	94.00%	97.00%
8	94.00%	95.00%	96.50%
9	91.00%	94.50%	97.00%
10	89.50%	93.00%	96.50%
Average	92.60%	93.45%	96.55%

จากข้อมูลในตารางที่ ก.10 เมื่อนำข้อมูลที่ได้มาสร้างเป็นกราฟแสดงประสิทธิภาพของขั้นตอนวิธีจำแนกประเภท สามารถแสดงได้ดังรูปที่ ก.5



รูปที่ ก.5 กราฟแสดงประสิทธิภาพในการจำแนกประเภทของขั้นตอนวิธีต่างๆ ของชุดข้อมูล Multiple Features

ภาคผนวก ข.

รหัสต้นฉบับภาษา MATLAB

ขั้นตอนวิธี Tree Bagging and Weighted Clustering Algorithm ได้มีการพัฒนาขั้นตอนวิธีด้วยโปรแกรม MATLAB Version 7.10.0.499 (R2010a) 32-bit โดยใช้ Toolbox ที่สำคัญ 2 ชนิดคือ Statistics Toolbox และ MATLABArsenal Toolbox

รูปแบบการใช้งานสามารถแบ่งออกเป็น 2 ประเภทคือ

1) รูปแบบใช้งานในการทดลอง

2) รูปแบบใช้งานทั่วไป

ข.1 รหัสต้นแบบภาษา MATLAB รูปแบบใช้งานในการทดลอง

รูปแบบการเรียกใช้โปรแกรมในส่วนของการทดลองมีการทำงานบน MATLABArsenal Toolbox ดังนั้นจะต้องทำการ Set Path แฟ้มชื่อ "MATLABArsenal" ก่อนที่จะใช้งาน

การเรียกคำสั่งใช้งาน สามารถเรียกได้โดยใช้คำสั่งต่อไปนี้

```
>> Arsenal('classify -t DATASET_NAME.txt --
   cross_validate -t 10 -- TBWC_classify -n 40 -k 100');
```

โดยคำสั่งใช้งาน จะมีพารามิเตอร์ที่เกี่ยวข้องดังนี้

ตารางที่ ข.1 (ก) พารามิเตอร์ของคำสั่งใช้งานรูปแบบใช้งานในการทดลอง

ชื่อพารามิเตอร์	ความหมาย	ตัวอย่าง
-t	ชื่อชุดข้อมูลเรียนรู้	Cardiocography-1.txt
--	รูปแบบการเรียนรู้	train_test_validate, cross_validate, train_only, test_only
-t	จำนวนรอบของ k-fold cross-validation	10
--	ขั้นตอนจำแนกวิธีที่เลือกใช้	TBWC_classify, kmeans_classfy
-n	จำนวนโมเดลในการสร้างโมเดลจำแนกประเภทด้วยขั้นตอนวิธี Bagging ด้วยต้นไม้ตัดสินใจ C4.5	40
-k	จำนวนคลัสเตอร์ในการจัดกลุ่มข้อมูลด้วยขั้นตอนวิธีการจัดกลุ่ม k-means	100

รหัสต้นฉบับภาษา MATLAB รูปแบบใช้งานในการทดลอง แสดงได้ดังรูปที่ ข.1

```
% TBWC_classify.m
%
% This source code is Copyright(c) 2010-2011
% by Chinnapat Kaewchinporn & Nattakan Vonsuchoto
%
% Tree Bagging and Weighted Clustering Algorithm
%
% Parameters:
% para: parameters
% 1. n: number of trees, default: 50
% 2. m: minimum number of observation, default: 1
% 3. c: category features, default: 0
% 4. k: number of cluster, default: 5
% 5. d: distance measure, default: cityblock
% 6. r: number of time to repeat the clustering, default: 5
% X_train: training examples
% Y_train: training labels
% X_test: testing examples
% Y_test: testing labels
% num_class: number of classes
% class_set: set of class labels such as [1,-1], the first one is the
% positive label
%
% Require functions:
% ParseParameter, GetModelFilename
function [Y_compute, Y_prob] = TBWC_classify(para, X_train, Y_train, X_test, Y_test, num_class, class_set)
p = str2num(char(ParseParameter(para, {'-n';'-m';'-c';'-k';'-r'}, {'100';'1';'0';'22';'5'})));
ntrees = p(1);
minleaf = p(2);
cat = p(3);
k = p(4);
replicate = p(5);
% Parameter estimation
fprintf('C4.5 Decision Tree Phase\n');
if(cat > 0)
    t = TreeBagger(ntrees,X_train,Y_train,'method','c','minleaf',minleaf,'cat',cat);
else
    t = TreeBagger(ntrees,X_train,Y_train,'method','c','minleaf',minleaf);
end

numofattr = size(X_train,2);
wa = zeros(1,numofattr);
```

รูปที่ ข.1 (ก) รหัสต้นฉบับภาษา MATLAB รูปแบบใช้งานในการทดลอง

```
for i=1:ntrees
%    fprintf('Tree number: %d\n',i);
     wat = zeros(1,numofattr);
     subt = t.Trees{i};
     subtheight  = heightOf(subt);
     [~,attrcut] = cutvar(subt);
     for j=1:subt.numnodes
        usedattr = attrcut(j,1);
%       fprintf('    Attribute: %d \t',usedattr);
        if(usedattr > 0 && wat(usedattr) == 0)
             wat(usedattr) = (subtheight-levelOf(subt,j)+1)
/(subtheight+1);
%            fprintf('Level: %.4f \t',levelOf(subt,j));
%            fprintf('Weight: %.4f\n',wat(usedattr));
             wa(usedattr) =  wa(usedattr) + wat(usedattr);
%       else
%            fprintf('Leaf\n');
        end;
     end;
end;
wa = wa./ntrees;
wa = (wa-min(wa))/(max(wa)-min(wa));

cleanattr = 0;
for i=numofattr:-1:1
     if (wa(i) == 0)
%         disp(i);
          cleanattr = cleanattr + 1;
     end;
end;
fprintf('Number of Clean Attribute: %d\n',cleanattr);
% disp(size(X_train));
% disp(size(Xt_train));
fprintf('k-means Clustering Phase\n');
% disp(centroids);
% centroids is center of cluster
fprintf('Learning Phase\n');
[idx,centroids] = weight_kmeans(X_train,k,'distance','city'
,'emptyaction','singleton','replicates',replicate,
'weight',wa);
% centroids is center of cluster
centroids_class = zeros(k,1);
% centroids_class is class of centroid
for i=1:k
     class_indices = (idx == i);
     class = Y_train(class_indices,:);
     centroids_class(i,1) = mode(class);
end
```

รูปที่ ข.1 (ข) รหัสต้นฉบับภาษา MATLAB รูปแบบใช้งานในการทดลอง

```
fprintf('Testing Phase\n');
D = distfun(X_test, centroids, wa, 'cityblock');
[~,Y_cluster] = min(D,[],2);

sumD = sum(D,2);
Y_compute = centroids_class(Y_cluster,1);
Y_prob = (sumD - D(Y_cluster))./sumD;
fprintf('\n');

function level = levelOf(t, nodeind)
parentOfi = t.parent(nodeind);
if(parentOfi ~= 0),
   level = levelOf(t,parentOfi) + 1;
else
   level = 0;
end

function height = heightOf(t)
nodeindex = t.numnodes;
height = levelOf(t,nodeindex);

function D = distfun(X, C, W, dist)
% DISTFUN Calculate point to cluster centroid distances.
[n,p] = size(X);
D = zeros(n,size(C,1));
nclusts = size(C,1);
switch dist
case 'sqeuclidean'
    for i = 1:nclusts
        D(:,i) = W(1,1).*((X(:,1)-C(i,1).^2);
        for j = 2:p
            D(:,i) = D(:,i)+W(1,j).*((X(:,j)-C(i,j)).^2);
        end
        % D(:,i) = sum((X - C(repmat(i,n,1),:)).^2, 2);
    end
case 'cityblock'
    for i = 1:nclusts
        D(:,i) = W(1,1).*(abs(X(:,1) - C(i,1)));

        for j = 2:p
            D(:,i) = D(:,i)+W(1,j).*(abs(X(:,j)-C(i,j)));
        end
        % disp(D);
        % D(:,i) = sum(abs(X - C(repmat(i,n,1),:)), 2);
    end
end
```

รูปที่ ข.1 (ค) รหัสต้นฉบับภาษา MATLAB รูปแบบใช้งานในการทดลอง

รูปแบบของข้อมูลนำเข้า (Input Formats) สามารถทำได้ 2 รูปแบบ

1) แต่ละแถวแทนตัวอย่าง 1 ตัวอย่าง ตัวเลขสุดท้ายคือค่ากลุ่ม ตัวเลขที่เหลือคือค่าคุณลักษณะ

```
<line> = <value>,<value>, … ,<value>,<target>
<target> = <integer>
<value> = <float or integer>
```

รูปที่ ข.2 รูปแบบของข้อมูลนำเข้า แบบที่ 1

```
0, 0, 0.40, 0.25, 0.10, 0.40, 0.25, 0.10, 0
0, 1, 0.10, 0.75, 0.05, 0.05, 0.32, 0.05, 0
0, 0, 0.05, 0.32, 0.05, 0.05, 0.80, 0.05, 1
0, 0, 0.10, 0.32, 0.10, 0.05, 0.32, 0.05, 1
0, 0, 0.15, 0.20, 0.05, 0.40, 0.25, 0.10, 0
0, 0, 0.05, 0.80, 0.05, 0.05, 0.80, 0.05, 0
```

รูปที่ ข.3 ตัวอย่างข้อมูลนำเข้า แบบที่ 1

2) แต่ละแถวแทนตัวอย่าง 1 ตัวอย่าง ตัวเลขแรกคือค่ากลุ่ม ค่าคุณลักษณะแสดงด้วยรูปแบบ “ลำดับคุณลักษณะ:ค่าคุณลักษณะ”

```
<line> = <target>, <feature>:<value>, <feature>:<value>, … ,<feature>:<value>
<target> = <integer>
<feature> = <integer> | “qid”
<value> = <float or integer>
```

รูปที่ ข.4 รูปแบบของข้อมูลนำเข้า แบบที่ 2

0, 1:0, 2:0.00, 3:0.25, 4:0.10		0, 3:0.25, 4:0.10
0, 1:0, 2:0.10, 3:0.75, 4:0.00		0, 2:0.10, 3:0.75
1, 1:0, 2:0.00, 3:0.32, 4:0.00	หรือ	1, 3:0.32
1, 1:1, 2:0.00, 3:0.00, 4:0.00		1, 1:1
0, 1:0, 2:0.00, 3:0.20, 4:0.00		0, 3:0.20
0, 1:0, 2:0.00, 3:0.80, 4:0.05		0, 3:0.80, 4:0.05

รูปที่ ข.5 ตัวอย่างข้อมูลนำเข้า แบบที่ 2

ข.2 รหัสต้นแบบภาษา MATLAB รูปแบบใช้งานทั่วไป

รูปแบบใช้งานทั่วไป ใช้ในการทำนายตัวอย่างสำหรับแก้ปัญหาการจำแนกประเภทข้อมูล การเรียกคำสั่งใช้งาน สามารถเรียกได้โดยใช้คำสั่งต่อไปนี้

```
TBWC_object = TBWC(X_train,Y_train,'ntree','40','ncluster','100');
```

ตารางที่ ข.2 พารามิเตอร์ของคำสั่งใช้งานรูปแบบใช้งานทั่วไป

ชื่อพารามิเตอร์	ความหมาย	ตัวอย่าง
X_train	ตัวแปรเก็บค่าคุณลักษณะของชุดข้อมูลเรียนรู้	[0,0.2,0.4; 1,0.6,0.2; 0,0.1,0.3]
Y_train	ตัวแปรเก็บค่ากลุ่มของชุดข้อมูลเรียนรู้	[1;2;1]
'ntree','40'	จำนวนโมเดลในการสร้างโมเดลจำแนกประเภทด้วยขั้นตอนวิธี Bagging ด้วยต้นไม้ตัดสินใจ C4.5	40
'ncluster','100'	จำนวนคลัสเตอร์ในการจัดกลุ่มข้อมูลด้วยขั้นตอนวิธีการจัดกลุ่ม k-means	100
TBWC_object	ตัวแปรเก็บตัวจำแนกประเภท TBWC	

การใช้งานคำสั่งในการทำนายข้อมูล สามารถเรียกได้โดยใช้คำสั่งต่อไปนี้

```
Y_compute = TBWC_object.predict(X_test)
```

ตารางที่ ข.3 พารามิเตอร์ของคำสั่งใช้งานรูปแบบใช้งานทั่วไปในการทำนายข้อมูล

ชื่อพารามิเตอร์	ความหมาย	ตัวอย่าง
X_test	ตัวแปรเก็บค่าคุณลักษณะของชุดข้อมูลทดสอบ	[0,0.2,0.4; 1,0.6,0.2]
Y_compute	ตัวแปรเก็บค่ากลุ่มของชุดข้อมูลทดสอบที่ได้จากการทำนายข้อมูล	[1;2]

ผลลัพธ์ที่ได้จะคืนค่ากลุ่มของข้อมูลแต่ละตัว (Y_compute)

รหัสต้นฉบับภาษา MATLAB รูปแบบใช้งานทั่วไป แสดงได้ดังรูปที่ ข.6

```
% TBWC.m
classdef TBWC
   properties
        centroids = [];
        wa = [];
        cleanattr = 0;
        clean = [];
        centroids_class = [];
        ntrees = 0;
        minleaf = 0;
        cat = 0;
        ncluster = 0;
        distance = '';
        replicate = 0;
    end
     methods
        function a = TBWC(X_train, Y_train, varargin)
            a = learning(a,X_train,Y_train, varargin{:});
        end % TBWC constructor

        function Y_compute = predict(a, X_test)
            Y_compute = testing(a,X_test);
        end
     end % methods block
end

function Object = learning(Object, X_train, Y_train,
varargin)

pnames = {   'ntrees'  'minleaf' 'categories' 'ncluster'
'distance' 'replicates'};
dflts =  {   '10'     '1'          '0'          '5'
'cityblock'       '5'};
[eid,errmsg,ntrees,minleaf,cat,k,distance,replicate] ...
                    = getargs(pnames, dflts, varargin{:});
if ~isempty(eid)
    error(sprintf('TBWC:kmeans:%s',eid),errmsg);
end

ntrees = str2double(ntrees);
minleaf = str2double(minleaf);
cat = str2double(cat);
k = str2double(k);
replicate = str2double(replicate);

if(cat > 0)
    t = TreeBagger(ntrees,X_train,Y_train,'method','c',
'minleaf',minleaf,'cat',cat);
else
    t = TreeBagger(ntrees,X_train,Y_train,'method','c',
'minleaf',minleaf);
end
```

รูปที่ ข.6 (ก) รหัสต้นฉบับภาษา MATLAB รูปแบบใช้งานทั่วไป

```
numofattr = size(X_train,2);
wa = zeros(1,numofattr);
for i=1:ntrees
%    fprintf('Tree number: %d\n',i);
    wat = zeros(1,numofattr);
    subt = t.Trees{i};
    subtheight  = heightOf(subt);
    [~,attrcut] = cutvar(subt);
    for j=1:subt.numnodes
       usedattr = attrcut(j,1);
%       fprintf('    Attribute: %d \t',usedattr);
       if(usedattr > 0 && wat(usedattr) == 0)
            wat(usedattr) = (subtheight-
levelOf(subt,j)+1)/(subtheight+1);
%            fprintf('Level: %.4f \t',levelOf(subt,j));
%            fprintf('Weight: %.4f\n',wat(usedattr));
            wa(usedattr) =  wa(usedattr) + wat(usedattr);
%       else
%            fprintf('Leaf\n');
       end;
    end;
end;
wa = wa./ntrees;
wa = (wa - min(wa))/(max(wa)-min(wa));
%disp(wa);
%disp(size(wa));

cleanattr = 0;
clean = [];
for i=numofattr:-1:1
    if (wa(i) == 0)
%           disp(i);
           X_train(:,i) = [];
           cleanattr = cleanattr + 1;
           clean = [clean i];
    end;
end;
%disp(size(X_train));
%disp(size(Xt_train));
%disp(centroids);
% centroids is center of cluster
if(distance(1) == 's')
    distance = 'sqeuclidean';
else
    distance = 'cityblock';
end
[idx,centroids] = weight_kmeans(X_train,k,'distance',
distance,'emptyaction','singleton','replicates',replicate,
'weight',wa);
% centroids is center of cluster
centroids_class = zeros(k,1);
% centroids_class is class of centroid
```

รูปที่ ข.6 (ข) รหัสต้นฉบับภาษา MATLAB รูปแบบใช้งานทั่วไป

```
for i=1:k
    class_indices = (idx == i);
    class = Y_train(class_indices,:);
    centroids_class(i,1) = mode(class);
end
Object.centroids = centroids;
Object.wa = wa;
Object.cleanattr = cleanattr;
Object.clean = clean;
Object.centroids_class = centroids_class;
Object.ntrees = ntrees;
Object.minleaf = minleaf;
Object.cat = cat;
Object.ncluster = k;
Object.distance = distance;
Object.replicate = replicate;
end

function Y_compute = testing(Object, X_test)
    if(size(Object.clean) > 0)
        numofclean = size(Object.clean);
        for i=numofclean:-1:1
%               disp(Object.clean(i));
              X_test(:,Object.clean(i)) = [];
        end;
    end
    D = distfun(X_test, Object.centroids, Object.wa,
Object.distance);
    [~,Y_cluster] = min(D,[],2);
    Y_compute = Object.centroids_class(Y_cluster,1);
end

function level = levelOf(t, nodeind)
parentOfi = t.parent(nodeind);
if(parentOfi ~= 0),
   level = levelOf(t,parentOfi) + 1;
else
   level = 0;
end
end

function height = heightOf(t)
nodeindex = t.numnodes;
height = levelOf(t,nodeindex);
end

function D = distfun(X, C, W, dist)
%DISTFUN Calculate point to cluster centroid distances.
[n,p] = size(X);
D = zeros(n,size(C,1));
nclusts = size(C,1);
```

รูปที่ ข.6 (ค) รหัสต้นฉบับภาษา MATLAB รูปแบบใช้งานทั่วไป

```
147 switch dist
148 case 'sqeuclidean'
149     for i = 1:nclusts
150         D(:,i) = W(1,1).*((X(:,1) - C(i,1)).^2);
151         for j = 2:p
152             D(:,i) = D(:,i) + W(1,j).*((X(:,j) -
    C(i,j)).^2);
153         end
154         % D(:,i) = sum((X - C(repmat(i,n,1),:)).^2, 2);
155     end
156 case 'cityblock'
157     for i = 1:nclusts
158         D(:,i) = W(1,1).*(abs(X(:,1) - C(i,1)));
159
160         for j = 2:p
161             D(:,i) = D(:,i) + W(1,j).*(abs(X(:,j) -
    C(i,j)));
162         end
163         % disp(D);
164         % D(:,i) = sum(abs(X - C(repmat(i,n,1),:)), 2);
165     end
166 end
167 end
```

รูปที่ ข.6 (ง) รหัสต้นฉบับภาษา MATLAB รูปแบบใช้งานทั่วไป