เทคนิคการจำแนกประเภทข้อมูลส่วนน้อยบนข้อมูลไม่สมดุล ด้วยวิธีการแบ่งข้อมูล

นายกิตติพงศ์ ชมบุญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2558

CLASSIFICATION TECHNIQUE FOR MINORITY CLASS ON IMBALANCED DATASET WITH DATA PARTITIONING METHOD

Kittipong Chomboon

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy in Computer Engineering

Suranaree University of Technology

Academic Year 2015

เทคนิคการจำแนกประเภทข้อมูลส่วนน้อยบนข้อมูลไม่สมดุลด้วยวิธีการแบ่งข้อมูล

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

กรรมการสอบวิทยานิพนธ์

____
(รศ. ดร.นิตยา เกิดประสพ)
ประธานกรรมการ

____
(รศ. ดร.กิตติศักดิ์ เกิดประสพ)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

____
(ผศ. ดร.ปรเมศวร์ ห่อแก้ว)
กรรมการ

____
(ผศ. ดร.ศุภกฤษฏิ์ นิวัฒนากูล)
กรรมการ

____
(ผศ. ดร.สายสุนีย์ จับโจร)
กรรมการ

__________________________	__________________________
(ศ. ดร.ชูกิจ ลิมปิจำนงค์)	(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
รองอธิการบดีฝ่ายวิชาการและนวัตกรรม	คณบดีสำนักวิชาวิศวกรรมศาสตร์

กิตติพงศ์ ชมบุญ : เทคนิคการจำแนกประเภทข้อมูลส่วนน้อยบนข้อมูลไม่สมดุลด้วยวิธีการแบ่งข้อมูล (CLASSIFICATION TECHNIQUE FOR MINORITY CLASS ON IMBALANCED DATASET WITH DATA PARTITIONING METHOD)
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ, 98 หน้า

การจำแนกประเภทข้อมูลโดยใช้ข้อมูลที่ไม่สมดุลนั้นเป็นปัญหาสำคัญในการทำเหมืองข้อมูลที่น่าสนใจเนื่องจากการทำเหมืองข้อมูลด้วยข้อมูลที่ไม่สมดุลซึ่งมีข้อมูลคลาสส่วนใหญ่ และคลาสส่วนน้อยอยู่ปะปนกันนั้น ข้อมูลส่วนใหญ่จะมีคุณสมบัติบางประการที่บดบังคุณสมบัติของข้อมูลส่วนน้อย ทำให้การจำแนกประเภทข้อมูลส่วนน้อยนั้นไม่สามารถจำแนกได้อย่างมีประสิทธิภาพ ยกตัวอย่างเช่นข้อมูลไม่สมดุลของผลวินิจฉัยผู้ป่วยที่เป็นโรคมะเร็ง โดยมีข้อมูลส่วนใหญ่เป็นข้อมูลผู้ป่วยปกติ และมีข้อมูลส่วนน้อยที่เป็นโรคมะเร็ง ซึ่งเมื่อทำการจำแนกประเภทข้อมูลด้วยวิธีการจำแนกประเภทข้อมูลแบบปกติที่ให้ความสำคัญกับข้อมูลทุกคลาสเท่าเทียมกันนั้นจะทำให้ประสิทธิภาพในการจำแนกประเภทข้อมูลผู้ป่วยที่เป็นโรคมะเร็งซึ่งเป็นข้อมูลส่วนน้อยนั้นมีประสิทธิภาพไม่ดีเท่าที่ควร ดังนั้นในงานวิจัยนี้จึงได้นำเสนอเทคนิคการจำแนกประเภทข้อมูลที่มีขนาดของคลาสไม่สมดุลด้วยวิธีการแบ่งข้อมูล

ในงานวิจัยนี้เป็นการเสนอแนวคิดเพื่อแก้ปัญหาที่คุณสมบัติบางประการของข้อมูลส่วนใหญ่บดบังคุณสมบัติของข้อมูลส่วนน้อยจึงทำให้ประสิทธิภาพในการจำแนกข้อมูลส่วนน้อยนั้นไม่มีประสิทธิภาพดีเท่าที่ควร ด้วยวิธีการแบ่งข้อมูล โดยในงานวิจัยนี้จะแบ่งข้อมูลออกเป็น 2 ส่วน ได้แก่ ส่วนที่มีการซ้อนทับกัน และส่วนที่ไม่มีการซ้อนทับกัน โดยในแต่ละข้อมูลนั้นจะมีโมเดลในการจำแนกประเภทข้อมูล 2 โมเดล การทำนายคลาสของข้อมูลใหม่จะใช้ทั้งสองโมเดลประกอบกันเพื่อเพิ่มประสิทธิภาพในการจำแนกข้อมูลส่วนน้อยให้มีความถูกต้องสูงขึ้น ผลที่ได้จากการวิจัยคือการใช้วิธีการแบ่งข้อมูลด้วยการวัดระยะแบบ Euclidean และการใช้อัลกอริทึม SVM Linear kernel ให้ประสิทธิภาพในการจำแนกข้อมูลส่วนน้อยที่ดีที่สุด

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา________________________

ปีการศึกษา 2558 ลายมือชื่ออาจารย์ที่ปรึกษา____________________

KITTIPONG CHOMBOON : CLASSIFICATION TECHNIQUE FOR
MINORITY CLASS ON IMBALANCED DATASET WITH DATA
PARTITIONING METHOD. THESIS ADVISOR ASSOC. PROF.
KITTISAK KERDPRASOP, Ph.D., 98 PP.

DATA PARTITIONING / IMBALANCED DATA/ DATA MINING

Classification using imbalanced dataset is a challenging problem in the data mining research area. The difficulty is due to the fact that the number of data instances in the minority class is much less than the number of instances in the majority class. The majority data can over-shadow the minority data and make the classification performance of the minority class unacceptable. For instance, the imbalance between non-cancer patients and patients with breast cancer. The minority of breast cancer records in the majority group of non-cancer patients can absent when classifying with traditional techniques. This thesis, therefore, proposes a partitioning technique to handle the imbalanced dataset problem.

This research solves the imbalanced dataset problem by partitioning data into two groups: overlap and non-overlap data. Each partition has its own classification model. To predict the future event, both classifiers are used in order to improve the minority class prediction. The experimental results show that partitioning technique based on Euclidean distance measure when applied to the SVM with linear kernel yields the best performance in classifying minority data.

School of Computer Engineering Student's Signature________________

Academic Year 2015 Advisor's Signature________________

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่างๆที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัย ดังต่อไปนี้

รองศาสตราจารย์ ดร.นิตยา เกิดประสพ และรองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ให้คำปรึกษาในการทำงานวิจัย การจัดการรูปแบบ และช่วยตรวจทานความถูกต้องของวิทยานิพนธ์

ผู้ช่วยศาสตราจารย์ ดร.พิชโยทัย มหัทธนาภิวัฒน์ ผู้ช่วยศาสตราจารย์ ดร.คะชา ชาญศิลป์ ผู้ช่วยศาสตราจารย์ สมพันธ์ ชาญศิลป์ และ ผู้ช่วยศาสตราจารย์ ดร.ปรเมศวร์ ห่อแก้ว อาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้ความรู้พื้นฐานในสาขาต่าง ๆ ของวิศวกรรมคอมพิวเตอร์ และคุณกีระชาติ สุขสุทธิ์ ที่ช่วยตรวจวิทยานิพนธ์ฉบับร่าง

นอกจากนี้ขอขอบคุณครู อาจารย์ทั้งในอดีตและปัจจุบันที่ให้ความรู้แก่ผู้วิจัยจนประสบความสำเร็จในชีวิต

ท้ายที่สุด ขอกราบขอบพระคุณ บิดา มารดา ที่ให้กำเนิด อบรม เลี้ยงดูและส่งเสริมการศึกษาเป็นอย่างดี ทำให้ผู้วิจัยมีความพร้อมด้านความรู้ และมีกำลังใจในการทำวิจัย

กิตติพงศ์ ชมบุญ

สารบัญ

หน้า

บทที่

สารบัญ (ต่อ)

หน้า

ภาคผนวก

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

ปัจจุบันได้มีการค้นคว้า วิจัย และพัฒนาเทคโนโลยีคอมพิวเตอร์ให้ก้าวหน้าขึ้นเป็นอย่างมาก ทำให้เทคโนโลยีคอมพิวเตอร์มีประสิทธิภาพสูง และทำให้การจัดเก็บข้อมูลสามารถทำได้อย่างมีประสิทธิภาพ ส่งผลให้ในปัจจุบันได้มีการเก็บข้อมูลต่าง ๆ ไว้ในรูปแบบอิเล็กทรอนิกส์เป็นจำนวนมากทำให้การวิเคราะห์ข้อมูลแบบเก่า ซึ่งใช้เพียงมนุษย์เป็นผู้วิเคราะห์ข้อมูลทั้งหมดนั้นสามารถทำได้ยาก การทำเหมืองข้อมูล (Data Mining) จึงได้ถูกพัฒนาขึ้นโดยใช้เทคโนโลยีคอมพิวเตอร์และปัญญาประดิษฐ์เป็นตัวช่วยในการวิเคราะห์ข้อมูล ทำให้การวิเคราะห์ข้อมูลจำนวนมาก ๆ สามารถทำได้สะดวก รวดเร็ว และมีความแม่นยำในการวิเคราะห์ข้อมูลที่เพิ่มขึ้น โดยการทำเหมืองข้อมูลนั้นเป็นการวิเคราะห์ข้อมูลโดยใช้หลักการทางสถิติ จึงทำให้ในปัจจุบันการทำเหมืองข้อมูลเป็นงานที่สำคัญและน่าสนใจ

การทำเหมืองข้อมูลที่นิยมใช้ในงานด้านอุตสาหกรรมและวิทยาศาสตร์ คือการจำแนกประเภทข้อมูล (Data Classification) โดยจะทำการวิเคราะห์ข้อมูลที่มีอยู่แล้วเพื่อให้ได้แบบรูปของข้อมูล (Pattern) โมเดลจำแนกประเภทข้อมูล (Classification Model) หรือกฎการจำแนกประเภทข้อมูล (Classification Rules) เพื่อใช้ในการพยากรณ์ประเภทของข้อมูลในอนาคต หรือข้อมูลที่ยังไม่ได้จำแนกประเภท สำหรับเทคนิคที่นิยมใช้ในการจำแนกประเภทข้อมูลนั้นมีหลายเทคนิค เช่น เทคนิคนาอีฟเบย์ (Naïve Bayes) ซึ่งเทคนิคนี้เป็นการจำแนกข้อมูลโดยใช้หลักการพื้นฐานทางด้านสถิติในการจำแนกประเภทข้อมูล หรือการใช้เทคนิคต้นไม้ตัดสินใจ (Decision Tree) ซึ่งเป็นการจำลองรูปแบบของต้นไม้แต่มีรูปแบบกลับหัวกันคือ จะมีรากของต้นไม้ (Root Node) ขึ้นมาก่อนซึ่งจะหมายถึง แอตทริบิวต์ของข้อมูล และแยกโหนดออกไปเป็นลำดับชั้น ลักษณะคล้ายต้นไม้กลับหัว โดยที่โหนดใบ (Leaf Node) ซึ่งเป็นโหนดระดับล่างสุดนั้นจะเป็นโหนดที่ตัดสินชนิดของข้อมูลที่มีลักณะตามเส้นทางจากโหนดรากจนถึงโหนดใบนั้น ๆ

ในการตัดสินใจเลือกเทคนิคการจำแนกประเภทข้อมูลชนิดต่าง ๆ นั้นส่วนมากจะดูจากประสิทธิภาพและความแม่นยำในการจำแนกประเภทข้อมูล (Accuracy) เทคนิคการจำแนกประเภทข้อมูลแต่ละเทคนิคไม่สามารถใช้ได้ดีกับทุกชุดข้อมูล เช่นการใช้เทคนิคต้นไม้ตัดสินใจกับชุดข้อมูล

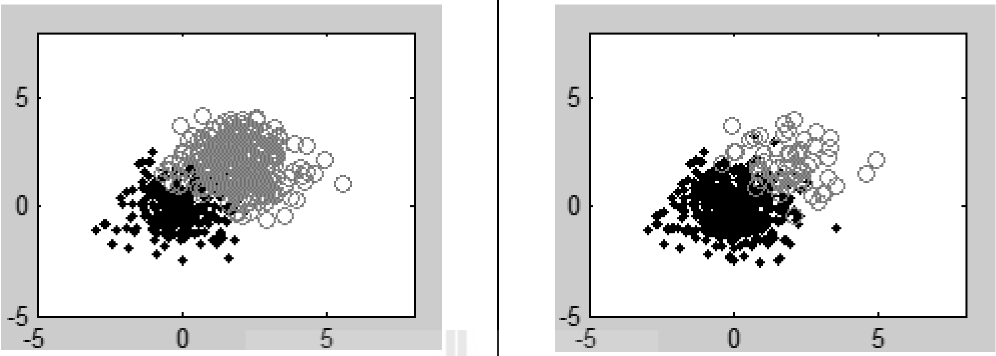
ที่สมดุล (Balanced Data) ก็จะให้ผลที่ดีกว่าใช้กับชุดข้อมูลที่ไม่สมดุล (Imbalanced Data) ซึ่งข้อมูลที่สมดุลนั้นจะมีลักษณะมีจำนวนข้อมูลในแต่ละคลาสที่ใกล้เคียงกัน แต่ข้อมูลที่ไม่สมดุลนั้นจะเป็นชุดข้อมูลที่มีจำนวนข้อมูลในแต่ละคลาสแตกต่างกัน ยกตัวอย่างเช่น ชุดข้อมูลการผลิตสินค้า ที่เป็นการเก็บข้อมูลกระบวนการต่าง ๆ ในการผลิตสินค้าซึ่งมีคลาสคือ สินค้าที่ใช้งานได้และสินค้าที่ใช้งานไม่ได้ จะเห็นได้ว่าส่วนมากคลาสของสินค้าที่ใช้งานได้จะมีมากกว่าคลาสของสินค้าที่ใช้งานไม่ได้ หรือจะเป็นข้อมูลในด้านการแพทย์จะเห็นได้ว่าผู้ป่วยที่เป็นโรคร้ายแรงนั้นมีจำนวนน้อยกว่าผู้ป่วยปกติอย่างมาก ซึ่งในการจำแนกประเภทข้อมูลไม่สมดุลนั้นเป็นเรื่องที่น่าสนใจเนื่องจากในบางกรณีของข้อมูลส่วนน้อยเป็นข้อมูลที่สำคัญ ทำให้ผู้วิเคราะห์ต้องการเพิ่มความสามารถในการจำแนกกลุ่มข้อมูลที่มีอยู่น้อยให้มีประสิทธิภาพสูง

ลักษณะของข้อมูลที่ไม่สมดุลนั้นจำนวนข้อมูลในแต่ละคลาสเป้าหมายจะมีจำนวนที่แตกต่างกันมาก (Chawla et al., 2002; Chawla et al., 2004; He and Garcia, 2009; Jo and Japkowicz, 2004; Wang and Japkowicz, 2010) เช่น ชุดข้อมูลหนึ่งมีจำนวนข้อมูลทั้งหมด 100 ข้อมูล โดยมี 2 คลาสได้แก่คลาส A มีข้อมูล 85 ข้อมูล และคลาส B มีข้อมูล 15 ข้อมูล เราจะเรียกคลาส A ซึ่งมีขนาดใหญ่ว่า คลาสส่วนมาก (Majority Class) และข้อมูลในคลาส B ซึ่งเป็นคลาสที่เล็กกว่าจะเรียกว่า คลาสส่วนน้อย (Minority Class) และเมื่อเรานำข้อมูลที่มีลักษณะข้อมูลที่ไม่สมดุลไปทำการจำแนกประเภทข้อมูลด้วยอัลกอริทึมพื้นฐานแล้ว จะทำให้ประสิทธิภาพในการจำแนกคลาสส่วนน้อยมีประสิทธิภาพต่ำ ยกตัวอย่างดังรูปที่ 1.1 ที่ใช้อัลกอริทึม k-NN โดยกำหนด k เท่ากับ 5 จะเห็นได้ว่าข้อมูลที่สมดุลนั้นจะให้ค่าความถูกต้องในการจำแนกของทั้งสองคลาสมีค่าใกล้เคียงกัน (รูปที่ 1.1(ก)) ส่วนข้อมูลที่ไม่สมดุลนั้น (รูปที่ 1.1(ข)) ข้อมูลในคลาสส่วนมาก จะมีอิทธิพลในการจำแนกสูงกว่าข้อมูลในคลาสส่วนน้อย เนื่องจากปริมาณข้อมูลที่ต่ำกว่ามากทำให้ค่าความถูกต้องในการจำแนกของคลาสส่วนน้อยลดลงจาก 95% ในรูปที่ 1.1(ก) เหลือเพียง 35% ดังในรูปที่ 1.1(ข)

จากปัญหาของการจำแนกคลาสส่วนน้อยที่ให้ค่าความแม่นหรือความถูกต้องต่ำในข้อมูลที่ไม่สมดุลนั้นทำให้มีนักวิจัยจำนวนมากทำการค้นคว้าและเสนอแนวคิดต่าง ๆ ที่จะแก้ไขปัญหาการจำแนกข้อมูลที่ไม่สมดุล โดยให้ความสำคัญกับประสิทธิภาพในการจำแนกคลาสส่วนน้อย ให้มีประสิทธิภาพที่ดีขึ้น ซึ่งงานวิจัยส่วนใหญ่ได้เสนอแนวคิดบนพื้นฐานของการใช้เทคนิคการสุ่มซ้ำ (Resampling) เช่นในงานวิจัยของ Estabrooks and Japkowicz (2001) ได้ใช้เทคนิคการสุ่มเพิ่มและการสุ่มลดร่วมกัน หลังจากนั้นจึงใช้อัลกอริทึมต่าง ๆ ในการจำแนกเพื่อทดสอบประสิทธิภาพในการจำแนกข้อมูล แต่ก็ไม่สามารถสรุปได้ว่าการสุ่มเพิ่มหรือการสุ่มลดวิธีไหนจะให้ประสิทธิภาพในการจำแนกที่ดีกว่ากัน นอกจากนั้นแล้วในงานวิจัยของ Alhammady and Ramamohanarao (2004) นำเสนอวิธีการที่ใช้โครงสร้างต้นไม้ตัดสินใจในการจำแนกประเภทข้อมูลที่ไม่สมดุลโดยได้

	A	B	<- Predicted
A	224	24	Acc(90%)
B	13	239	Acc(95%)

	A	B	<- Predicted
A	432	6	Acc(99%)
B	40	22	Acc(35%)



ก) ข้อมูลสมดุล (คลาส A มี 248 ข้อมูล คลาส B มี 252 ข้อมูล)

ข) ข้อมูลไม่สมดุล (คลาส A มี 438 ข้อมูล คลาส B มี 62 ข้อมูล)

รูปที่ 1.1 เปรียบเทียบประสิทธิภาพในการจำแนกข้อมูลสมดุลและข้อมูลไม่สมดุล

นำเสนออัลกอริทึมใหม่ชื่อว่า Emerging Pattern and Decision Tree (EPDT) ซึ่งเป็นเทคนิคในการเลือกคุณสมบัติของข้อมูลโดยวิเคราะห์จาก Growth Rates ในงานวิจัยของ Kotsiantis et al. (2006) ได้รวบรวมเทคนิคและวิธีการต่าง ๆ ในการจัดการเกี่ยวกับข้อมูลที่ไม่สมดุลได้แก่เทคนิคของ Fan et al. (1999) ซึ่งได้เสนอการถ่วงค่าน้ำหนักในการทำนายผิด ซึ่งเป็นการกำหนดค่าน้ำหนักในการทำนายข้อมูลส่วนน้อยให้มีความสำคัญมากขึ้นกับอัลกอริทึม Adaboost ส่วนงานวัยจัยของ Japkowicz and Stephen (2002) ได้เสนอเทคนิคการเรียนรู้แบบมีค่าใช้จ่าย (Cost-Sensitive) ซึ่งได้เปรียบเทียบกับเทคนิคการสุ่มเพิ่มหรือการสุ่มลดโดยงานวิจัยนี้สามารถสรุปได้ว่าเทคนิคการเรียนรู้แบบมีค่าใช้จ่ายมีประสิทธิภาพในการจำแนกที่ดีกว่าการสุ่มเพิ่มหรือการสุ่มลด และในงานวิจัยของ Weiss and Provost (2003) เสนอวิธีการจัดการกับข้อมูลที่ไม่สมดุลด้วยการคัดเลือกข้อมูลฝึกสอนที่เหมาะสมโดยใช้อัลกอริทึม Progressive-Sampling ซึ่งเป็นอัลกอริทึมที่อาศัยพื้นฐานของค่าความสันพันธ์ของข้อมูลเป็นตัวช่วยในการเลือกชุดข้อมูลฝึกสอน ส่วนงานวิจัยของ Kotsiantis and Pintelas (2003) ได้เสนอวิธีการเพิ่มประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลด้วยวิธีการโหวตเพื่อทำนายคลาสของข้อมูลใหม่โดยใช้อัลกอริทึม 3 อัลกอริทึมได้แก่ Naïve Bayes, C4.5, 5NN และในงานวิจัยของ Barandela et al. (2003) ได้เสนอเทคนิคการกำหนดค่าน้ำหนักของระยะทางของอัลกอริทึม k-NN เพื่อใช้ในการปรับค่าระยะทางที่เหมาะสมสำหรับอัลกอริทึม k-NN เพื่อให้มีประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลที่มีประสิทธิภาพ งานวิจัยของ Estabrooks et al. (2004) ได้เสนอเทคนิคที่ใช้อัลกอริทึม AdaBoost ในการจำแนกข้อมูลที่ไม่สมดุลที่เป็นข้อมูลข้อความ

(Text Classification) โดยทำการวิเคราะห์จากมาตรวัด Precision และ Recall เพื่อใช้ในการบอกประสิทธิภาพในการจำแนกข้อมูล งานวิจัยของ Zheng et al. (2004) ได้เสนอการคัดเลือกฟีเจอร์ (Feature Selection) ซึ่งเป็นการคัดเลือกฟีเจอร์ที่มีการกระจายของข้อมูลคลาส Positive และ Negative อย่างชัดเจนเพื่อจำแนกข้อมูลที่ไม่สมดุล ในงานวิจัยของ Wu et al. (2007) ได้เสนอวิธีการสร้างข้อมูลเป็นกลุ่มย่อย ๆ ด้วยวิธีการจัดกลุ่มข้อมูลเพื่อทำให้ข้อมูลเกิดความสมดุลด้วยเทคนิค Classification using local clustering (COG) ซึ่งได้ใช้อัลกอริทึม SVM ในการจำแนกข้อมูล ต่อมางานวิจัยของ He and Ghodsi (2010) ได้ใช้เทคนิคการสุ่มลด (Undersampling) เพื่อลดจำนวนคลาสส่วนมาก ร่วมกับการใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) เพื่อจำแนกคลาสส่วนน้อย และในงานวิจัยของ Seiffert et al. (2010) ได้เสนอวิธีการปรับปรุงข้อมูลด้วยเทคนิค RUSBoost ซึ่งเป็นการปรับปรุงข้อมูลด้วยวิธีการสุ่มลดและได้นำไปใช้กับอัลกอริทึม C4.5, Naïve Bayes, RIPPER โดยผลที่มีความแม่นยำในการจำแนกมากกว่า SMOTEBoost ต่อมาในงานวิจัยของ Yang et al., (2012) ได้เสนอวิธีการจัดการกับข้อมูลที่ไม่สมดุลด้วยการคัดเลือกคุณสมบัติของข้อมูลซึ่งได้เสนอเทคนิค Uncorrelated Discriminant Analysis (UDA) ซึ่งจะเป็นการคัดเลือกคุณสมบัติที่มีค่าความสัมพันธ์กันน้อยและเมื่อเลือกคุณสมบัติที่มีความสัมพันธ์กันน้อยแล้วจะทำให้ระยะห่างของจุดกึ่งกลางระหว่างคลาสมากที่สุด งานวิจัยของ Antonelli et al. (2014) นั้นได้ใช้วิธีการทาง Fuzzy ในการจำแนกประเภทข้อมูลกับข้อมูลที่ไม่สมดุลโดยที่เสนอให้ใช้ Fuzzy rule-based classifiers (FRBC) เป็นอัลกอริทึมในการจำแนกประเภทข้อมูลที่ไม่สมดุล งานวิจัยของ Li et al. (2014) ได้ทำการจำแนกประเภทข้อมูลที่ไม่สมดุลด้วยวิธีการถ่วงน้ำหนักด้วยเทคนิค Extreme learning machine (ELM) แล้วทำการ Boosting และได้ทำการเปรียบเทียบกับวิธี Adaboost ผลปรากฏว่าวิธีที่นำเสนอให้ผลลัพธ์ที่ดีกว่า และงานวิจัยของ Cateni et al. (2014) ได้ใช้เทคนิคการสุ่มเพิ่ม (Oversampling) เพื่อเพิ่มคลาสส่วนน้อยให้มีจำนวนใกล้เคียงกับคลาสส่วนมาก และใช้วิธีการสุ่มลดในการลดจำนวนคลาสส่วนมากให้มีจำนวนใกล้เคียงกับจำนวนในคลาสส่วนน้อย โดยใช้ทั้งสองเทคนิคร่วมกันในการพัฒนาเทคนิคการจำแนก งานวิจัยของ Vluymans et al. (2015) ได้นำวิธีการของ Fuzzy มาช่วยในการจำแนกประเภทข้อมูลไม่สมดุลโดยการใช้ Fuzzy มาช่วยในการถ่วงน้ำหนักในแต่ละคลาสเพื่อเพิ่มความแม่นยำในการจำแนกประเภทข้อมูล งานวิจัยของ Piyanoot et al. (2015) ได้ใช้เทคนิคการแบ่งข้อมูลที่ซ้อนทับกัน และใช้อัลกอริทึม RBF, RBFN, DBSCAN ในการจำแนกประเภทข้อมูล

ดังนั้นในงานวิจัยนี้ผู้วิจัยจึงได้เสนอเทคนิคในการเพิ่มประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลโดยการแบ่งข้อมูลที่มีการซ้อนทับกันระหว่างคลาสด้วยเทคนิคใหม่โดยการใช้การวัดระยะห่าง Euclidean Distance และทดสอบเปรียบเทียบประสิทธิภาพในการจำแนกข้อมูลด้วยค่าความแม่นในการจำแนก (Classification Accuracy) และการวัดประสิทธิภาพในการจำแนกข้อมูลแบบอื่น ๆ ได้แก่ TP Rate, G-Means, F-Measure, Accuracy, TN Rate

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาและพัฒนาเทคนิคการจำแนกข้อมูลไม่สมดุลให้มีความแม่นยำเพิ่มขึ้น
2. เพื่อเปรียบเทียบประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลระหว่างเทคนิคที่พัฒนาขึ้นกับเทคนิคที่ใช้อยู่ในปัจจุบัน
3. เพื่อเพิ่มประสิทธิภาพในการจำแนกข้อมูลในคลาสที่สนใจให้สามารถจำแนกได้ดี

1.3 ขอบเขตของการวิจัย

1. ข้อมูลที่ใช้จำเป็นต้องเป็นข้อมูลตัวเลขเท่านั้น (ยกเว้นแอตทริบิวต์คลาสที่เป็นค่าข้อความ หรือ Nominal)
2. ข้อมูลที่ใช้จำเป็นต้องมีคลาสเป้าหมาย 2 คลาสเท่านั้น
3. การเปรียบเทียบประสิทธิภาพจะใช้เกณฑ์ความถูกต้องหรือความแม่นของการจำแนกข้อมูล
4. ข้อมูลที่ใช้ในงานวิจัยนี้เป็นข้อมูลจริงที่นำมาจาก UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) และ Knowledge Extraction based on Evolutionary Learning (http://www.keel.es/)

1.4 ประโยชน์ที่จะได้รับ

จากการศึกษาพัฒนาเทคนิคสำหรับการจำแนกข้อมูลที่มีคลาสเป้าหมาย 2 คลาสแต่ปริมาณข้อมูลใน 2 คลาสแตกต่างกันมากด้วยวิธีการค้นหาข้อมูลที่มีการซ้อนทับกันนั้นคาดหวังว่าจะเกิดประโยชน์คือเพิ่มประสิทธิภาพในการจำแนกประเภทของคลาสส่วนน้อยให้มีความแม่นยำในการจำแนกเพิ่มขึ้น

บทที่ 2
ปริทัศน์วรรณกรรม

ในส่วนของปริทัศน์วรรณกรรมนี้ประกอบด้วยการทบทวนวรรณกรรมและงานวิจัยที่เกี่ยวข้อง ซึ่งมีรายละเอียดเกี่ยวกับ การจำแนกประเภทข้อมูล วิธีการจำแนกประเภทด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน การวัดระยะห่างระหว่างข้อมูลเพื่อตรวจสอบการซ้อนทับกันของข้อมูลต่างคลาส เกณฑ์ที่ใช้ในการวัดประสิทธิภาพการจำแนก และงานวิจัยที่เกี่ยวข้อง

2.1 การจำแนกประเภทข้อมูล

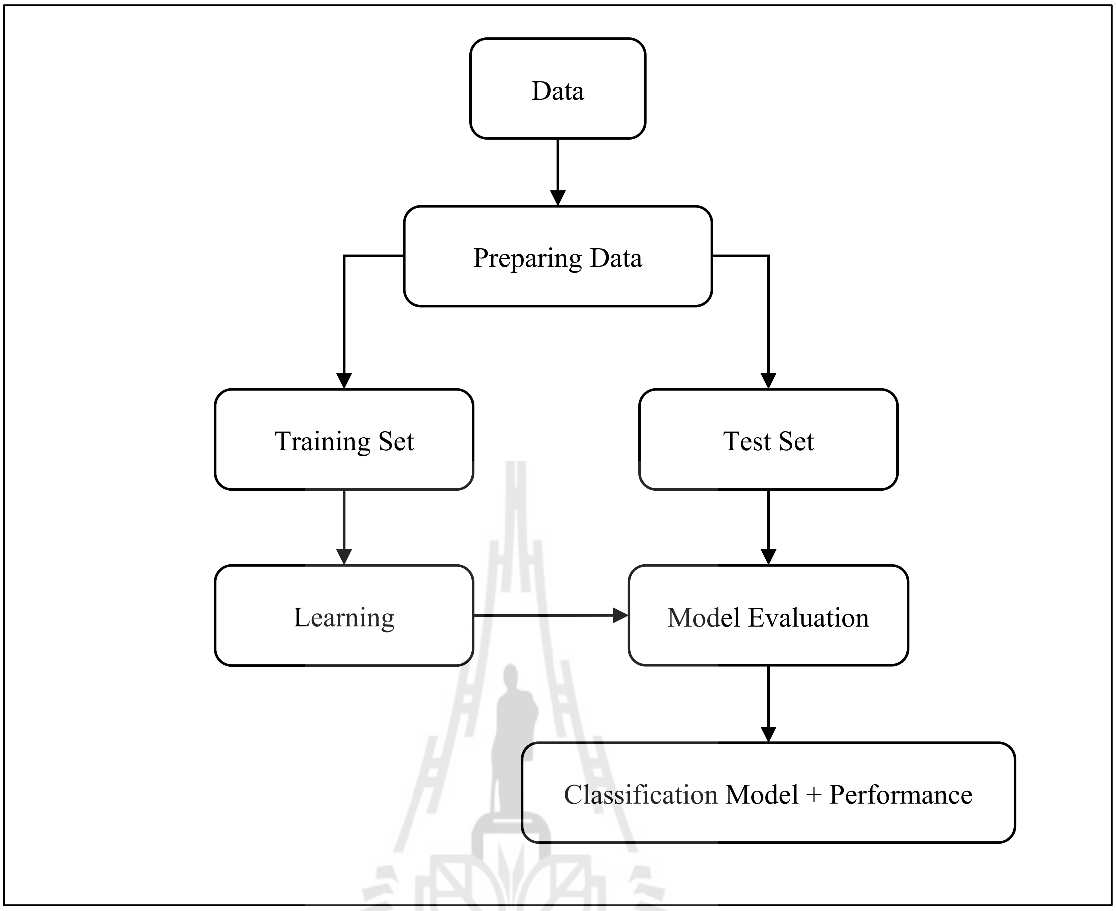
การจำแนกประเภทข้อมูล (Chawla, 2005; Manevitz and Yousef, 2007; Pratik and Upasna, 2013) เป็นการทำเหมืองข้อมูลประเภทหนึ่งซึ่งนิยมใช้ในงานด้านอุตสาหกรรมและวิทยาศาสตร์ ซึ่งการจำแนกประเภทข้อมูลคือกระบวนการสร้างโมเดลจำแนกประเภทข้อมูลจากข้อมูลที่มีอยู่ เพื่อใช้ในการจำแนกประเภทของข้อมูลในอนาคต ยกตัวอย่างเช่น การใช้ข้อมูลการวินิจฉัยโรคมะเร็งเต้านม เพื่อสร้างโมเดลจำแนกประเภทผู้ที่เข้าตรวจโรคมะเร็งเต้านมว่าเป็นมะเร็งหรือไม่เป็นมะเร็ง ซึ่งในการสร้างโมเดลจำแนกประเภทข้อมูลจำเป็นต้องมีเป้าหมายในการสร้างซึ่งเรียกว่า “คลาส” (Class) จากตัวอย่างที่กล่าวมาข้างต้นคลาสของข้อมูลการวินิจฉัยโรคมะเร็งเต้านมนั้นมี 2 คลาส คือ “เป็นโรคมะเร็งเต้านม” และ “ไม่เป็นโรคมะเร็งเต้านม” ปัจจัยสำคัญที่สุดของการสร้างโมเดลจำแนกประเภทข้อมูลนั้นคือ ข้อมูลที่ใช้ในการสร้างโมเดลจำแนกประเภทข้อมูล ซึ่งข้อมูลที่นำมาใช้ต้องมีปริมาณที่มากพอสำหรับการสร้างโมเดลจำแนกประเภทข้อมูล กล่าวคือข้อมูลที่นำมาใช้ในการจำแนกจะต้องมีคลาสที่แน่นอนและข้อมูลในแต่ละคลาสมีจำนวนไม่น้อยเกินไป ยกตัวอย่างจากข้อมูลในตารางที่ 2.1 ซึ่งเป็นข้อมูลของดอกไอริสชนิดต่าง ๆ โดยในข้อมูลชุดข้อมูลประกอบด้วยข้อมูลทั้งหมด 16 ข้อมูล (https://archive.ics.uci.edu/ml/datasets.html) มีจำนวนลักษณะประจำหรือแอตทริบิวต์ (Attribute) ทั้งหมด 5 แอตทริบิวต์ได้แก่ Sepal.Length, Sepal.Width, Petal.Length, Petal.Width และ Species โดยมีแอตทริบิวต์ Species เป็นคลาสเป้าหมายซึ่งมีทั้งหมด 2 คลาสได้แก่ setosa และ versicolor โดยข้อมูลดอกไอริสชนิดต่าง ๆ นั้นเป็นข้อมูลที่สมบรูณ์เนื่องจากเป็นข้อมูลที่ไม่มีข้อมูลในแอตทริบิวต์ใดเลยที่ขาดหายไป (Missing Value) หรือไม่มีแอตทริบิวต์ใดเลยที่ไม่ทราบค่า (Null Available)

ตารางที่ 2.1 แสดงข้อมูลดอกไอริสสายพันธ์ setosa และ versicolor

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
7	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4	1.3	versicolor
6.5	2.8	4.6	1.5	versicolor
5.7	2.8	4.5	1.3	versicolor
6.3	3.3	4.7	1.6	versicolor
4.9	2.4	3.3	1	versicolor

การจำแนกประเภทข้อมูลเกิดขึ้นจากการหาความสัมพันธ์จากข้อมูลในชุดข้อมูลขนาดใหญ่ ซึ่งขั้นตอนและกระบวนการต่าง ๆ ของการสร้างโมเดลจำแนกประเภทข้อมูลแสดงในรูปที่ 2.1 โดยสามารถอธิบายขั้นตอนและกระบวนการต่าง ๆ ได้ดังต่อไปนี้

ขั้นตอนแรกคือการเตรียมข้อมูล (Preparing Data) ในขั้นตอนนี้เป็นการเตรียมข้อมูลให้พร้อมใช้งานสำหรับการจำแนกประเภทพข้อมูลเนื่องจาก ข้อมูลดิบที่ได้มาอาจมีข้อมูลเกินความจำเป็นหรือไม่เกี่ยวข้องในการจำแนกประเภท ตัวอย่างเช่น จากข้อมูลในตารางที่ 2.1 ถ้าหากมีแอตทริบิวต์วันที่เก็บข้อมูลเพิ่มเข้ามา ซึ่งข้อมูลวันที่เก็บข้อมูลไม่จำเป็นต่อการจำแนกประเภทดอกไอริส ถ้าหากนำข้อมูลวันที่เก็บข้อมูลมาใช้ในการจำแนกประเภทอาจจะทำให้ประสิทธิภาพในการจำแนกต่ำกว่าที่ควรจะเป็น และในขั้นตอนการเตรียมข้อมูลนี้ยังรวมถึงวิธีจัดการเกี่ยวกับข้อมูลที่ขาดหายไปด้วย ว่าจะใช้วิธีใดในการจัดการข้อมูลที่ขาดหายไป ก่อนที่จะนำข้อมูลที่สมบูรณ์ไปใช้ในการจำแนกประเภทข้อมูล



รูปที่ 2.1 แสดงกระบวนการจำแนกประเภทข้อมูล

ขั้นตอนที่สองคือการแบ่งข้อมูล (Partitioning Data) ในขั้นตอนการแบ่งข้อมูลนี้จะเป็นการแบ่งข้อมูลออกเป็น 2 ส่วนคือส่วนข้อมูลที่ใช้ในการฝึกสอน (Training Set) และข้อมูลสำหรับทดสอบประสิทธิภาพ (Test Set) ซึ่งในการแบ่งข้อมูลสำหรับฝึกสอนและข้อมูลสำหรับทดสอบประสิทธิภาพโดยทั่วไปแล้วจะทำการแบ่งข้อมูลให้ชุดข้อมูลที่ใช้ในการฝึกสอนมีปริมาณมากกว่าข้อมูลที่ใช้สำหรับทดสอบประสิทธิภาพ ข้อมูลที่ใช้ในการฝึกสอนเป็นข้อมูลที่ใช้เป็นอินพุตสำหรับอัลกอริทึมในการเรียนรู้เพื่อสร้างโมเดลจำแนกประเภทข้อมูล ส่วนข้อมูลสำหรับทดสอบประสิทธิภาพคือชุดข้อมูลที่เตรียมไว้เพื่อทดสอบประสิทธิภาพของโมเดลจำแนกประเภทข้อมูลที่เรียนรู้มาจากชุดข้อมูลฝึกสอน จากตารางที่ 2.1 สามารถแบ่งข้อมูลฝึกสอนและข้อมูลทดสอบแสดงได้ดังตารางที่ 2.2

ตารางที่ 2.2 Train Data, Test Data ของข้อมูลไอริส

Partition	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Train Data	5.1	3.5	1.4	0.2	setosa
	4.9	3	1.4	0.2	setosa
	4.7	3.2	1.3	0.2	setosa
	4.6	3.1	1.5	0.2	setosa
	5	3.6	1.4	0.2	setosa
	7	3.2	4.7	1.4	versicolor
	6.4	3.2	4.5	1.5	versicolor
	6.9	3.1	4.9	1.5	versicolor
	5.5	2.3	4	1.3	versicolor
	6.5	2.8	4.6	1.5	versicolor
Test Data	5.4	3.9	1.7	0.4	setosa
	4.6	3.4	1.4	0.3	setosa
	5	3.4	1.5	0.2	setosa
	5.7	2.8	4.5	1.3	versicolor
	6.3	3.3	4.7	1.6	versicolor
	4.9	2.4	3.3	1	versicolor

ขั้นตอนที่สามคือการเรียนรู้ (Learning) จากข้อมูลฝึกสอนเพื่อสร้างโมเดลจำแนกประเภทข้อมูล ในขั้นตอนนี้จะเป็นการเรียนรู้แบบรูปและสร้างโมเดลจำแนกประเภทข้อมูลด้วยอัลกอริทึมต่าง ๆ ยกตัวอย่างเช่น Decision Tree, Support Vector Machine, Naïve Bayes เป็นต้น ซึ่งอัลกอริทึมเหล่านี้จะทำหน้าที่สร้างโมเดลจำแนกประเภทข้อมูลด้วยวิธีการที่แตกต่างกันเพื่อจำแนกประเภทของข้อมูลให้ได้แม่นยำที่สุด

ขั้นตอนที่สี่คือการตรวจสอบประสิทธิภาพของโมเดล (Model Evaluation) ในขั้นตอนนี้จะเป็นการทดสอบประสิทธิภาพของโมเดลจำแนกประเภทข้อมูลที่ได้จากชุดข้อมูลฝึกสอนและอัลกอริทึมต่าง ๆ เพื่อสรุปประสิทธิภาพในการจำแนกกับชุดข้อมูลทดสอบและช่วยในการตัดสินใจเลือกใช้โมเดลจำแนกประเภทข้อมูลสำหรับข้อมูลในอนาคต จากข้อมูลฝึกสอนและข้อมูลทดสอบในตารางที่ 2.2 เมื่อทำการเรียนรู้ชุดด้วยข้อมูลฝึกสอนและใช้ชุดข้อมูลทดสอบในการทดสอบประสิทธิภาพจะได้ประสิทธิภาพในการจำแนกดังรูปที่ 2.2 จากผลการทดสอบประสิทธิภาพของ

การจำแนกประเภทข้อมูลดังแสดงในรูปที่ 2.3 จะเห็นได้ว่าอัลกอริทึม Support Vector Machine ให้ประสิทธิภาพการจำแนกที่มีความแม่นสูง ในงานวิจัยนี้จึงเลือกใช้ Support Vector Machine เป็นอัลกอริทึมสำหรับจำแนกข้อมูลเนื่องจากมีการสร้างไฮเปอร์เพลนขึ้นมาจึงทำให้มีความยืดหยุ่นในการจำแนกข้อมูล และเมื่อนำไปใช้กับข้อมูลจริงน่าจะให้ผลลัพธ์ที่ดีกว่าอัลกอริทึม Naïve Bayes

Decision Tree

Predict Actual	setosa	versicolor
setosa	3	0
versicolor	3	0
Accuracy	(3/6) = 50%	

Naïve Bayes

Predict Actual	setosa	versicolor
setosa	3	0
versicolor	0	3
Accuracy	(6/6) = 100%	

Support Vector Machine

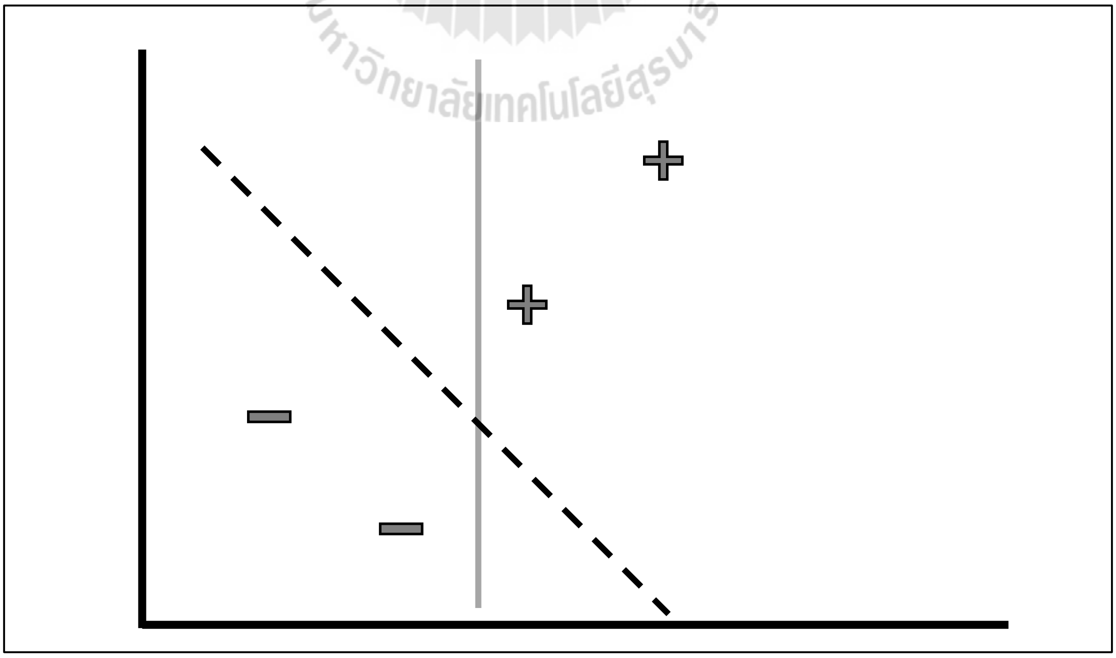
Predict Actual	setosa	versicolor
setosa	3	0
versicolor	0	3
Accuracy	(6/6) = 100%	

รูปที่ 2.2 แสดงผลการทดสอบโมเดล DT, NB, SVM

2.2 ซัพพอร์ตเวกเตอร์แมชชีน

ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) เป็นอัลกอริทึมสำหรับจำแนกประเภทข้อมูลชนิดที่พัฒนาจากอัลกอริทึมเพอร์เซพตรอน (Perceptron) ซึ่งเป็นการใช้เส้นตรงในการจำแนกข้อมูลสองประเภทออกจากกันโดยลากเส้นตรงเพื่อพยายามแบ่งข้อมูลทั้งสองออกจากกันให้ดีที่สุด หลังจากนั้นอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนได้มีการพัฒนาโดยมีแนวคิดหลักคือจะใช้เส้นตรงในการแบ่งข้อมูลอย่างไรให้มีประสิทธิภาพกับข้อมูลที่เข้ามาใหม่ ซึ่งนั่นก็คือการสร้างไฮเปอร์เพลน (Hyperplane) ให้อยู่ตรงกลางระหว่างทั้งสองข้อมูลมากที่สุด ซึ่งไฮเปอร์เพลนนี้จะมีขอบทั้งสองหรือเรียกว่ามาร์จิน (Margin) ที่มีขนาดเท่ากันและมีความกว้างมากที่สุดเท่าที่เป็นไปได้ หลังจากนั้นได้พัฒนาที่จะไม่ใช้เส้นตรงเพียงอย่างเดียวในการแบ่งข้อมูลเนื่องจากอาจจะไม่เหมาะสมกับข้อมูลชนิดต่าง ๆ จึงได้เกิดเป็นเคอร์เนล (Kernel) ต่าง ๆ ที่ใช้ในการแบ่งข้อมูล

ซัพพอร์ตเวกเตอร์แมชชีน (Cortes and Vapnik, 1995; Farquad and Bose, 2012; Guyon et al, 2002; He and Ghodsi, 2010; Tomar and Agarwal, 2015) อย่างง่ายจะเป็นการใช้เส้นตรงในการแบ่งข้อมูลแสดงในรูปที่ 2.3 จะเป็นการแบ่งข้อมูลทั้งสองกลุ่มคือกลุ่มที่เป็น + และกลุ่มที่เป็น – ออกจากกันโดยใช้สมการเส้นตรงจะเห็นได้ว่าในรูปนั้นจะมีเส้น 2 เส้นที่แบ่งกลุ่มข้อมูลทั้งสองกลุ่มออกจากกันก็คือเส้นทึบและเส้นประ โดยทั้งสองเส้นนั้นสามารถแบ่งข้อมูลออกเป็น 2 ประเภทที่ชัดเจนแต่ว่าซัพพอร์ตเวกเตอร์แมชชีนนั้นจะคำนึงถึงมาร์จินที่กว้างที่สุดที่เป็นไปได้นั่นก็คือเส้นประที่แบ่งข้อมูล



รูปที่ 2.3 แสดงตัวอย่างการแบ่งข้อมูลด้วยเส้นตรง

การทำงานของอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนจะเริ่มจากการสร้างไฮเปอร์เพลนโดยอาศัยหลักการของเส้นตรงก่อนคือ ax + by = c ซึ่งเมื่อเปลี่ยนเป็นไฮเปอร์เพลนแล้วจะได้ดังสมการที่ (2-1) โดยกำหนดให้ข้อมูลทั้งหมดคือ $D = \{(x_i, y_i)\}_{i=1}^{n}$ ซึ่งมีจำนวนข้อมูลทั้งหมด n จำนวน และมีคลาสของข้อมูลทั้งหมด 2 คลาสคือ $y_i \in \{+1, -1\}$ ซึ่งก็คือคลาส +1 และคลาส -1 โดยถ้าหากข้อมูลใด ๆ ที่อยู่บนระนาบเดียวกันกับไฮเปอร์เพลนจะมีค่าดังสมการที่ (2-2)

$$h(x) = w^T x + b \tag{2-1}$$

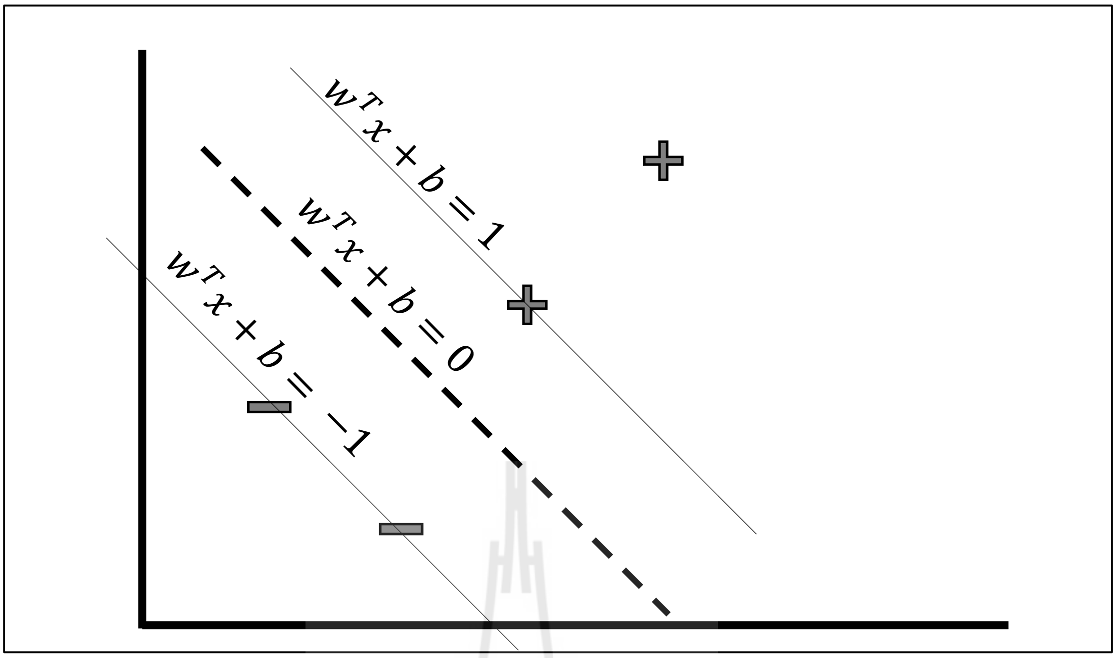
โดยที่ h(x) คือ ไฮเปอร์เพลน
w คือเวกเตอร์ถ่วงน้ำหนัก
x คือข้อมูลใน D
b คือค่าไบแอส (Bias)

$$h(x) = w^T x + b = 0 \tag{2-2}$$

โดยไฮเปอร์เพลน h(x) (Jayadeva et al, 2007) นั้นจะทำการแบ่งข้อมูลทั้งสองคลาสออกจากกัน โดยที่มี w เป็นเวกเตอร์ที่ตั้งฉากกับระนาบของไฮเปอร์เพลน ซึ่งเป็นตัวกำหนดทิศทางและความเอียงของไฮเปอร์เพลน ส่วน x นั้นเป็นข้อมูลในชุดข้อมูลมีลักษณะเป็นเวกเตอร์ และเมื่อ y_i มีค่าเป็น +1 จะทำให้ h(x) มีค่ามากกว่าหรือเท่ากับ 1 ส่วนถ้าหาก y_i มีค่าเป็น -1 จะทำให้ h(x) มีค่าน้อยกว่าหรือเท่ากับ –1 แสดงดังรูปที่ 2.4 โดยสามารถเขียนเป็นสมการได้ดังสมการที่ (2-3) ซึ่งเมื่อนำค่า y_i คูณเข้าไปแล้วจะทำให้เกิดเป็นสมการที่ (2-4) สำหรับทุก ๆ ค่าของ y_i

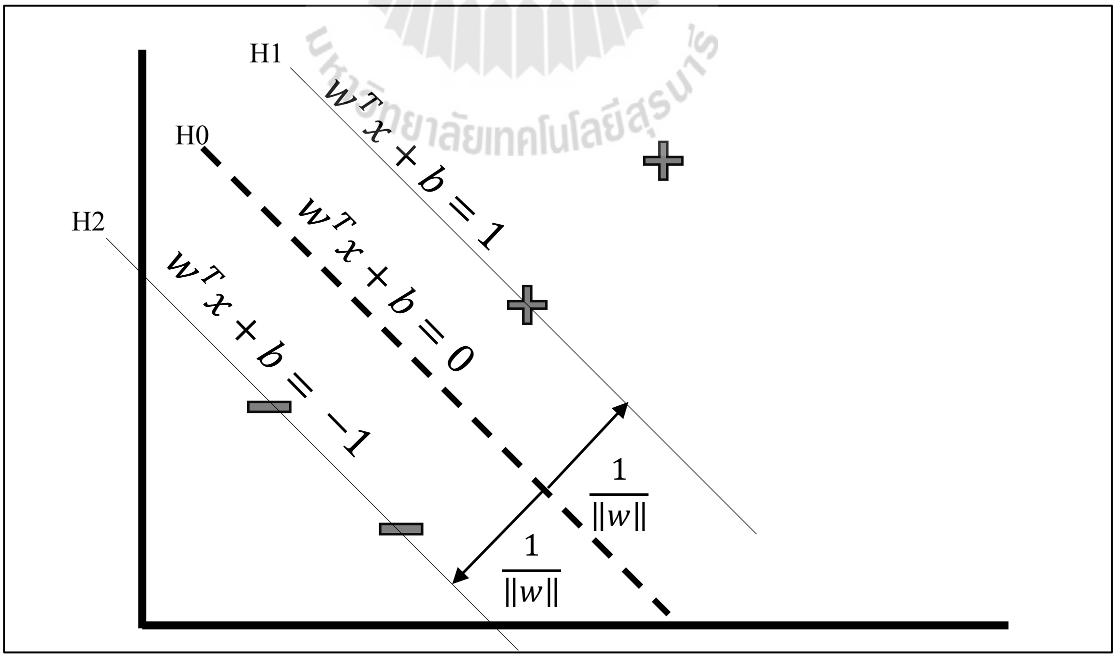
$$\begin{aligned} w^T x + b &\geq 1, when\ y_i = +1 \\ w^T x + b &\leq 1, when\ y_i = -1 \end{aligned} \tag{2-3}$$

$$y_i(w^T x + b) \geq 1 \tag{2-4}$$



รูปที่ 2.4 แสดงสมการไฮเปอร์เพลนและมาร์จิน

ดังนั้นระยะที่ตั้งฉากระหว่างไฮเปอร์เพลน (H0) และมาร์จินของข้อมูลในคลาส + (H1) สามารถแสดงได้ดังรูปที่ 2.5 ซึ่งสามารถคำนวณได้จาก $\frac{w^T x + b}{\|w\|} = \frac{1}{\|w\|}$ ดังนั้นระยะห่างระหว่าง H1 และ H2 หรือขอบของคลาส + และคลาส - มีค่า $\frac{2}{\|w\|}$



รูปที่ 2.5 แสดงขนาดของมาร์จิน

ในการจำแนกข้อมูลโดยการใช้ซัพพอร์ตเวกเตอร์แมชชีนนั้นจำเป็นที่ต้องให้ระยะห่างของไฮเปอร์เพลนกับมาร์จินมีค่ามากที่สุดเพื่อความแม่นยำในการจำแนกนั้นสามารถคำนวณค่าระยะห่างของมาร์จินกับไฮเปอร์เพลนได้จาก $\frac{1}{\|w\|}$ ซึ่งจะทำให้ได้ค่ามากที่สุดนั้นจำเป็นที่ $\|w\|$ จะต้องมีค่าน้อยที่สุดแสดงเงื่อนไขนี้ได้ดังสมการที่ (2-5) และ (2-6)

$$\text{Objective Function: } min_{w,b}\left\{\frac{\|w\|^2}{2}\right\} \tag{2-5}$$

$$\text{Linear Constraints: } y_i(w^T x + b) \geq 1, \forall x_i \in D \tag{2-6}$$

ในการแก้ปัญหาการหาค่าน้อยที่สุดจะใช้เทคนิค Lagrange Multipliers ในการแก้ปัญหาคือนำค่า α_i คูณเข้าไปในสมการ ซึ่งอยู่ภายใต้เงื่อนไขของ Karush-Kuhn-Tucker (KKT) แสดงได้ดังสมการที่ (2-7)

$$\alpha_i(y_i(w^T x + b) - 1) = 0 \tag{2-7}$$
$$and\ \alpha_i \geq 0$$

ในการหาค่า w และ b นั้นจะต้องคำนวณค่า α_i โดยที่ i = 1, 2, 3 , …, n หลังจากนั้นเราสามารถหาค่า w และ b ได้แต่อยู่ภายใต้เงื่อนไขของ KTT ซึ่งสามารถแปลงรูปได้ดังสมการที่ (2-8)

$$\alpha_i(y_i(w^T x + b) - 1) = 0 \tag{2-8}$$

จากการพิจารณาพารามิเตอร์ในสมการที่ (8) สามารถเกิดเหตุการณ์ขึ้นได้ 2 กรณีคือ

1. α_i มีค่าเป็น 0
2. $y_i(w^T x + b) - 1 = 0$ หรือหมายถึง $y_i(w^T x + b) = 1$

ซึ่งถ้าหากผลลัพธ์ที่ได้คือ α_i มีค่ามากกว่า 0 ก็จะเป็นดังกรณีที่ 2 แสดงว่าข้อมูลตัวนั้นจะเป็นข้อมูลซัพพอร์ตเวกเตอร์ แต่ถ้าหาก $y_i(w^T x + b) > 1$ ก็แสดงว่าจะเป็นกรณีที่ 1 คือข้อมูลนั้นจะไม่เป็นซัพพอร์ตเวกเตอร์ ดังนั้นเราสามารถคำนวณ w จากค่าซัพพอร์ตเวกเตอร์ได้ดังสมการที่ (2-9)

$$w = \sum_{i,\alpha_i > 0} \alpha_i y_i x_i \tag{2-9}$$

ดังนั้นในการหาค่าของ w นั้นจำเป็นที่ α_i จะต้องมีค่ามากกว่า 0 ดังนั้นเราสามารถแก้สมการเพื่อหาค่า b หรือค่าไบแอสได้ดังนี้

$$\alpha_i(y_i(w^T x + b) - 1) = 0$$
$$y_i(w^T x + b) = 1$$
$$b_i = \frac{1}{y_i} - w^T x = y_i - w^T x$$

เมื่อได้ค่า b_i แต่ละตัวเรียบร้อยแล้วทำการหาค่าเฉลี่ยของค่า b_i จะได้ดังสมการที่ (2-10)

$$b = avg_{\alpha>0}\{b_i\} \tag{2-10}$$

เทคนิคของอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนคือการหาข้อมูลซัพพอร์ตเวกเตอร์และทำการสร้างไฮเปอร์เพลนที่สามารถแบ่งข้อมูลแต่ละประเภทออกจากกันได้โดยมีมาร์จินกว้างที่สุดเท่าที่เป็นไปได้ โดยในการจำแนกสามารถจำแนกข้อมูลที่อยู่ใน D ได้ดังสมการที่ (2-11)

$$\hat{y} = sign\big(h(z)\big) = sign(w^T z + b) \tag{2-11}$$

โดย $\hat{y}$ คือคลาสที่ทำนายเป็น +1 หรือ -1 เท่านั้น

Sign(•) คือฟังก์ชันปรับค่าถ้าหากมีค่ามากกว่า 0 จะให้ค่าเป็น +1 และถ้ามีค่าน้อยกว่า 0 จะให้ค่าเป็น -1

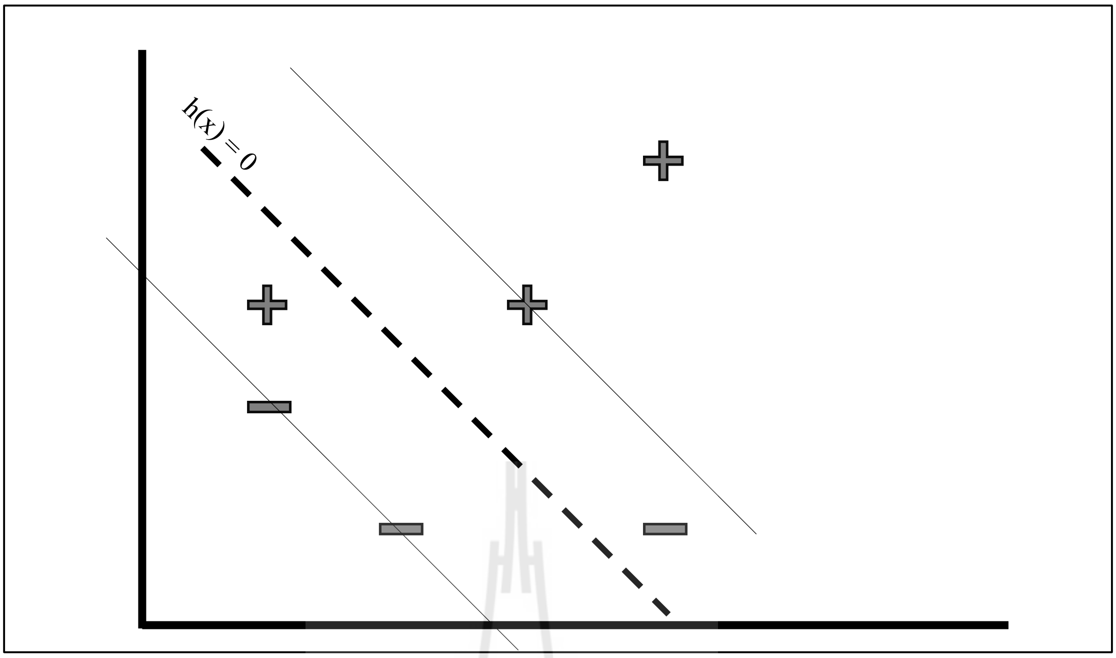
หลังจากที่ได้ไฮเปอร์เพลนและมาร์จินสำหรับการจำแนกประเภทข้อมูลเรียบร้อยแล้วเพื่อความยืดหยุ่นของอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน ได้มีการพัฒนาเทคนิคซอฟต์มาจิน (Soft Margin) ขึ้นมาเพื่อเพิ่มความยืดหยุ่นของไฮเปอร์เพลนซึ่งแสดงในรูปที่ 2.6 ซึ่งจะเห็นได้ว่าไฮเปอร์เพลน h(x) นั้นอนุญาตให้ข้อมูล + สามารถอยู่ฝั่งด้านข้อมูล – ได้และข้อมูล – ก็อยู่ด้าน + ได้ด้วยเช่นเดียวกัน

ซึ่งเทคนิคซอฟต์มาจินนั้นจะทำการเพิ่มค่า Slack Variable เข้าไปเพื่อให้ซัพพอร์ตเวกเตอร์แมชชีนมีความยืดหยุ่นขึ้น โดยไฮเปอร์เพลนนั้นสามารถหาได้จากสมการที่ (2-4) และเทคนิคซอฟต์มาจินนั้นจะเพิ่มค่า Slack Variable เข้าไปในสมการที่ (2-4) ได้ดังสมการที่ (2-12)

$$y_i(w^T x + b) \geq 1 - \xi_i \tag{2-12}$$

โดยกำหนดให้ ξ_i คือค่า Slack Variable และเมื่อ $\xi_i > 0$ จะทำให้เกิดขึ้นได้ทั้งหมด 3 กรณีดังต่อไปนี้

1. $\xi_i = 0$ หมายความว่าข้อมูลที่ไม่ได้อยู่ในมาร์จินและอยู่ถูกฝั่ง
2. $0 < \xi_i < 1$ หมายความว่าข้อมูลนั้นอยู่ภายในมาร์จินและอยู่ถูกฝั่ง
3. $\xi_i \geq 1$ หมายความว่าข้อมูลนั้นเป็นข้อมูลที่จำแนกผิด หรือก็คือข้อมูลอยู่ผิดฝั่ง



รูปที่ 2.6 แสดงเทคนิคซอฟต์มาจิน

เมื่อทำการเพิ่มค่า Slack Variable แล้วในการจำแนกประเภทของข้อมูลจะสามารถแสดงได้ดังสมการที่ (2-13) และ (2-14)

Objective Function: $$\mathit{min}_{w,b,\xi_i}\left\{\frac{\|w\|^2}{2}+C\sum_{i=1}^{n}\left(\xi_i\right)^k\right\} \quad (2\text{-}13)$$

Linear Constraints: $$y_i(w^T x+b)\geq 1-\ \xi_i, \forall x_i \in D \quad (2\text{-}14)$$

$$\xi_i > 0\ \forall x_i \in D$$

โดยที่ค่า C และ k คือค่าที่บ่งบอกถึงการจำแนกผิดประเภทซึ่งค่า C จะบ่งบอกถึงจำนวนข้อมูลที่อนุโลมให้ทำนายผิดได้ ยกตัวอย่างเช่นถ้าหากค่า C = 2 ก็คืออนุโลมให้ทำนายผิดได้ 2 ข้อมูลซึ่งถ้าหากค่า C เป็นค่ามาก ๆ แล้วจะส่งผลกระทบกับการจำแนกข้อมูลได้ ส่วนค่า k นั้นจะสามารถกำหนดได้เพียง 2 ค่าเท่านั้นคือให้ค่า k มีค่าเป็น 1 หรือ 2 เท่านั้นถ้าหากกำหนดค่า k ให้มีค่าเท่ากับ 1 จะหมายความว่าให้หาผลรวมที่น้อยที่สุดของ Slack Variable ถ้าหากกำหนดค่า k ให้มีค่าเท่ากับ 2 จะหมายความว่าให้หาผลรวมที่น้อยที่สุดของ Slack Variable กำลัง 2

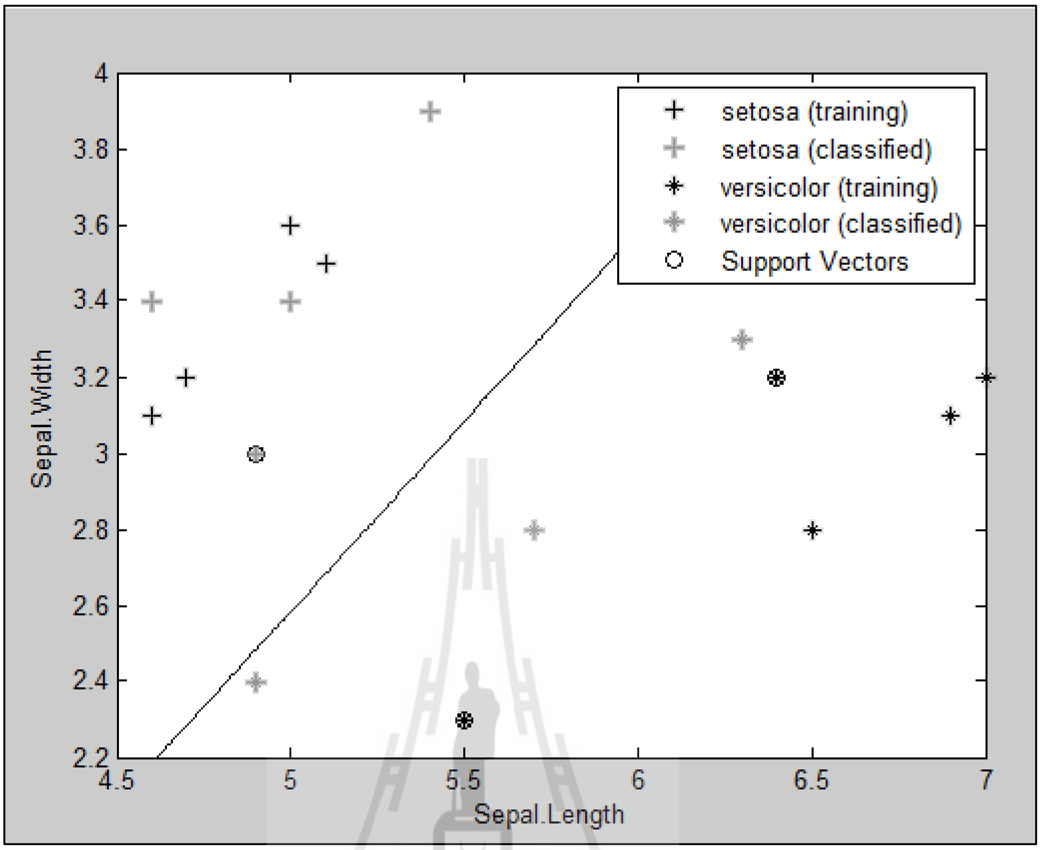
เนื่องจากข้อมูลบางชนิดไม่เหมาะกับการใช้สมการเส้นตรงในการจำแนกข้อมูลจึงได้มีการพัฒนาเคอร์เนล (Kernel) หรือฟังก์ชันแก่น (Chistianini and Shawe-Taylor, 2000; Muller et al, 2001; Scholkopf et al, 1999) ที่ใช้กับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนขึ้น ซึ่งเคอร์เนลต่าง ๆ นั้นจะเป็นการใช้สมการต่าง ๆ เพื่อกำหนดไฮเปอร์เพลนในรูปแบบอื่น ๆ ที่ไม่ใช่เส้นตรง ยกตัวอย่างเช่นเส้นโค้ง (Polynomial) ซึ่งเคอร์เนลที่นิยมใช้ในการจำแนกข้อมูลต่าง ๆ แสดงในตารางที่ 2.3

ตารางที่ 2.3 เคอร์เนลฟังก์ชันสำหรับซัพพอร์ตเวกเตอร์แมชชีน

Kernel	Inner Product Kernel
Linear	$x^T x_i$
Polynomial	$(x^T x_i + n)^d$
Radial-basis function	$\exp(-\gamma\|x - x_i\|^2), \gamma > 0$
Two layer perceptron	$\tanh(\beta_0 x^T x_i + \beta_1)$

ถ้าหากนำข้อมูลจากข้อมูลในตารางที่ 2.2 มาทำการสร้างโมเดลในการจำแนกด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนจะได้ดังรูปที่ 2.7 ซึ่งในรูปในแนวแกน X คือค่า Sepal.Length ส่วนในแนวแกน Y คือค่า Sepal.Width และมีไฮเปอร์เพลนซึ่งก็คือเส้นสีดำที่เป็นเส้นที่ใช้ในการจำแนกข้อมูลซึ่งหาได้จากการใช้ซัพพอร์ตเวกเตอร์ โดยในที่นี้จะมีข้อมูลซัพพอร์ตเวกเตอร์ 3 ข้อมูล โดยเป็นข้อมูลของคลาส setosa จำนวน 1 ข้อมูลและข้อมูลคลาส versicolor จำนวน 2 ข้อมูล ซึ่งข้อมูลที่เป็นซัพพอร์ตเวกเตอร์คือข้อมูลในภาพที่มีวงกลมล้อมรอบ โดยข้อมูลคลาส setosa นั้นจะแทนด้วยสัญลักษณ์ + ซึ่งถ้าหากเป็น + สีแดงคือข้อมูลฝึกสอนและ + สีม่วงคือข้อมูลทดสอบ ส่วนข้อมูลคลาส versicolor จะแทนด้วยสัญลักษณ์ * ซึ่งถ้าหากเป็น * สีเขียวคือข้อมูลฝึกสอนและ * สีฟ้าคือข้อมูลทดสอบ จากไฮเปอเพลนที่สร้างขึ้นนั้นสามารถจำแนกข้อมูลทดสอบได้ถูกต้องจำนวน 6 ข้อมูลจากข้อมูลทดสอบทั้งหมด 6 ข้อมูล ซึ่งหมายถึงประสิทธิภาพในการจำแนกประเภทของข้อมูลดอกไอริสจากตารางที่ 2.2 นั้นมีค่าความถูกต้อง (6/6) 100%

อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนนั้นเป็นอัลกอริทึมที่นิยมใช้ในการจำแนกประเภทของข้อมูลและมีงานวิจัยต่าง ๆ ที่ทำการศึกษาเกี่ยวกับอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน เช่น งานวิจัยของ Kong et al. (2015) นั้นทำการศึกษาเกี่ยวกับการเพิ่มประสิทธิภาพความรวดเร็วในการจำแนกของอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน ด้วยเทคนิคการคัดเลือกฟีเจอร์เพื่อลดจำนวนข้อมูลในการวิเคราะห์ทำให้อัลกอริทึมสามารถทำงานได้รวดเร็วยิ่งขึ้น งานวิจัยของ Shao et al. (2011) ได้



รูปที่ 2.7 แสดงการจำแนกข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน

พัฒนาอัลกอริทึมใหม่ชื่อว่า Twin Bounded Support Vector Machine (TBSVM) โดยอาศัยพื้นฐานจากอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน โดยอัลกอริทึม TBSM นี้จะใช้วิธีการหาไฮเปอร์เพลนด้วยเทคนิค Successive over-relaxation (SOR) ซึ่งทำให้สามารถทำการสร้างไฮเปอร์เพลนได้รวดเร็วกว่าการใช้ซัพพอร์ตเวกเตอร์แมชชีนแบบปกติ และงานวิจัยของ Peng and Xu (2012) เสนอให้ใช้การวัดระยะทางแบบ Mahalanobis ในการวัดระยะห่างระหว่างคลาสเพื่อสร้างไฮเปอร์เพลนโดยที่อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนที่ใช้เทคนิคนี้จะทำให้สามารถทำงานได้รวดเร็วกว่าปกติ งานวิจัยของ Batuwita and Palade (2010) ได้ใช้วิธีการสุ่มเพิ่มกับข้อมูลไม่สมดุลเพื่อใช้กับอัลกอริทึม SVM ได้อย่างมีประสิทธิภาพโดยการลดข้อมูล 75% จากข้อมูลทั้งหมดแล้วนำข้อมูลนั้นมาทำการสุ่มเพิ่มข้อมูลในคลาสส่วนน้อยก่อนที่จะนำไปใช้กับอัลกอริทึม SVM ซึ่งประสิทธิภาพในการจำแนกดีกว่าไม่ลดข้อมูล

2.3 การวัดระยะห่างระหว่างข้อมูล

งานวิจัยของวิทยานิพนธ์ฉบับนี้ใช้แนวทางการแยกข้อมูลที่มีการซ้อนทับระหว่างข้อมูลต่างคลาสออกมา จากข้อมูลที่ไม่มีการซ้อนทับ และในการพิจารณาการซ้อนทับของข้อมูลต่างคลาสจะใช้การวัดระยะห่างด้วยวิธีแบบ Euclidean และแบบ Hausdorff

การคำนวณหาระยะห่างระหว่างข้อมูลด้วย Euclidean Distance (Lee et al., 2014) เป็นการคำนวณระยะห่างแบบทั่วไปซึ่งเป็นการคำนวณระยะห่างแบบกระจัดที่จะไม่มีค่าติดลบซึ่งสามารถคำนวณได้จากสมการที่ (2-15) โดยกำหนดให้ข้อมูลมีมิติของข้อมูลทั้งหมด n มิติ และต้องการหาระยะห่างระหว่างข้อมูล 2 ข้อมูล คือ $p = (p_1, p_2, \ldots, p_n)$ และ $q = (q_1, q_2, \ldots, q_n)$

$$Dist(p,q) = Dist(q,p)$$

$$Dist(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2}$$

$$Dist(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad (2\text{-}15)$$

การหาระยะห่างระหว่างข้อมูลแบบ Euclidean Distance ด้วยตัวอย่างข้อมูล A, B, C, D, E, F แสดงดังตารางที่ 2.4 ซึ่งเป็นการหาระยะห่างระหว่างคู่ของข้อมูลทั้งหมดโดยใช้สมการที่ (15)

ตารางที่ 2.4 ตัวอย่างการหาระยะห่างระหว่างข้อมูลแบบ Euclidean

ข้อมูล	A(3,8,4)	B(6,1,4)	C(8,0,9)	D(4,3,9)	E(2,1,5)	F(3,4,5)
A(3,8,4)	0.00	7.62	2.24	5.92	3.16	8.60
B(6,1,4)	7.62	0.00	8.31	2.24	5.66	9.90
C(8,0,9)	2.24	8.31	0.00	6.48	3.61	10.34
D(4,3,9)	5.92	2.24	6.48	0.00	4.58	8.77
E(2,1,5)	3.16	5.66	3.61	4.58	0.00	10.30
F(3,4,5)	8.60	9.90	10.34	8.77	10.30	0.00

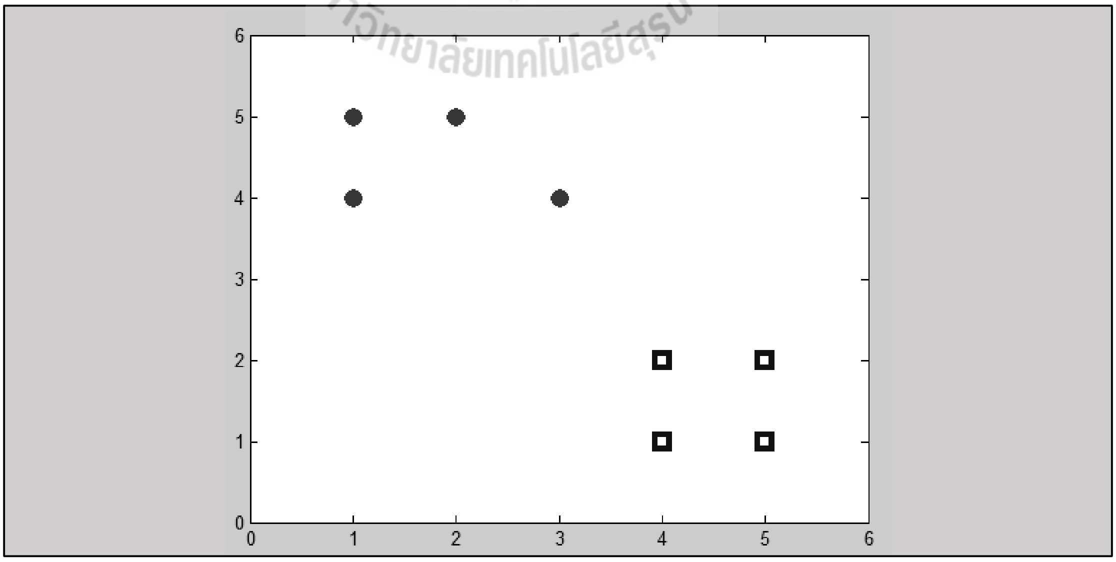
การคำนวณระยะห่างแบบ Hausdorff Distance (Fischer et al., 2015) นั้นจะอาศัยหลักการหาระยะห่างแบบ Euclidean Distance เป็นพื้นฐานในการวัดระยะห่างโดยมีการปรับปรุงเพิ่มเติมให้การวัดระยะห่างแบบ Hausdorff Distance นั้นจะเป็นการวัดระยะเพื่อหาระยะห่างที่มากที่สุดของกลุ่มข้อมูล 2 กลุ่มโดยเมื่อหาเรียบร้อยแล้วจะมีค่าเพียงค่าเดียว ซึ่งวิธีการหาระยะห่างแบบนี้สามารถอธิบายได้ด้วยสมการที่ (2-16)

$$h(A,B) = \max(max_{a\in A}min_{b\in B}dist(a,b), max_{b\in B}min_{a\in A}dist(a,b)) \quad (2\text{-}16)$$

การอธิบายขั้นตอนการคำนวณระยะทางด้วย Hausdorff Distance จะใช้ข้อมูลในตารางที่ 2.5 ในการแสดงตัวอย่างซึ่งมีข้อมูล 2 กลุ่มคือ A และ B โดยสามารถแสดงเป็นรูปภาพได้ดังรูปที่ 2.8 ซึ่งข้อมูลกลุ่ม A จะแสดงด้วยวงกลม และข้อมูลกลุ่ม B จะแสดงด้วยสี่เหลี่ยม

ตารางที่ 2.5 แสดงข้อมูลตัวอย่างในการคำนวณ Hausdorff Distance

A		B	
X	Y	X	Y
1	4	4	1
1	5	4	2
2	5	5	1
3	4	5	2



รูปที่ 2.8 ภาพข้อมูลในการคำนวณ Hausdorff Distance

ขั้นตอนแรกจะเป็นการหาระยะทางที่เกิดขึ้นทั้งหมดระหว่างข้อมูลทั้งสองกลุ่มคือข้อมูลกลุ่ม A และกลุ่ม B ซึ่งจะเป็นการหาระยะห่างแบบ Euclidean ระหว่างข้อมูลทุก ๆ คู่ของข้อมูล ระยะห่างที่คำนวณได้แสดงในตารางที่ 2.6

ตารางที่ 2.6 แสดงระยะห่างระหว่างข้อมูลทั้งหมด

B A	(4,1)	(4,2)	(5,1)	(5,2)
(1,4)	4.24	3.61	5.00	4.47
(1,5)	5.00	4.24	5.66	5.00
(2,5)	4.47	3.61	5.00	4.24
(3,4)	3.16	2.24	3.61	2.83

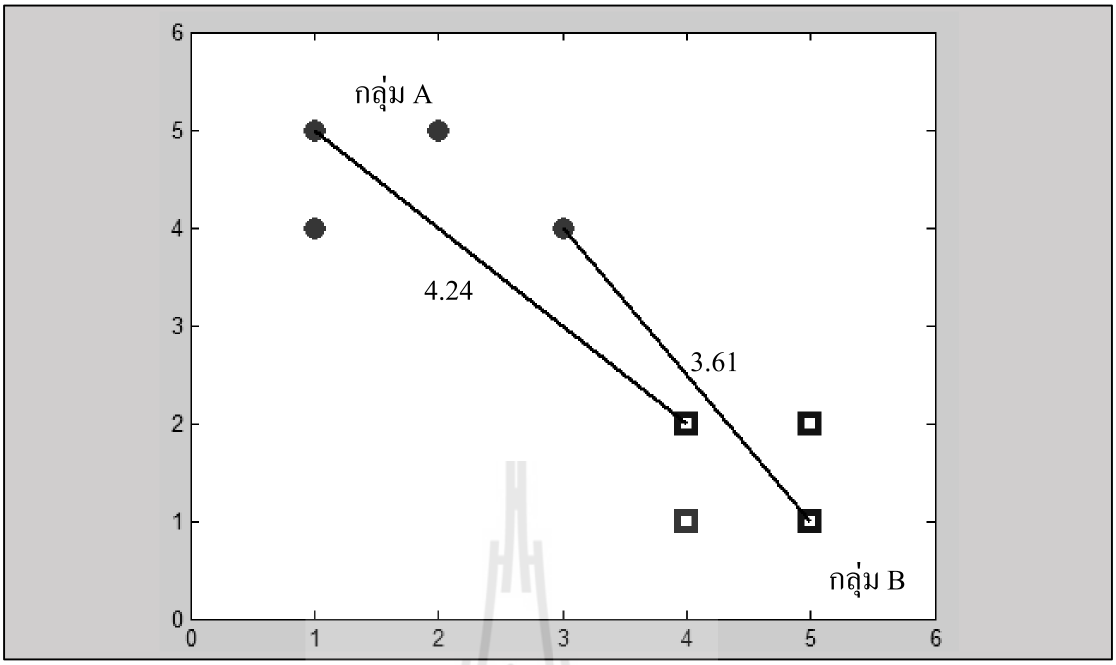
ขั้นตอนต่อไปจะเป็นการหาค่า $max_{a\in A}min_{b\in B}dist(a,b)$ ซึ่งจะแสดงในตารางที่ 2.7 ซึ่งมีค่าเท่ากับ 4.24 ส่วนค่า $max_{b\in B}min_{a\in A}dist(a,b)$ แสดงในตารางที่ 2.8 มีค่าเท่ากับ 3.61 สามารถแสดงได้ดังรูปที่ 2.7

ตารางที่ 2.7 แสดงระยะห่างระหว่างข้อมูล A และ B ได้ Hausdorff distance = 4.24

B A	(4,1)	(4,2)	(5,1)	(5,2)
(1,4)	4.24	**3.61**	5.00	4.47
(1,5)	5.00	**4.24**	5.66	5.00
(2,5)	4.47	**3.61**	5.00	4.24
(3,4)	3.16	**2.24**	3.61	2.83

ตารางที่ 2.8 แสดงระยะห่างระหว่างข้อมูล B และ A ได้ Hausdorff distance = 3.61

B A	(4,1)	(4,2)	(5,1)	(5,2)
(1,4)	4.24	3.61	5.00	4.47
(1,5)	5.00	4.24	5.66	5.00
(2,5)	4.47	3.61	5.00	4.24
(3,4)	**3.16**	**2.24**	**3.61**	**2.83**



รูปที่ 2.9 แสดงระยะทางของ Hausdorff Distance : h(A,B) = 4.24 และ h(B,A) = 3.61

ดังนั้นจากรูปที่ 2.9 สามารถสรุปได้ว่า h(A,B) นั้นมีค่าเท่ากับ 4.24 โดยที่เป็นระยะห่างระหว่างกลุ่ม A ที่ข้อมูลมีค่าเท่ากับ (1,5) กับข้อมูลกลุ่ม B ที่มีค่าเท่ากับ (4,2) ในขณะที่ h(B,A) มีค่าเท่ากับ 3.61 เนื่องจากเป็นการวัดจากกลุ่ม B ที่ข้อมูล (5,1) ไปยังกลุ่ม A ที่ข้อมูล (3,4)

2.4 เกณฑ์ที่ใช้ในการวัดประสิทธิภาพการจำแนก

เกณฑ์ที่ใช้ในการวัดประสิทธิภาพในการจำแนกประเภทข้อมูลได้มาจากการเปรียบเทียบคลาสที่ได้จากการทำนายมาเปรียบเทียบกับคลาสที่แท้จริงของข้อมูลโดยแสดงผลที่เป็นไปได้จากการทำนายในลักษณะของเมตริกซ์ดังตารางที่ 2.8

ตารางที่ 2.8 แสดง confusion matrix ของค่าทำนาย (Prediction) เปรียบเทียบกับค่าจริง (Actual)

Prediction Actual	Positive	Negative
Positive	TP	FN
Negative	FP	TN

จากตารางที่ 2.8 ซึ่งเป็นตารางที่เป็นพื้นฐานในการคำนวณแบบต่าง ๆ เพื่อวัดประสิทธิภาพในการจำแนกประเภทของข้อมูลซึ่งมีรายละเอียดดังต่อไปนี้

TP คือ จำนวนข้อมูลที่ทำนายถูกว่าเป็นคลาส Positive ยกตัวอย่างเช่น ทำนายว่าวันพรุ่งนี้ฝนจะตก แล้วผลปรากฏว่าพรุ่งนี้ฝนตก

FN คือ จำนวนข้อมูลที่ทำนายผิดว่าเป็นคลาส Negative ยกตัวอย่างเช่น ทำนายว่าวันพรุ่งนี้ฝนจะไม่ตก แล้วผลปรากฏว่าพรุ่งนี้ฝนตก

FP คือ จำนวนข้อมูลที่ทำนายผิดว่าเป็นคลาส Positive ยกตัวอย่างเช่น ทำนายว่าวันพรุ่งนี้ฝนจะตก แล้วผลปรากฏว่าพรุ่งนี้ฝนไม่ตก

TN คือ จำนวนข้อมูลที่ทำนายถูกว่าเป็นคลาส Negative ยกตัวอย่างเช่น ทำนายว่าวันพรุ่งนี้ฝนจะไม่ตก แล้วผลปรากฏว่าพรุ่งนี้ฝนไม่ตก

ค่าความแม่นในการจำแนกประเภทข้อมูล (Accuracy) ซึ่งเป็นความถูกต้องโดยรวมของทุกคลาสนั้นสามารถคำนวณได้จากสมการที่ (2-17)

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2-17}$$

ค่าความแม่นในการจำแนกประเภทข้อมูลของคลาส Positive นั้นสามารถคำนวณได้จากสมการที่ (2-18)

$$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP + FN} \tag{2-18}$$

ค่าความแม่นในการจำแนกประเภทข้อมูลของคลาส Negative นั้นสามารถคำนวณได้จากสมการที่ (2-19)

$$True\ Negative\ Rate\ (TNR) = \frac{TN}{FP + TN} \tag{2-19}$$

ค่าความแม่นเฉลี่ยในการจำแนกประเภทข้อมูลของคลาส Positive และ Negative นั้นสามารถหาได้จากสมการ (2-20)

$$G - Means = \sqrt{TPR \bullet TNR} \tag{2-20}$$

ค่าความแม่นเฉลี่ยในการจำแนกประเภทข้อมูลของคลาส Positive ซึ่งเป็นการหาค่าเฉลี่ยระหว่าง TPR กับค่า Precision ซึ่งสามารถหาได้จากสมการที่ (2-21)

$$\begin{gathered} Precision = \frac{TP}{TP + FP} \\ F - Measure = \frac{2 \bullet TPR \bullet Precision}{TPR + Precision} \end{gathered} \tag{2-21}$$

ค่าความไม่สมดุลของข้อมูล (Imbalanced Ratio) สามารถหาได้จากจำนวนข้อมูลส่วนมากหารด้วยจำนวนข้อมูลส่วนน้อย ซึ่งแสดงดังสมการที่ (2-22)

$$IR = \frac{n_{majority}}{n_{minority}} \tag{2-22}$$

ค่าความซ้อนทับกันของข้อมูล (Overlaped Ratio) สามารถหาได้จากค่าเฉลี่ยของการซ้อนทับของแต่ละแอตทริบิวต์ กำหนดให้มีจำนวนแอตทริบิวต์เท่ากับ n แสดงได้ดังสมการ (2-23)

$$OR = \left(\frac{\sum_1^n \frac{Attr\ Overlap\ Range}{Attr\ Range}}{n} \right) x\ 100 \tag{2-23}$$

2.5 งานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้จะอธิบายถึงงานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทของข้อมูลที่ไม่สมดุล โดยมีนักวิจัยได้นำเสนอเทคนิคและวิธีการต่าง ๆ ในการจำแนกประเภทข้อมูลไม่สมดุลได้อย่างมีประสิทธิภาพ ซึ่งงานวิจัยที่ผู้วิจัยได้ศึกษานั้นผู้วิจัยสามารถสรุปงานวิจัยเหล่านั้นได้ดังต่อไปนี้

He and Ghodsi (2010) ได้ทำการศึกษาและวิจัยเกี่ยวกับการจำแนกประเภทข้อมูลที่ไม่สมดุลโดยใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนร่วมกับวิธีการสุ่มเพิ่มและวิธีการสุ่มลด เพื่อใช้ในการปรับปรุงข้อมูลที่ไม่สมดุล แล้วจึงทำการจำแนกประเภทข้อมูล ซึ่งข้อมูลที่ใช้ในงานวิจัยนี้ได้นำมาจากแหล่งข้อมูลมาตรฐาน UCI โดยใช้ข้อมูลทั้งหมด 5 ข้อมูลซึ่งผลสรุปของงานวิจัยนั้นสามารถสรุปได้ว่าเทคนิคที่งานวิจัยนี้นำเสนอสามารถจำแนกประเภทข้อมูลที่ไม่สมดุลได้ดี แต่ในงานวิจัยนี้ไม่ได้แสดงถึงความแม่นในการจำแนกคลาสส่วนน้อย

Yong (2012) ศึกษาและวิจัยเกี่ยวกับการจำแนกประเภทข้อมูลที่ไม่สมดุลโดยมุ่งเน้นไปที่การเพิ่มประสิทธิภาพการจำแนกข้อมูลคลาสส่วนน้อย โดยงานวิจัยนี้ได้ใช้เทคนิคการจัดกลุ่มข้อมูลคลาสน้อยด้วยวิธี K-Means แล้วทำการสุ่มเพิ่มด้วยเทคนิค Genetic Algorithm ซึ่งเป็นการสร้างข้อมูลคลาสส่วนน้อยใหม่ ให้มีขนาดใกล้เคียงกับข้อมูลในคลาสส่วนมาก หลังจากนั้นใช้อัลกอริทึม k-NN (k-Nearest Neighbors) และซัพพอร์ตเวกเตอร์แมชชีนในการจำแนกข้อมูล ซึ่งข้อมูลที่ใช้ในงานวิจัยนี้ใช้ข้อมูลทั้งหมด 4 ชุดข้อมูลโดยนำข้อมูลทั้งหมดมาจากฐานข้อมูล UCI เกณฑ์ในการวัดประสิทธิภาพในงานวิจัยนี้ใช้เป็นค่าเฉลี่ยของ Accuracy ทั้งหมด 10 ครั้ง ซึ่งสามารถสรุปผลการวิจัยนี้ได้ว่า เทคนิคที่นำเสนอสามารถเพิ่มประสิทธิภาพในการจำแนกข้อมูลไม่สมดุลได้ดีขึ้นเมื่อเปรียบเทียบกับการจำแนกแบบปกติ งานวิจัยนี้มุ่งเน้นเฉพาะความแม่นในการจำแนกข้อมูลเท่านั้น ไม่ได้ให้ความสำคัญกับความแม่นในการจำแนกคลาสส่วนน้อย

Cateni et al. (2014) ได้นำเสนอเทคนิคในการจำแนกประเภทข้อมูลที่ไม่สมดุลและมีคลาสจำนวน 2 คลาส โดยได้เสนอเทคนิคที่ใช้วิธีการสุ่มเพิ่มและวิธีการสุ่มลดให้ทำงานร่วมกันแล้วทำการใช้อัลกอริทึมในการจำแนกทั้งหมด 4 ชนิดด้วยกันคือ ซัพพอร์ตเวกเตอร์แมชชีน ต้นไม้ตัดสินใจ (Decision Tree) Self-Organizing Map และ Bayesian Classifiers โดยได้ใช้ข้อมูลทั้งหมด 2 แบบคือ ข้อมูลที่สังเคราะห์ขึ้นเอง และข้อมูลจริงจากฐานข้อมูล UCI ทีมวิจัยนี้ได้ใช้วิธีแบ่งข้อมูลฝึกสอนและข้อมูลทดสอบโดยมีอัตราส่วนของข้อมูลทั้ง 2 คลาสที่ใกล้เคียงกัน โดยแบ่งข้อมูล 75% เป็นข้อมูลฝึกสอน ส่วนอีก 25% เป็นข้อมูลทดสอบ ผลสรุปของงานวิจัยนี้สามารถสรุปได้ว่าวิธีที่นำเสนอมานั้นสามารถจำแนกประเภทข้อมูลส่วนน้อยในข้อมูลที่ไม่สมดุลได้อย่างมีประสิทธิภาพ ในงานวิจัยนี้เป็นการเปรียบเทียบประสิทธิภาพระหว่างการใช้เทคนิคปกติ และเทคนิคที่นำเสนอเท่านั้น ขาดการเปรียบเทียบประสิทธิภาพกับงานวิจัยอื่น

Datta and Das (2015) ได้เสนอเทคนิคใหม่เพื่อจำแนกประเภทข้อมูลในข้อมูลที่ไม่สมดุล โดยเรียกเทคนิคใหม่นี้ว่า Near-Bayesian Support Vector Machine (NBSVM) ซึ่งเป็นเทคนิคที่พัฒนาอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนให้สามารสร้างไฮเปอร์เพลนที่สามารแบ่งข้อมูลที่ไม่สมดุลได้ดีกว่าอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนแบบปกติ ข้อมูลที่ใช้ในงานวิจัยมีทั้งหมด 15 ข้อมูลซึ่งได้นำมาจากฐานข้อมูลทั้งหมด 3 ฐานข้อมูลได้แก่ UCI, IDA และ Statlog โดยใช้เกณฑ์ในการประเมินประสิทธิภาพของการจำแนกคือ Accuracy, G-Means ด้วยวิธีการทดสอบแบบ 10 Fold Cross Validation ผลสรุปของงานวิจัยนี้สามารถสรุปได้ว่าเทคนิคที่งานวิจัยนี้นำเสนอสามารถจำแนกประเภทข้อมูลที่ไม่สมดุลได้อย่างมีประสิทธิภาพ แต่ยังขาดการเปรียบเทียบความแม่นในการจำแนกคลาสส่วนน้อย ซึ่งถ้ามีการเปรียบเทียบความแม่นในการจำแนกคลาสส่วนน้อยจะทำให้งานวิจัยนี้น่าสนใจมากขึ้น

Zhang and Li (2014) ศึกษาและวิจัยเกี่ยวกับการจำแนกประเภทข้อมูลที่ไม่สมดุลโดยอาศัยเทคนิคการสุ่มเพิ่ม ซึ่งทีมวิจัยนี้ได้เสนอเทคนิคใหม่ในการสุ่มเพิ่มคือ Random Walk Over-Sampling (RWO-Sampling) ในการสุ่มเพิ่มนี้จะทำการเพิ่มข้อมูลในคลาสส่วนน้อยให้มีจำนวนใกล้เคียงกับคลาสส่วนมาก โดยจะไม่มีผลกระทบหรือการเปลี่ยนแปลงโครงสร้างของข้อมูลส่วนน้อย กล่าวคือข้อมูลที่สุ่มเพิ่มจะไม่ทำให้ค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐานของข้อมูลคลาสส่วนน้อยต่างไปจากเดิม โดยข้อมูลที่ใช้ในงานวิจัยนำมาจากฐานข้อมูล UCI และใช้ข้อมูลทั้งหมด 21 ชุดข้อมูล และใช้อัลกอริทึมในการจำแนก 3 ชนิดได้แก่ ต้นไม้ตัดสินใจ ซัพพอร์ตเวกเตอร์แมชชีน และนาอีฟเบย์ ด้วยการใช้เทคนิคการทดสอบแบบ 10 Fold Cross Validation ในการประเมินประสิทธิภาพของการจำแนกประเภทข้อมูล ซึ่งโดยสรุปแล้วเทคนิคที่งานวิจัยนี้นำเสนอสามารถเพิ่มประสิทธิภาพในการจำแนกข้อมูลที่ไม่สมดุลได้ดี

Gao et al. (2014) เสนอเทคนิคใหม่ในการจำแนกประเภพข้อมูลที่ไม่สมดุลเทคนิคใหม่นี้มีชื่อว่า PDFOS ซึ่งใช้แนวคิดของการสุ่มข้อมูลในคลาสส่วนน้อยเพิ่มขึ้น โดยอาศัยหลักการการประมาณค่าความหนาแน่นของข้อมูลส่วนน้อยเป็นหลัก จากนั้นจะเป็นการปรับสมดุลของข้อมูลที่ไม่สมดุล เพื่อให้การจำแนกประเภทข้อมูลสามารถทำงานได้อย่างมีประสิทธิภาพ ซึ่งข้อมูลที่ใช้ในงานวิจัยนี้มีทั้งหมด 6 ชุดข้อมูลโดยนำข้อมูลมาจากฐานข้อมูล UCI โดยมีการเปรียบเทียบประสิทธิภาพในการจำแนกด้วยวิธีทดสอบแบบ cross-validations โดยใช้เกณฑ์ในการวัดประสิทธิภาพได้แก่ G-Means, F-Measures โดยสรุปแล้วเทคนนิคที่งานวิจัยนี้นำเสนอนั้นมีประสิทธิภาพในการจำแนกข้อมูลที่ไม่สมดุล แต่ไม่ได้สรุปในเรื่องของความแม่นยำในการจำแนกคลาสส่วนน้อย

Piyanoot et al. (2015) ศึกษาเกี่ยวกับเทคนิคการจำแนกข้อมูลที่ไม่สมดุลโดยเสนอเทคนิคในการแบ่งข้อมูลที่ไม่สมดุลออกเป็น 3 ส่วนย่อย ๆ ได้แก่ ข้อมูลที่ไม่ซ้อนทับกัน (Non-Overlapped) ข้อมูลที่ซ้อนทับ (Overlapped) และขอบของข้อมูลที่ซ้อนทับ (Borderline) แนวคิดของงานวิจัยนี้จะเป็นการนำส่วนย่อยในแต่ละส่วนไปสร้างโมเดลในการจำแนกกล่าวคือมี 3 โมเดลโดยในแต่ละส่วนนั้นจะใช้อัลกอริทึมในการจำแนกที่แตกต่างกัน โดยในงานวิจัยนี้ได้ใช้ข้อมูลที่สังเคราะห์ขึ้นเองและข้อมูลที่นำมาจากฐานข้อมูลมาตรฐาน KEEL และ UCI ซึ่งมีชุดข้อมูลที่นำมาทดสอบทั้งหมด 21 ชุดข้อมูลเป็นข้อมูลสังเคราะห์ 13 ชุดข้อมูลและเป็นชุดข้อมูลจากฐานข้อมูลมาตรฐาน 8 ชุดข้อมูล ซึ่งใช้เกณฑ์ในการวัดประสิทธิภาพคือ TP rate, F-Measure, G-Means โดยสรุปแล้วเทคนิคที่งานวิจัยนี้นำเสนอในการจำแนกข้อมูลที่ไม่สมดุลมีประสิทธิภาพในการจำแนกที่ดีสำหรับข้อมูลสังเคราะห์ แต่เมื่อใช้ชุดข้อมูลจริงจากฐานข้อมูลกลับมีประสิทธิภาพต่ำกว่าการใช้ข้อมูลสังเคราะห์

แนวคิดของวิทยานิพนธ์ฉบับนี้ใช้หลักการพื้นฐานเหมือนงานวิจัยของ Piyanoot et al. (2015) ที่พิจารณาบริเวณซ้อนทับ และไม่ซ้อนทับของข้อมูลต่างคลาส แต่ใช้เทคนิคการแบ่งข้อมูลและอัลกอริทึมที่ใช้ในการจำแนกประเภทข้อมูลที่แตกต่างกัน โดยงานวิจัยในวิทยานิพนธ์เล่มนี้เสนอแนวทางการแบ่งข้อมูลเป็นส่วนย่อย ก่อนที่จะสร้างโมเดลในการจำแนกข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน

ตารางที่ 2.10 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลส่วนน้อยในข้อมูลไม่สมดุล

กระบวนการทำงาน	งานวิจัยที่เกี่ยวข้อง							
	ก	ข	ค	ง	จ	ฉ	ช	ญ*
อัลกอริทึมที่เกี่ยวข้อง								
Support Vector Machine	✓	✓	✓	✓			✓	✓
Decision Tree			✓		✓			
DBSCAN							✓	
RBF							✓	
RBFN							✓	
Genetic Algorithm		✓						
K-Means						✓		
Naïve Bayes					✓			
KNN		✓						
เสนออัลกอริทึมใหม่				✓		✓		
เทคนิคที่ใช้จัดการกับข้อมูล								
Over-Sampling	✓	✓	✓		✓	✓		
Under-Sampling	✓		✓					
แบ่งข้อมูล 3 ส่วนด้วยเทคนิค Hausdorff							✓	
แบ่งข้อมูล 2 ส่วนด้วยเทคนิค Euclidean								✓
มาตรวัดที่ใช้วัดประสิทธิภาพของโมเดล								
Accuracy		✓	✓	✓	✓			✓
F-Measure	✓				✓	✓	✓	✓
G-Means	✓			✓	✓	✓	✓	✓
TP Rate					✓		✓	✓
TN Rate								✓
วัตถุประสงค์ของการวิจัย								
เพื่อทดสอบประสิทธิภาพของโมเดล	✓	✓	✓	✓	✓	✓	✓	✓
เพื่อทดสอบความถูกต้อง	✓	✓	✓	✓	✓	✓	✓	✓
เพื่อเสนอแนวคิดใหม่	✓	✓	✓	✓	✓	✓	✓	✓

ก หมายถึง งานวิจัยของ He He, Ali Ghodsi (2010)

ข หมายถึง งานวิจัยของ Yang Yong (2012)

ค หมายถึง งานวิจัยของ Silvia Cateni, Valentina Colla, Marco Vannucci (2014)

ง หมายถึง งานวิจัยของ Shounak Datta, Swagatam Das (2015)

จ หมายถึง งานวิจัยของ Huaxiang Zhang, Mingfang Li (2014)

ฉ หมายถึง งานวิจัยของ Ming Gaoa, Xia Hong, Sheng Chen, Chris J. Harris, Emad Khalaf (2014)

ช หมายถึง งานวิจัยของ Piyanoot Vorraboot, Suwanna Rasmequan, Krisana Chinnasarn, Chidchanok Lursinsap (2015)

ฌ หมายถึง งานวิจัยของวิทยานิพนธ์ฉบับนี้

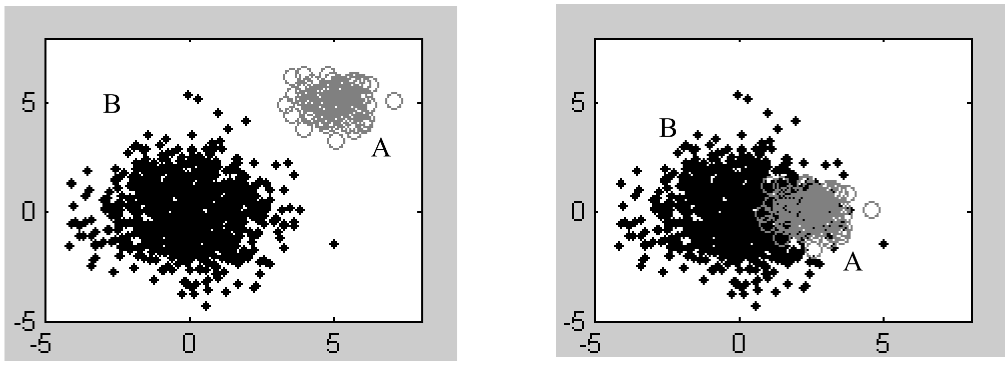
บทที่ 3

วิธีดำเนินการวิจัย

ในงานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาอัลกอริทึมที่ใช้ในการจำแนกประเภทข้อมูลที่มีขนาดของคลาสไม่สมดุล เพื่อเพิ่มประสิทธิภาพในการจำแนกข้อมูลให้ดีขึ้น โดยเฉพาะข้อมูลในคลาสส่วนน้อยที่มักจะจำแนกได้ยาก โดยขอบเขตของการวิจัยจะใช้ข้อมูลประเภทตัวเลข และมีคลาสเป้าหมายจำนวน 2 คลาส ซึ่งในบทนี้จะกล่าวถึงรายละเอียดวิธีการดำเนินการวิจัย และขั้นตอนต่าง ๆ ในงานวิจัย

3.1 กรอบแนวคิดของการวิจัย

แนวคิดของงานวิจัยนี้เกิดจากการสังเกตลักษณะข้อมูลซึ่งโดยทั่วไปแล้วข้อมูลที่มีคลาสเป้าหมาย 2 คลาสจะมีลักษณะของข้อมูลแสดงตัวอย่างดังรูปที่ 3.1 ซึ่งโดยทั่วไปแล้วข้อมูลจะเป็นข้อมูลที่ไม่สมดุล จากรูปที่ 3.1 คลาส A คือข้อมูลวงกลม และคลาส B คือจุดสีดำ ซึ่งจะเห็นได้ว่าจะมีคลาสส่วนน้อย (A) และคลาสส่วนใหญ่ (B) โดยในรูปที่ 3.1 (ก) นั้นถ้าหากข้อมูลไม่มีการซ้อนทับกันของข้อมูลจะทำให้การจำแนกนั้นมีประสิทธิภาพสูง ในทางกลับกันในรูปที่ 3.1 (ข) นั้นมีการซ้อนทับกันของข้อมูลจึงทำให้ประสิทธิภาพในการจำแนกข้อมูลต่ำลง ซึ่งคุณสมบัติของคลาสส่วนใหญ่นั้นจะบดบังคุณสมบัติของคลาสส่วนน้อยทำให้เกิดการจำแนกผิดพลาด ดังนั้นผู้วิจัยจึงได้ออกแบบกรอบแนวคิดของการวิจัยโดยแบ่งข้อมูลออกเป็นส่วนย่อยแสดงดังรูปที่ 3.2



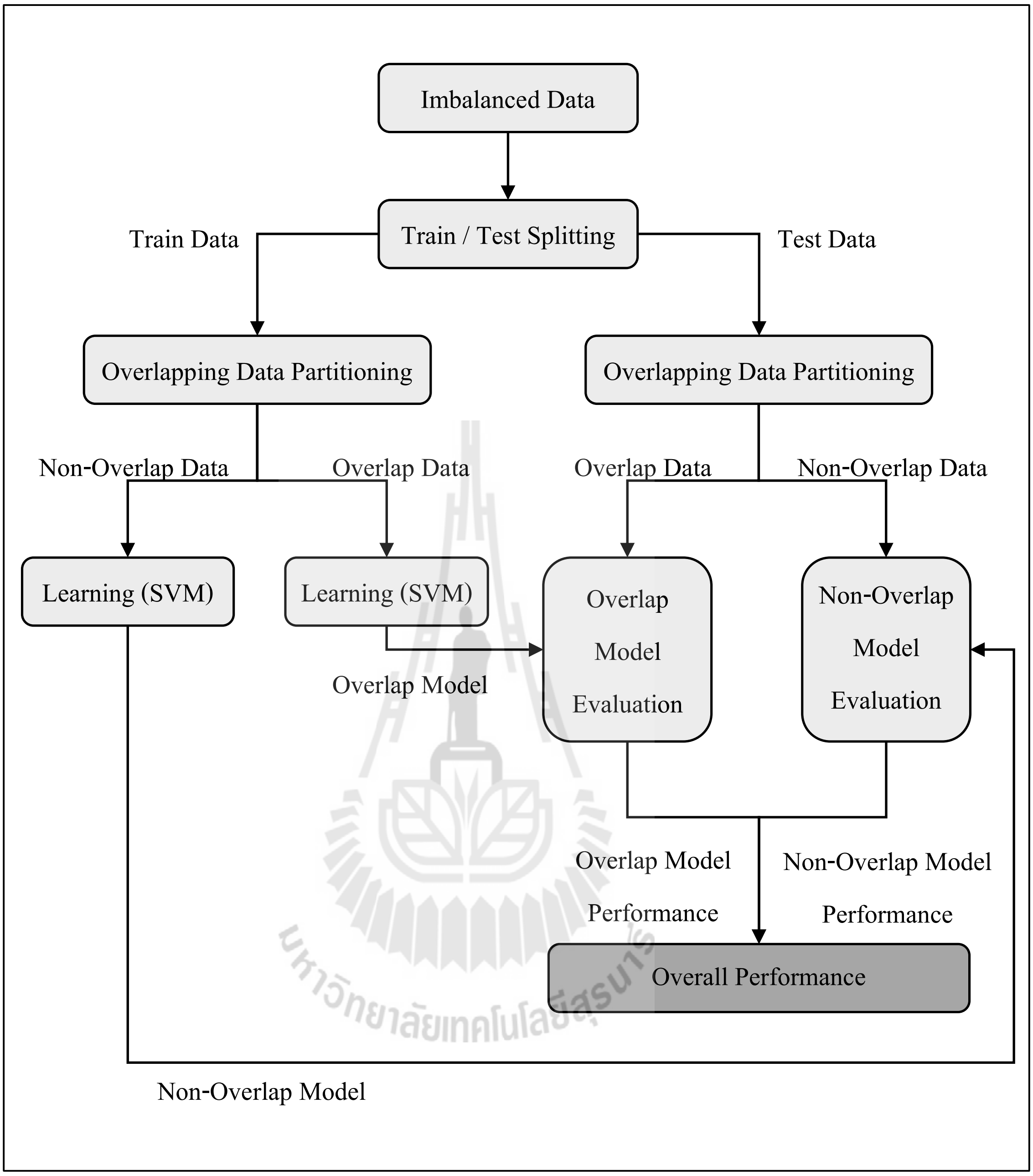
SVM Accuracy (100%)

ก) ข้อมูลไม่สมดุลและคลาสไม่ซ้อนทับ

SVM Accuracy (85.2%)

ข) ข้อมูลไม่สมดุลและคลาสซ้อนทับ

รูปที่ 3.1 แสดงตัวอย่างข้อมูลไม่สมดุล (ก) และ (ข) ที่มีคลาสซ้อนทับกัน



รูปที่ 3.2 กรอบแนวคิดและขั้นตอนในการดำเนินการวิจัย

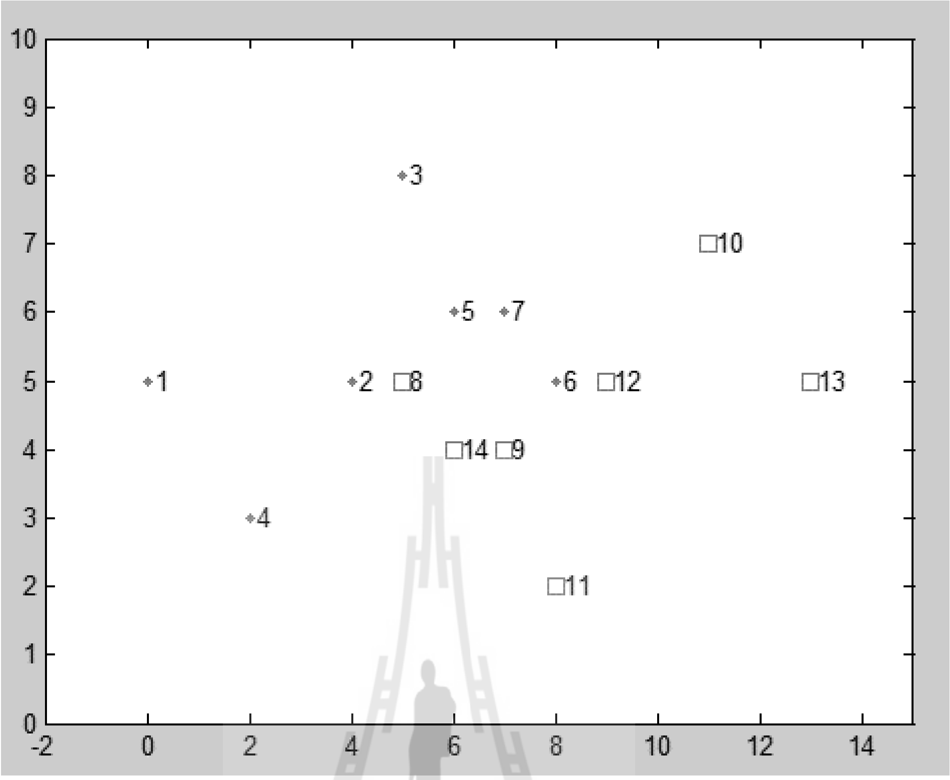
จากรูปที่ 3.2 แสดงรายละเอียดของกรอบแนวคิดและขั้นตอนในการดำเนินการวิจัย ซึ่งในแต่ละขั้นตอนของการดำเนินการวิจัยสามารถอธิบายรายละเอียดขั้นตอนต่าง ๆ ได้ดังต่อไปนี้

3.1.1 ข้อมูลตัวอย่าง (Data)

ข้อมูลที่ใช้ในงานวิจัยจำเป็นต้องเป็นข้อมูลตัวเลข เนื่องจากในขั้นตอนการแบ่งข้อมูลออกเป็น 2 ส่วนคือข้อมูลส่วนที่มีการซ้อนทับ และข้อมูลที่ไม่มีการซ้อนทับกันนั้นมีขั้นตอนการหาระยะห่างระหว่างข้อมูล และข้อมูลที่นำมาใช้จำเป็นที่จะต้องเป็นข้อมูลที่ไม่มีข้อมูลสูญหาย หรือ Missing Value เพราะว่าข้อมูลดังกล่าวจะไม่สามารถหาระยะห่างระหว่างข้อมูลได้จำเป็นต้องทำการเตรียมข้อมูล (Pre-Processing) ให้เรียบร้อยก่อนนำข้อมูลมาใช้งาน โดยในตารางที่ 3.1 แสดงตัวอย่างข้อมูลที่พร้อมใช้งาน ซึ่งประกอบด้วยข้อมูลจำนวน 14 แถวและมี 4 แอตทริบิวต์คือ Id, X, Y และ Class ซึ่งมีคลาสเป้าหมาย 2 คลาสซึ่งอยู่ในแอตทริบิวต์ Class คือคลาส A และ B ในการใช้งานข้อมูลจะไม่นำแอตทริบิต์ id มาใช้งานเนื่องจากเป็นแอตทริบิวต์ในการบอกหมายเลขข้อมูลเท่านั้น จากข้อมูลในตารางที่ 3.1 สามารถแสดงเป็นรูปภาพชุดข้อมูลที่แสดงในรูปที่ 3.3

ตารางที่ 3.1 แสดงข้อมูลตัวอย่าง

Id	X	Y	Class
1	0	5	A
2	4	5	A
3	5	8	A
4	2	3	A
5	6	6	A
6	8	5	A
7	7	6	A
8	5	5	B
9	7	4	B
10	11	7	B
11	8	2	B
12	9	5	B
13	13	5	B
14	6	4	B



รูปที่ 3.3 แสดงข้อมูลตัวอย่าง

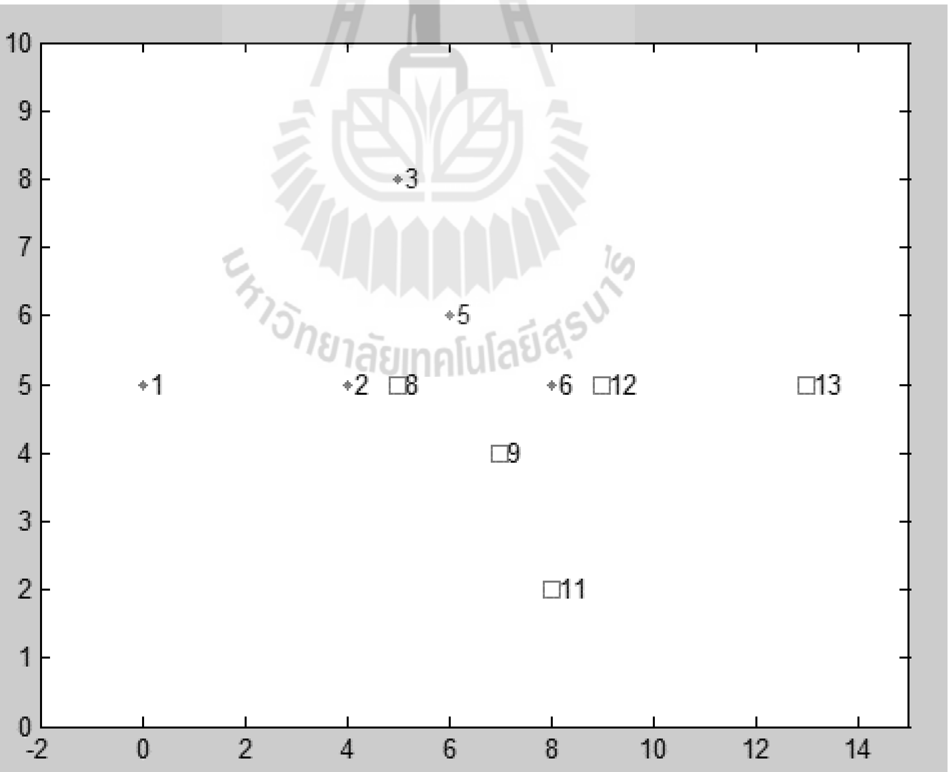
จากรูปที่ 3.3 แสดงข้อมูลตัวอย่างจากตาราง 3.1 โดยมีรายละเอียดคือจุดกลมสีน้ำเงินคือข้อมูลคลาส A และจุดกล่องสี่เหลี่ยมสีแดงคือข้อมูลคลาส B และตัวเลขกำกับในแต่ละจุดคือ Id ของข้อมูลนั้น ๆ

3.1.2 การแยกข้อมูลฝึกสอนและข้อมูลทดสอบ (Train/Test Splitting)

ในส่วนของขั้นตอนการแบ่งข้อมูลนั้นในงานวิจัยนี้จะทำการแบ่งข้อมูลออกเป็น 2 ส่วนโดยวิธีการสุ่ม ซึ่งทั้ง 2 ส่วนที่แบ่งออกมาได้แก่ส่วนของข้อมูลฝึกสอน (Training Set) และส่วนข้อมูลทดสอบ (Test Set) โดยจะใช้อัตราส่วนข้อมูลฝึกสอน 70% และข้อมูลทดสอบ 30% จากจำนวนข้อมูลทั้งหมด เมื่อใช้ข้อมูลตัวอย่างจากตารางที่ 3.1 ในการแบ่งข้อมูลแล้วจะได้ชุดข้อมูลฝึกสอนดังตารางที่ 3.2 และรูปที่ 3.4 ส่วนชุดข้อมูลทดสอบแสดงดังตารางที่ 3.3 และรูปที่ 3.5

ตารางที่ 3.2 แสดงชุดข้อมูลฝึกสอน

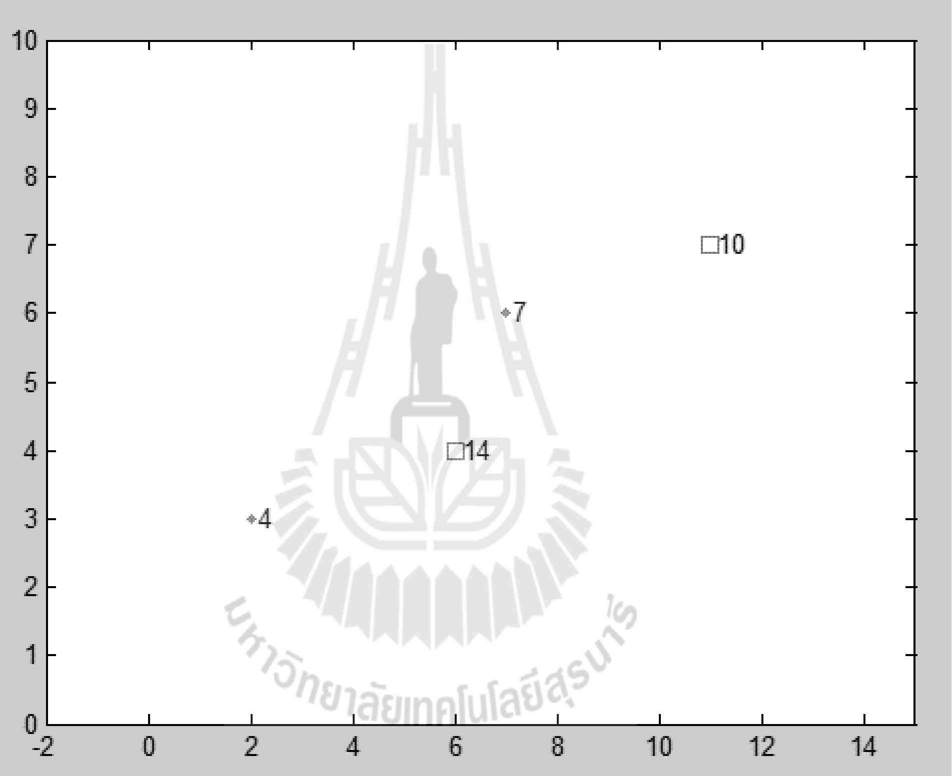
Id	X	Y	Class
1	0	5	A
2	4	5	A
3	5	8	A
5	6	6	A
6	8	5	A
8	5	5	B
9	7	4	B
11	8	2	B
12	9	5	B
13	13	5	B



รูปที่ 3.4 แสดงชุดข้อมูลฝึกสอน

ตารางที่ 3.3 แสดงชุดข้อมูลทดสอบ

Id	X	Y	Class
4	2	3	A
7	7	6	A
10	11	7	B
14	6	4	B



รูปที่ 3.5 แสดงชุดข้อมูลทดสอบ

3.2 การแบ่งข้อมูลที่ซ้อนทับกัน (Overlapping Data Partitioning)

ในขั้นตอนการแบ่งข้อมูลที่ซ้อนทับกันและข้อมูลที่ไม่ซ้อนทับกันนั้นในงานวิจัยนี้ใช้เทคนิคการวัดระยะทางในการแบ่งข้อมูล ซึ่งใช้ 2 เทคนิคในการวัดระยะทางคือ Euclidean Distance และ Hausdorff Distance ซึ่งจะแสดงตัวอย่างการแบ่งข้อมูลทั้ง 2 เทคนิคโดยใช้ชุดข้อมูลฝึกสอนจากตารางที่ 3.2

3.2.1 การแบ่งข้อมูลที่ซ้อนทับกันด้วยเทคนิค Euclidean Distance

จากขั้นตอนวิธีการในรูปที่ 3.6 อันดับแรกทำการหาจุดกึ่งกลางข้อมูลในแต่ละคลาส ซึ่งหาได้โดยการหาค่าเฉลี่ยในแต่ละแอตทริบิวต์ในกรณีที่เป็นคลาส A หาค่าเฉลี่ยแอตทริบิวต์ X คือ (0+4+5+6+8)/5 = 4.6 และค่าเฉลี่ยแอตทริบิวต์ Y คือ (5+5+6+8+5)/5 = 5.8 และคลาส B หาค่าเฉลี่ยแอตทริบิวต์ X คือ (5+7+8+9+13)/5 = 8.4 และค่าเฉลี่ยแอตทริบิวต์ Y คือ (5+4+2+5+5)/5 = 4.2 ซึ่งแสดงรายละเอียดในตารางที่ 3.4

การแบ่งข้อมูลที่ซ้อนทับกันด้วยเทคนิค Euclidean Distance

ข้อมูลเข้า : ข้อมูลฝึกสอน

ผลลัพธ์ : ข้อมูลที่มีการซ้อนทับ และ ข้อมูลที่ไม่มีการซ้อนทับ

วิธีการ :

1. ทำการหาจุดกึ่งกลางข้อมูลในแต่ละคลาสโดยการหาค่าเฉลี่ยในแต่ละแอตทริบิวต์
2. คำนวณหาระยะห่างที่มากที่สุดระหว่างจุดกึ่งกลางในแต่ละคลาส (D_{in}) กับข้อมูลในคลาสนั้น ๆ
3. กำหนดให้ D_{in3} เท่ากับค่าควอร์ไทล์ที่ 3 ของค่า D_{in}
4. คำนวณระระห่างระหว่างจุดกึ่งกลางคลาส กับข้อมูลคลาสอื่น (D_{out}) ถ้าหากค่าระยะห่าง D_{out} มีค่าน้อยกว่าหรือเท่ากับ D_{in3} จะถือว่าข้อมูลนั้นเป็นข้อมูลที่มีการซ้อนทับ แต่ถ้าหากว่า D_{out} มีค่ามากกว่า D_{in3} จะถือว่าเป็นข้อมูลที่ไม่มีการซ้อนทับ

รูปที่ 3.6 ขั้นตอนการแบ่งข้อมูลที่ซ้อนทับกัน

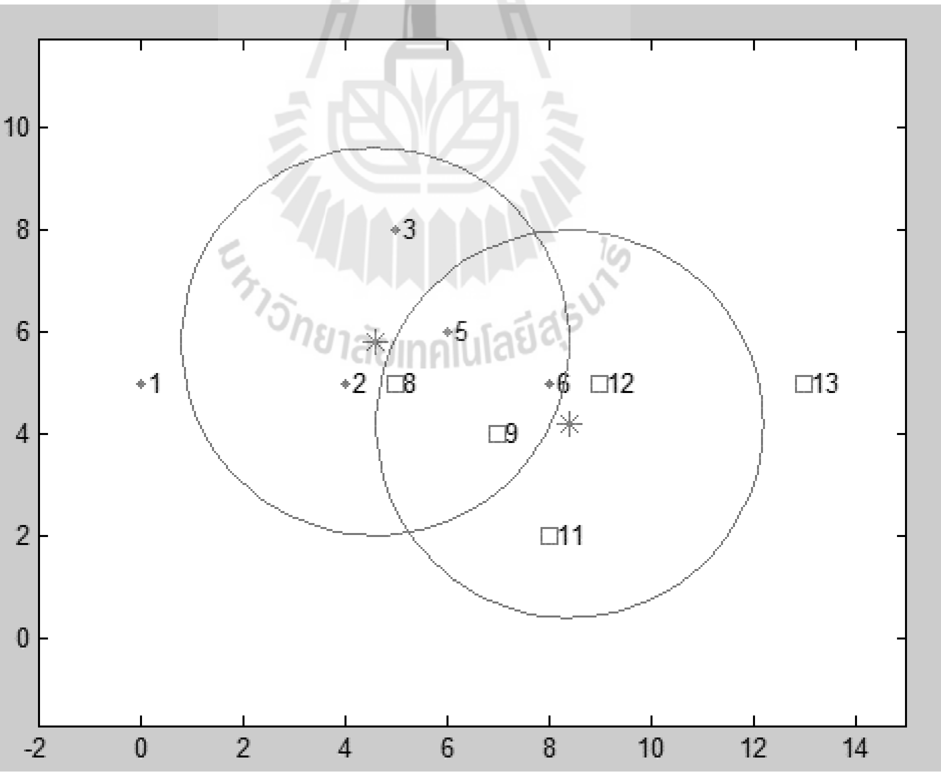
ตารางที่ 3.4 แสดงจุดศูนย์กลางของชุดข้อมูลฝึกสอน

X`	Y`	Class
4.6	5.8	A
8.4	4.2	B

เมื่อได้จุดกึ่งกลางข้อมูลเรียบร้อยแล้วขั้นตอนต่อไปเป็นการคำนวณระยะทางด้วยเทคนิค Euclidean Distance ซึ่งเมื่อคำนวณแล้วจะได้ดังตารางที่ 3.5 ซึ่งค่า D_{in} คือระยะห่างระหว่างข้อมูลในคลาสกับจุดกึ่งกลางคลาส ส่วน D_{out} คือระยะห่างระหว่างข้อมูลในคลาสกับจุดกึ่งกลางคลาสอื่น จากข้อมูลในตารางที่ 3.5 ค่า D_{in3} มีค่าเท่ากับ 3.79 ซึ่งสามารถแบ่งข้อมูลได้ดังรูปที่ 3.7 โดยในตารางที่ 3.6 จะแสดงข้อมูลที่ซ้อนทับกัน และตารางที่ 3.7 จะแสดงข้อมูลที่ไม่ซ้อนทับกัน

ตารางที่ 3.5 แสดงการหาระยะห่างด้วยเทคนิค Euclidean Distance ด้วยข้อมูลฝึกสอน

Id	X	Y	X\`	Y\`	D_{in}	D_{out}	Class
1	0	5	4.6	5.8	4.67	8.44	A
2	4	5	4.6	5.8	1.00	4.47	A
3	5	8	4.6	5.8	2.24	5.10	A
5	6	6	4.6	5.8	1.41	**3.00**	A
6	8	5	4.6	5.8	3.49	**0.89**	A
8	5	5	8.4	4.2	3.49	**0.89**	B
9	7	4	8.4	4.2	1.41	**3.00**	B
11	8	2	8.4	4.2	2.24	5.10	B
12	9	5	8.4	4.2	1.00	4.47	B
13	13	5	8.4	4.2	4.67	8.44	B



รูปที่ 3.7 แสดงการหาข้อมูลที่ซ้อนทับและไม่ซ้อนทับกันด้วยข้อมูลฝึกสอน

ตารางที่ 3.6 แสดงข้อมูลฝึกสอนที่ซ้อนทับกัน

Id	X	Y	Class
5	6	6	A
6	8	5	A
8	5	5	B
9	7	4	B

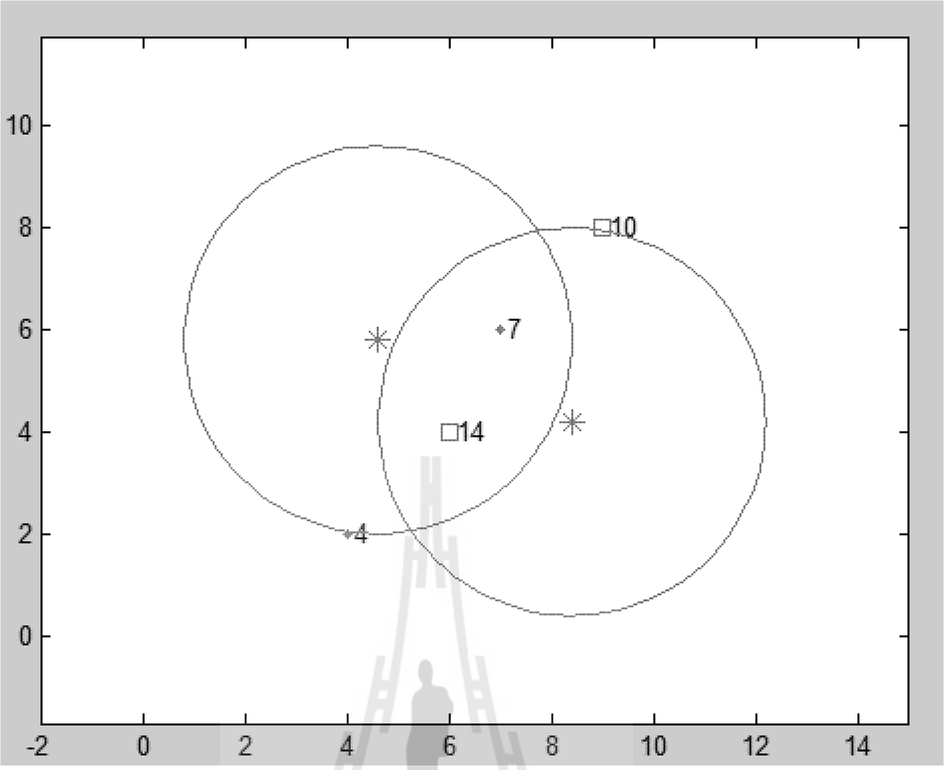
ตารางที่ 3.7 แสดงข้อมูลฝึกสอนที่ไม่ซ้อนทับกัน

Id	X	Y	Class
1	0	5	A
2	4	5	A
3	5	8	A
11	8	2	B
12	9	5	B
13	13	5	B

หลังจากที่แบ่งการซ้อนทับกันของข้อมูลด้วยชุดข้อมูลฝึกสอนเรียบร้อยแล้ว นำค่า D_{in3} ซึ่งมีค่าเท่ากับ 3.79 และนำค่า X`, Y` มาใช้ในการแบ่งข้อมูลที่ซ้อนทับกันของข้อมูลทดสอบแสดงดังตารางที่ 3.8 และรูปที่ 3.8 ส่วนในตารางที่ 3.9 แสดงชุดข้อมูลทดสอบที่ซ้อนทับกัน และในตารางที่ 3.10 แสดงชุดข้อมูลทดสอบที่ไม่ซ้อนทับกัน

ตารางที่ 3.8 แสดงการหาระยะห่างด้วยเทคนิค Euclidean Distance ด้วยข้อมูลทดสอบ

Id	X	Y	X`	Y`	D_{in3}	D_{out}	Class
4	2	3	4.6	5.8	3.79	6.51	A
7	7	6	4.6	5.8	3.79	**2.28**	A
10	11	7	8.4	4.2	3.79	6.51	B
14	6	4	8.4	4.2	3.79	**2.28**	B



รูปที่ 3.8 แสดงการหาข้อมูลที่ซ้อนทับและไม่ซ้อนทับกันด้วยข้อมูลทดสอบ

ตารางที่ 3.9 ชุดข้อมูลทดสอบที่มีการซ้อนทับกัน

Id	X	Y	Class
7	7	6	A
14	6	4	B

ตารางที่ 3.10 ชุดข้อมูลทดสอบที่ไม่มีการซ้อนทับกัน

Id	X	Y	Class
4	2	3	A
10	11	7	B

3.2.2 การแบ่งข้อมูลที่ซ้อนทับกันด้วยเทคนิค Hausdorff Distance

ในการแบ่งข้อมูลที่ซ้อนทับด้วยเทคนิค Hausdorff Distance ใช้ชุดข้อมูลฝึกสอนจากตารางที่ 3.2 และชุดข้อมูลทดสอบในตาราง 3.3 ซึ่งการแบ่งข้อมูลที่ซ้อนทับกันด้วยเทคนิค Hausdorff Distance สามารถอธิบายขั้นตอนต่าง ๆ ได้ดังรูปที่ 3.9

การแบ่งข้อมูลที่ซ้อนทับกันด้วยเทคนิค Hausdorff Distance

ข้อมูลเข้า : ข้อมูลฝึกสอน

ผลลัพธ์ : ข้อมูลที่มีการซ้อนทับ และ ข้อมูลที่ไม่มีการซ้อนทับ

วิธีการ :

1. หาระยะห่างระหว่างข้อมูลทุกข้อมูลในคลาสที่หนึ่ง (P) กับข้อมูลทุกข้อมูลในคลาสที่สอง (N)
2. หาระยะห่างที่น้อยที่สุดในแต่ละข้อมูลของคลาส P ทุกข้อมูลในคลาส N โดยกำหนดให้เป็นค่า P_{min}
3. หาค่าที่มากที่สุดของค่า P_{min} โดยกำหนดให้เป็นค่า dP_{max} ซึ่งจะได้จุดข้อมูลในคลาส P ที่มีระยะห่างที่มากที่สุดกับข้อมูลคลาส N ซึ่งกำหนดให้ข้อมูลนั้นคือ P_{point}
4. หาระยะห่างที่น้อยที่สุดในแต่ละข้อมูลของคลาส N กับทุกข้อมูลในคลาส P โดยกำหนดให้เป็นค่า N_{min}
5. หาค่าที่มากที่สุดของค่า N_{min} โดยกำหนดให้เป็นค่า dN_{max} ซึ่งจะได้จุดข้อมูลในคลาส N ที่มีระยะห่างที่มากที่สุดกับข้อมูลคลาส P ซึ่งกำหนดให้ข้อมูลนั้นคือ N_{point}
6. หาระยะห่างระหว่างข้อมูลทุกข้อมูลกับ P_{point} ถ้าหากมีค่าน้อยกว่าค่า dP_{max} จะถือว่าเป็นข้อมูลที่ไม่ซ้อนทับ ถ้าหากมีค่ามากกว่าหรือเท่ากับค่า dP_{max} จะถือว่าเป็นข้อมูลซ้อนทับ
7. หาระยะห่างระหว่างข้อมูลทุกข้อมูลกับ N_{point} ถ้าหากมีค่าน้อยกว่าค่า dN_{max} จะถือว่าเป็นข้อมูลที่ไม่ซ้อนทับ ถ้าหากมีค่ามากกว่าหรือเท่ากับค่า dN_{max} จะถือว่าเป็นข้อมูลซ้อนทับ

รูปที่ 3.9 แสดงการแบ่งข้อมูลที่ซ้อนทับกันด้วยเทคนิค Hausdorff Distance

อันดับแรกทำการหาระยะห่างระหว่างข้อมูลคลาสแรก (P) และข้อมูลคลาสที่สอง (N) ซึ่งจะเป็นการคำนวณระยะห่างของข้อมูล P ทีละข้อมูลเพื่อหาระยะห่างกับข้อมูลทุกข้อมูลในคลาส N โดยทำเช่นเดียวกันนี้ให้ครบทุกข้อมูลในคลาส P ซึ่งจะได้ระยะห่างระหว่างข้อมูลทุกข้อมูลของทั้งสองคลาสซึ่งแสดงได้ดังตารางที่ 3.11

ตารางที่ 3.11 แสดงระยะห่างระหว่างข้อมูลระหว่างคลาส A และ B

B(Id) A(Id)	8	9	11	12	13
1	5.00	7.07	8.54	9.00	13.00
2	1.00	3.16	5.00	5.00	9.00
3	3.00	4.47	6.71	5.00	8.54
5	1.41	2.24	4.47	3.16	7.07
6	3.00	1.41	3.00	1.00	5.00

เมื่อได้ระยะห่างระหว่างข้อมูลของทั้งสองคลาสเรียบร้อยแล้วต่อไปเป็นการหาค่า P_{min} ซึ่งสามารถหาได้จากตารางที่ 3.11 โดยสามารถแสดงได้ดังตารางที่ 3.12 ซึ่งมีทั้งหมด 6 ค่าแล้วทำการหาค่า P_{max} ซึ่งเป็นค่าที่มากที่สุดของค่า P_{min} ซึ่งในที่นี้คือ 5.00 โดยมีข้อมูลที่มี id เท่ากับ 1 เป็น P_{point}

ต่อไปเป็นการหาค่า N_{min} ซึ่งทำเช่นเดียวกันกับการหาค่า P_{min} เพียงแต่เปลี่ยนคลาสเท่านั้น ซึ่งแสดงได้ดังตารางที่ 3.13 ซึ่งมีทั้งหมด 6 ค่าแล้วทำการหาค่า N_{max} ซึ่งมีค่าเท่ากับ 5.00 โดยมีข้อมูลที่มี Id เท่ากับ 13 เป็น N_{point}

ตารางที่ 3.12 แสดงค่า P_{min}

B(Id) A(Id)	8	9	11	12	13
1	**5.00**	7.07	8.54	9.00	13.00
2	**1.00**	3.16	5.00	5.00	9.00
3	**3.00**	4.47	6.71	5.00	8.54
5	**1.41**	2.24	4.47	3.16	7.07
6	3.00	**1.41**	3.00	1.00	5.00

ตารางที่ 3.13 แสดงค่า N_{min}

B(Id) A(Id)	8	9	11	12	13
1	5.00	7.07	8.54	9.00	13.00
2	**1.00**	3.16	5.00	5.00	9.00
3	3.00	4.47	6.71	5.00	8.54
5	1.41	2.24	4.47	3.16	7.07
6	3.00	**1.41**	**3.00**	**1.00**	**5.00**

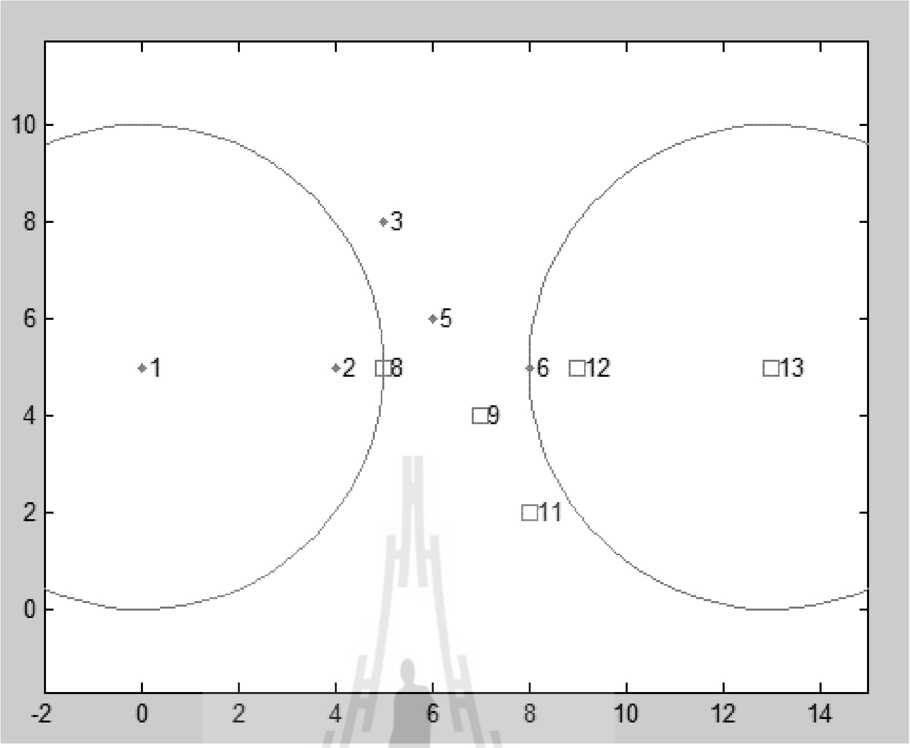
เมื่อทำการหาค่าที่ต้องการได้ครบแล้วต่อไปเป็นการหาข้อมูลที่ซ้อนทับและไม่ซ้อนทับ โดยการหาระยะทางของข้อมูลทั้งหมดกับข้อมูล P_{point} และข้อมูล N_{point} แล้วทำการหาค่าที่น้อยกว่าค่า dP_{max} และ dN_{max} ซึ่งมีค่าเท่ากับ 5.00 แสดงในตารางที่ 3.14 แสดงข้อมูลซ้อนทับและไม่ซ้อนทับได้ดังรูปที่ 3.9

ตารางที่ 3.14 แสดงข้อมูลที่ซ้อนทับและข้อมูลที่ไม่ซ้อนทับ

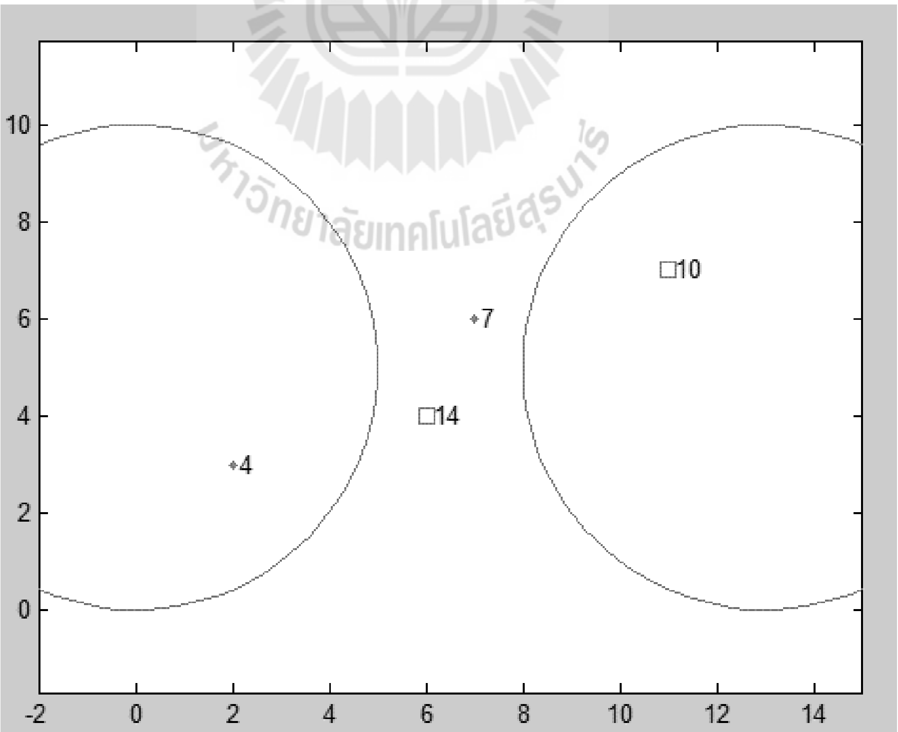
Id	1	2	3	5	6	8	9	11	12	13
1	**0.00**	**4.00**	5.83	6.08	8.00	5.00	7.07	8.54	9.00	13.00
13	13.00	9.00	8.54	7.07	5.00	8.00	6.08	5.83	**4.00**	**0.00**

จากรูปที่ 3.10 ซึ่งแสดงข้อมูลที่ซ้อนทับและไม่ซ้อนทับกันโดยข้อมูลที่ซ้อนทับกันนั้นเป็นข้อมูลที่อยู่นอกวงกลมทั้งสองซึ่งได้แก่ข้อมูล Id = 3,5,6,8,9,11 และข้อมูลที่ไม่ซ้อนทับกันได้แก่ข้อมูล Id = 1,2,12,13

จากรูปที่ 3.11 แสดงข้อมูลที่ซ้อนทับและไม่ซ้อนทับบนข้อมูลทดสอบ ปรากฏว่ามีข้อมูลที่ไม่ซ้อนทับกัน 2 ข้อมูลได้แก่ข้อมูล Id = 4,10 และข้อมูลที่ซ้อนทับ 2 ข้อมูลได้แก่ข้อมูล Id = 7,14



รูปที่ 3.10 แสดงข้อมูลซ้อนทับและไม่ซ้อนทับกันบนข้อมูลฝึกสอน



รูปที่ 3.11 แสดงข้อมูลซ้อนทับและไม่ซ้อนทับกันบนข้อมูลทดสอบ

3.3 มาตรวัดระยะทาง

ในส่วนนี้จะเป็นส่วนที่อธิบายถึงมาตรวัดระยะทางที่ใช้ในงานวิจัยฉบับนี้ ซึ่งในงานวิจัยนี้ใช้มาตรวัดระยะทาง 3 มาตรวัดได้แก่ Euclidean, City Block และ Mahalanobis

3.3.1 มาตรวัดระยะทางแบบ Euclidean

มาตรวัดระยะทางแบบ Euclidean เป็นมาตรวัดระยะทางที่วัดระยะทางระหว่างจุด 2 จุดโดยระยะทางที่ได้จะเป็นระยะทางที่สั้นที่สุด โดยการคำนวนแสดงดังสมการ (3-1)

$$D1(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3-1}$$

โดยที่ D1 คือระยะทางที่ได้จากมาตรวัดระยะทางแบบ Euclidean

x คือ เวกเตอร์ของจุดที่ 1 ($x = (x_1, x_2 \ldots, x_n)$)

y คือ เวกเตอร์ของจุดที่ 2 ($y = (y_1, y_2 \ldots, y_n)$)

n คือ จำนวนมิติของข้อมูล

3.3.2 มาตรวัดระยะทางแบบ City Block

มาตรวัดระยะทางแบบ City Block เป็นมาตรวัดระยะทางที่วัดระยะทางระหว่าง 2 จุด โดยระยะทางที่ได้จะเป็นระยะทางที่ได้จากการจำลองการเดินทางในเมืองจากจุดที่ 1 ไปจุดที่ 2 ว่าต้องใช้ระยะทางทั้งหมดกี่ Block ซึ่งสามารถหาระยะทางได้จากสมการ (3-2)

$$D2(x,y) = \sum_{i=1}^{n}|x_i - y_i| \tag{3-2}$$

โดยที่ D2 คือระยะทางที่ได้จากมาตรวัดระยะทางแบบ City Block

x คือ เวกเตอร์ของจุดที่ 1 ($x = (x_1, x_2 \ldots, x_n)$)

y คือ เวกเตอร์ของจุดที่ 2 ($y = (y_1, y_2 \ldots, y_n)$)

n คือ จำนวนมิติของข้อมูล

3.3.3 มาตรวัดระยะทางแบบ Mahalanobis

มาตรวัดระยะทางแบบ Mahalanobis เป็นมาตรวัดระยะทางที่วัดระยะทางระหว่าง 2 จุด โดยระยะทางที่ได้จะเป็นระยะทางรวมที่ได้จากการเฉลี่ยระยะทางในแต่ละมิติของข้อมูล สามารถคำนวนได้จากสมการ (3-3)

$$D3(x,y) = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{S(x_i - y_i)}} \qquad (3\text{-}3)$$

โดยที่ D3 คือระยะทางที่ได้จากมาตรวัดระยะทางแบบ Mahalanobis

x คือ เวกเตอร์ของจุดที่ 1 ($x = (x_1, x_2 \ldots, x_n)$)

y คือ เวกเตอร์ของจุดที่ 2 ($y = (y_1, y_2 \ldots, y_n)$)

n คือ จำนวนมิติของข้อมูล

S คือ Covariance matrix

3.4 การเรียนรู้โมเดลและการทดสอบประสิทธิภาพโมเดล

ในขั้นตอนนี้จะเป็นการอธิบายเกี่ยวกับการเรียนรู้โมเดลและการทดสอบประสิทธิภาพของโมเดลว่าโมเดลที่สร้างขึ้นนั้นมีประสิทธิภาพในการจำแนกประเภทข้อมูลเท่าใด

3.4.1 การเรียนรู้โมเดลในการจำแนก (Learning)

สำหรับขั้นตอนนี้จะเป็นการใช้อัลกอริทึมสำหรับจำแนกประเภทข้อมูลซึ่งในงานวิจัยนี้ได้เลือกใช้อัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนในการสร้างโมเดลในการจำแนกประเภทข้อมูลซึ่งในขั้นตอนนี้จะได้โมเดลในการจำแนกข้อมูล 2 โมเดลด้วยกันโดยเกิดจากการใช้ชุดข้อมูลฝึกสอนที่มีการซ้อนทับกันของข้อมูล และชุดข้อมูลฝึกสอนที่ไม่มีการซ้อนทับกันของข้อมูล

3.4.2 การทดสอบประสิทธิภาพของโมเดล (Model Evaluation)

ในขั้นตอนนี้จะเป็นการทดสอบประสิทธิภาพของโมเดลในการจำแนกประเภทข้อมูล 2 โมเดลที่ได้จากชุดข้อมูลฝึกสอน 2 ชุดคือชุดข้อมูลฝึกสอนที่มีการซ้อนทับกันของข้อมูลและชุดข้อมูลที่ไม่มีการซ้อนทับกันของข้อมูล ซึ่งการทดสอบประสิทธิภาพก็ใช้ชุดข้อมูล 2 ชุดข้อมูลในการทดสอบประสิทธิภาพในการจำแนกเช่นเดียวกัน คือชุดทดสอบที่มีการซ้อนทับกันของข้อมูล และชุดทดสอบที่ไม่มีการซ้อนทับกันของข้อมูล ซึ่งจะวัดประสิทธิภาพโดยการใช้ความแม่นยำในการจำแนก (Accuracy) เป็นตัวชี้วัดประสิทธิภาพ

จากตัวอย่างข้อมูลที่ใช้อธิบายในช่วงต้นของบทที่ 3 เมื่อนำข้อมูลฝึกสอนและข้อมูลทดสอบทำการวัดประสิทธิภาพในการจำแนกข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีนจะได้ประสิทธิภาพแสดงดังรูปที่ 3.12 จะเห็นได้ว่าถ้าหากทำการสร้างโมเดลด้วยข้อมูลฝึกสอนโดยไม่มีการแบ่งข้อมูลซ้อนทับและไม่ซ้อนทับกันนั้นจะทำให้ได้ประสิทธิภาพในการจำแนกที่ต่ำกว่าการแบ่งข้อมูลซ้อนทับและไม่ซ้อนทับออกจากกันเพื่อสร้างโมเดลในการจำแนกข้อมูล 2 โมเดล

SVM (Original)

Predict Actual	A	B
A	1	1
B	1	1
Accuracy	(2/4) = 50%	

ก) ไม่มีการแบ่งข้อมูล

SVM (Euclidean)

	Overlap		Non-Overlap	
Predict Actual	A	B	A	B
A	1	0	1	0
B	0	1	0	1
Accuracy	100%		100%	
Total	(4/4) = 100%			

ข) แบ่งข้อมูลด้วยวิธี Euclidean

SVM (Hausdorff)

	Overlap		Non-Overlap	
Predict Actual	A	B	A	B
A	1	0	1	0
B	0	1	0	1
Accuracy	100%		100%	
Total	(4/4) = 100%			

ข) แบ่งข้อมูลด้วยวิธี Hausdorff

รูปที่ 3.12 แสดงประสิทธิภาพการจำแนกของข้อมูลตัวอย่าง

3.4.3 การพยากรณ์ข้อมูลในอนาคต

สำหรับการจำแนกประเภทข้อมูลในอนาคตนั้น จะต้องมีโมเดลในการจำแนกประเภทของข้อมูลประเภทนั้น ๆ ซึ่งได้จากการใช้ข้อมูลประเภทดังกล่าวในการฝึกสอนเพื่อที่จะได้โมเดลในการจำแนกประเภทข้อมูล หลังจากนั้น จึงนำข้อมูลในอนาคตที่เป็นข้อมูลประเภทเดียวกันมาใช้กับโมเดลที่ได้สร้างไว้แล้ว ซึ่งเทคนิคการจำแนกประเภทข้อมูลในงานวิจัยนี้จะทำการแบ่งข้อมูลออกเป็น 2 ส่วนคือ ข้อมูลที่ซ้อนทับกัน และข้อมูลที่ไม่ซ้อนทับกัน เพื่อใช้กับโมเดลจำแนกประเภทข้อมูล 2 โมเดลได้แก่ โมเดลสำหรับข้อมูลที่ซ้อนทับกัน และโมเดลสำหรับข้อมูลที่ไม่ซ้อนทับกัน จากนั้นจึงนำประสิทธิภาพในการจำแนกข้อมูลทั้ง 2 ส่วนมารวมกันจึงเป็นประสิทธิภาพรวมในการจำแนกประเภทข้อมูลในอนาคต

3.5 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการพัฒนางานวิจัยนี้ ประกอบด้วย

1) เครื่องคอมพิวเตอร์สำหรับพัฒนา มีรายละเอียดดังนี้

- หน่วยประมวลผลกลาง : AMD FX 8320
- หน่วยความจำสำรอง : 120GB
- หน่วยความจำหลัก : 8GB

2) ระบบปฏิบัติการและโปรแกรมประยุกต์สำหรับการพัฒนา ประกอบด้วย

- ระบบปฏิบัติการ : Windows 7 Ultimate Service Pack 1 64bits
- เครื่องมือที่ใช้ในการพัฒนา : Matlab2013a

บทที่ 4

การทดสอบและอภิปรายผล

เนื้อหาในส่วนของบทที่ 4 นี้จะเป็นการทดสอบและอภิปรายผลของการใช้เทคนิคการจำแนกประเภทข้อมูลส่วนน้อยบนข้อมูลที่ไม่สมดุลด้วยวิธีการแบ่งข้อมูล ซึ่งในการแบ่งข้อมูลนั้นจะทำการแบ่งข้อมูลออกเป็น 2 ส่วนคือ ส่วนที่ข้อมูลซ้อนทับกัน และส่วนที่ข้อมูลไม่ซ้อนทับกัน เพื่อใช้ข้อมูลทั้งสองส่วนในการสร้างโมเดลในการจำแนกข้อมูลโดยจะได้โมเดลในการจำแนกข้อมูล 2 โมเดลเช่นเดียวกัน สำหรับเนื้อหาในบทนี้จะประกอบด้วยการเตรียมข้อมูลสำหรับการทดสอบ การออกแบบวิธีการทดสอบ ผลการทดสอบประสิทธิภาพ และอภิปรายผล

4.1 การเตรียมข้อมูลสำหรับการทดสอบ

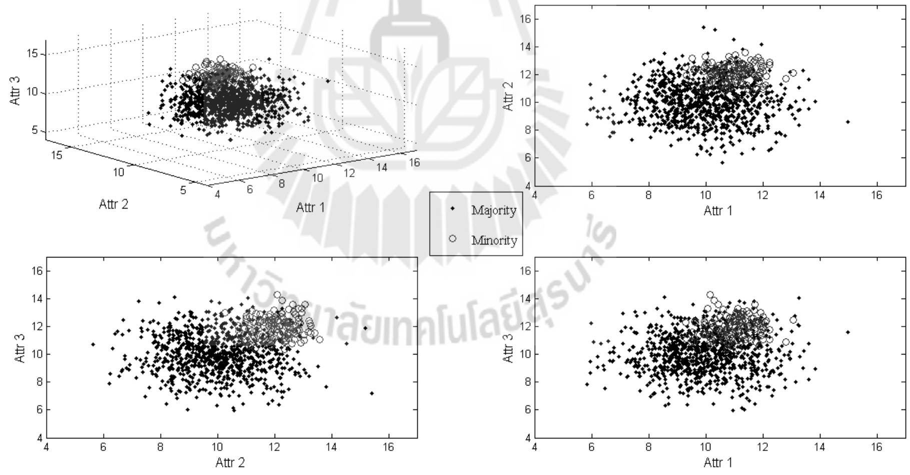
สำหรับข้อมูลที่ใช้ในงานวิจัยของวิทยานิพนธ์ฉบับนี้เพื่อวัดความแม่นในการจำแนกข้อมูลส่วนน้อยจากข้อมูลส่วนใหญ่ จะใช้ข้อมูลที่มี 2 คลาส และข้อมูลทุกชุดข้อมูลที่ใช้จะเป็นข้อมูลตัวเลขทั้งหมด ซึ่งชุดข้อมูลที่ใช้ในงานวิทยานิพนธ์นี้จะใช้ข้อมูล 2 ชนิดได้แก่ ชุดข้อมูลสังเคราะห์ และชุดข้อมูลจริงจากแหล่งข้อมูลมาตรฐาน โดยมีรายละเอียดดังต่อไปนี้

4.1.1 ชุดข้อมูลสังเคราะห์จากโปรแกรม

ข้อมูลสังเคราะห์ที่ใช้ในงานวิทยานิพนธ์ฉบับนี้จะเป็นข้อมูลที่ได้จากการสังเคราะห์ด้วยโปรกรม MATLAB2013b ซึ่งจะสังเคราะห์ชุดข้อมูลตัวเลข โดยมีคุณลักษณะ (Attributes) 3 คุณลักษณะและมีคลาสเป้าหมาย 2 คลาสเท่านั้น ซึ่งจะมีการกำหนดกระจายตัวของข้อมูล การซ้อนทับของข้อมูล และจำนวนข้อมูลในแต่ละคลาสเป้าหมาย โดยรายละเอียดของข้อมูลสังเคราะห์แสดงดังตารางที่ 4.1 และลักษณะการกระจายตัวของข้อมูลแสดงดังรูปที่ 4.1

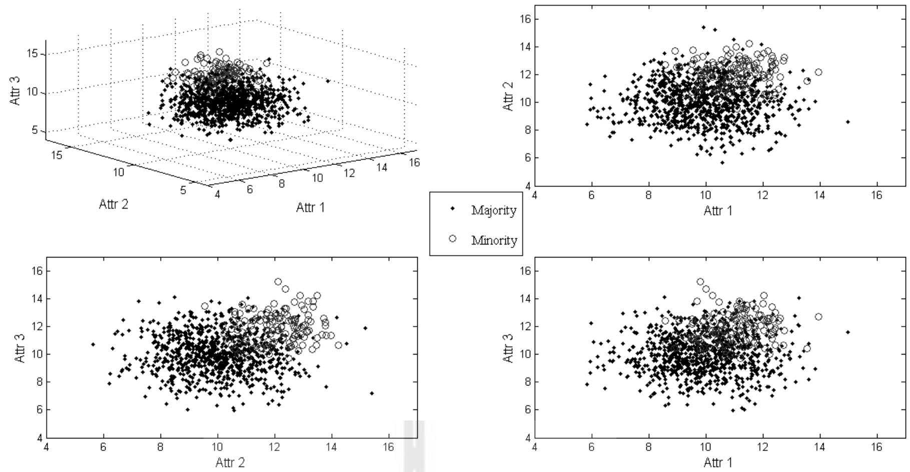
ตารางที่ 4.1 แสดงคุณลักษณะของชุดข้อมูลสังเคราะห์จากโปรแกรม

ชุดข้อมูล	ค่า Mean	ค่าความแปรปรวน		จำนวนตัวอย่าง			IR	OR
		Majority	Minority	Majority	Minority	Total		
D1	[10 10 10; 11 12 12]	[2 0 0; 0 2 0; 0 0 2]	[0.5 0 0; 0 0.5 0; 0 0 0.5]	868	132	1000	6.58	39.99
D2	[10 10 10; 11 12 12]	[2 0 0; 0 2 0; 0 0 2]	[1 0 0; 0 1 0; 0 0 1]	868	132	1000	6.58	51.32
D3	[10 10 10; 11 12 12]	[2 0 0; 0 2 0; 0 0 2]	[2 0 0; 0 2 0; 0 0 2]	868	132	1000	6.58	65.97

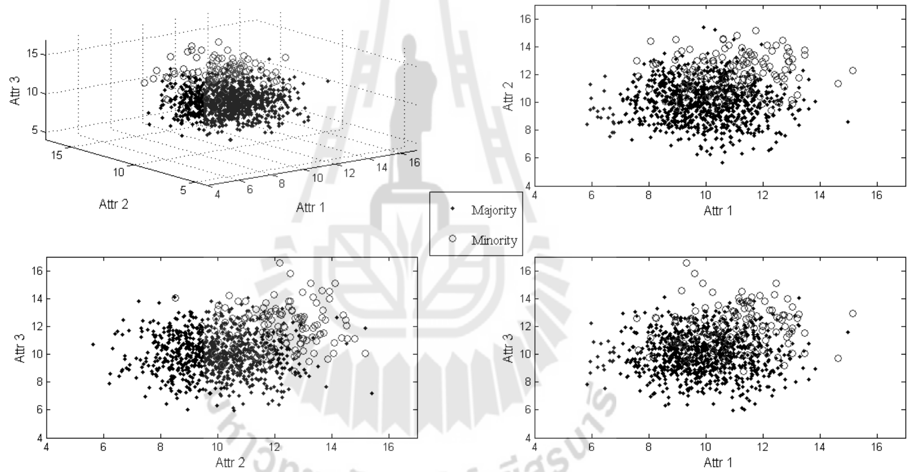


ก) ชุดข้อมูล D1

รูปที่ 4.1 แสดงคุณลักษณะของชุดข้อมูลสังเคราะห์จากโปรแกรม D1 ถึง D3



ข) ชุดข้อมูล D2



ค) ชุดข้อมูล D3

รูปที่ 4.1 แสดงคุณลักษณะของชุดข้อมูงสังเคราะห์จากโปรแกรม D1 ถึง D3 (ต่อ)

4.1.2 ชุดข้อมูลจากแหล่งข้อมูลมาตรฐาน

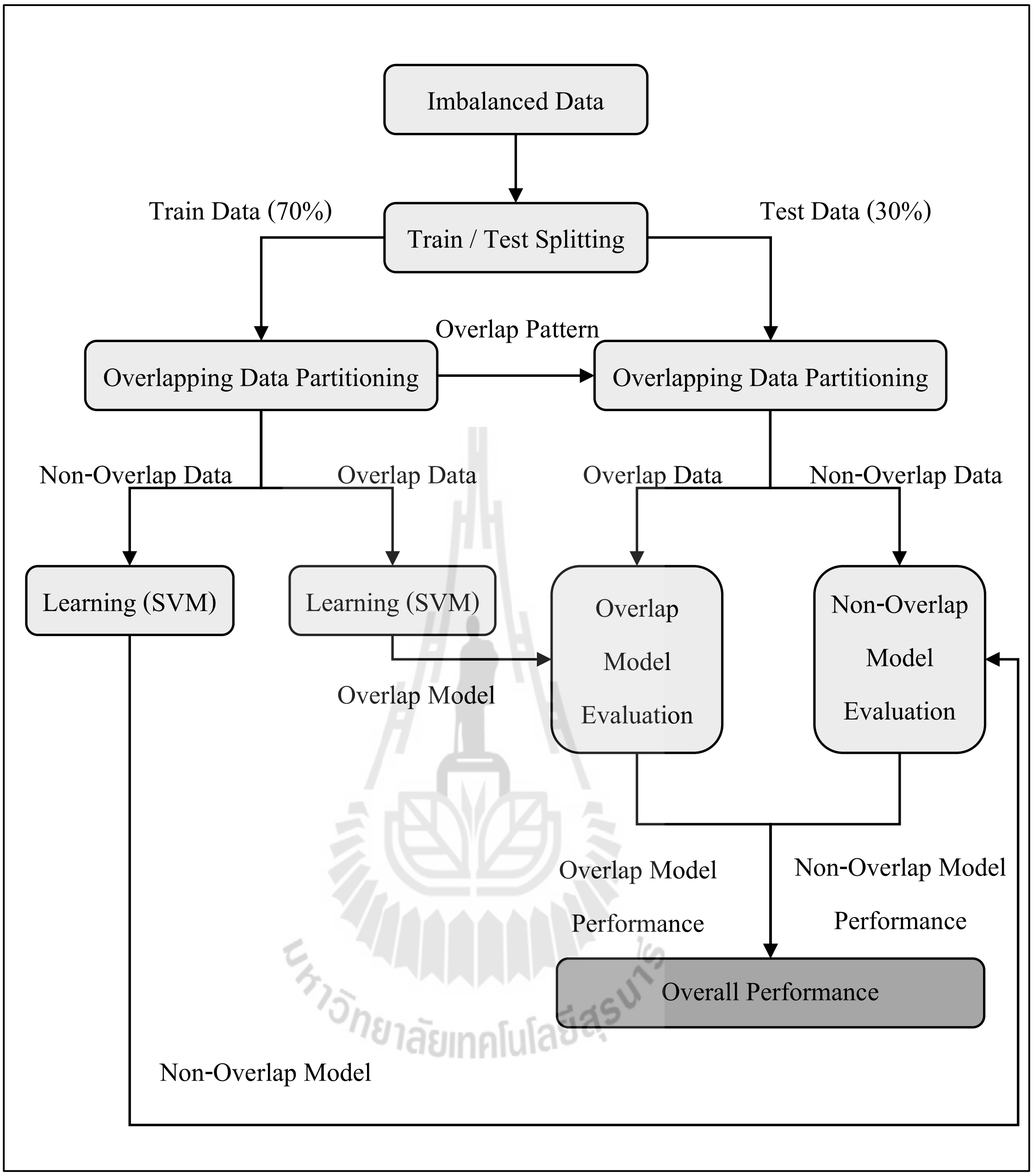
ข้อมูลจริงที่ใช้ในงานวิทยานิพนธ์ฉบับนี้เป็นข้อมูลจริงที่ได้มาจากแหล่งข้อมูลมาตรฐาน Keel Repository (http://www.keel.es/datasets.php) ซึ่งได้นำข้อมูลมาใช้ทั้งหมดจำนวน 7 ชุดข้อมูล โดยมีรายละเอียดแต่ละชุดข้อมูลแสดงดังตารางที่ 4.2

ตารางที่ 4.2 แสดงคุณลักษณะของชุดข้อมูลจริงจากฐานข้อมูลมาตรฐาน

ชุดข้อมูล	Attributes	จำนวนตัวอย่าง			IR	OR
		Majority	Minority	Total		
Haberman	3	225	81	306	2.78	89.86
Liver	6	200	145	345	1.38	65.27
Pima	8	500	268	768	1.87	85.03
Page-Blocks	10	444	28	472	15.86	49.65
Vehicle1	18	629	217	846	2.90	69.96
Vehicle3	18	634	212	846	2.99	71.69
German	24	700	300	1000	2.33	98.45

4.2 การออกแบบวิธีทดสอบ

ในการออกแบบวิธีทดสอบสำหรับเทคนิคการจำแนกประเภทข้อมูลส่วนน้อยบนข้อมูลที่ไม่สมดุลด้วยวิธีการแบ่งข้อมูลนั้นสามารถทำได้โดยนำข้อมูลไม่สมดุลแบ่ง 70% เป็นชุดข้อมูลฝึกสอนและ 30% เป็นข้อมูลทดสอบ หลังจากนั้นทำการหารูปแบบการแบ่งข้อมูลฝึกสอนออกเป็น 2 ส่วนคือข้อมูลที่ซ้อนทับ และข้อมูลที่ไม่ซ้อนทับ ด้วยการแบ่งข้อมูลที่ซ้อนทับกันด้วยเทคนิค Euclidean Distance เมื่อได้ข้อมูลที่ซ้อนทับ และข้อมูลที่ไม่ซ้อนทับเรียบร้อยแล้วจึงทำการสร้างโมเดลในการจำแนกข้อมูลจากข้อมูลทั้งสองส่วนด้วยอัลกอริทึม SVM ซึ่งจะได้ 2 โมเดลในการจำแนกนั่นคือ โมเดลในการจำแนกข้อมูลที่ซ้อนทับ และโมเดลในการจำแนกข้อมูลที่ไม่ซ้อนทับ ซึ่งในการทดสอบความแม่นในการจำแนกข้อมูล จะใช้รูปแบบในการแบ่งข้อมูลจากข้อมูลฝึกสอนมาแบ่งข้อมูลทดสอบออกเป็น 2 ชุดคือ ชุดข้อมูลทดสอบข้อมูลที่ซ้อนทับ และข้อมูลทดสอบข้อมูลที่ไม่ซ้อนทับเพื่อทดสอบความแม่นในการจำแนกข้อมูลทั้ง 2 โมเดลสามารถแสดงภาพรวมการออกแบบวิธีทดสอบได้ดังรูปที่ 4.2



รูปที่ 4.2 การออกแบบวิธีการทดสอบประสิทธิภาพในการจำแนก

จากรูปที่ 4.2 ในการวัดความแม่นในการจำแนกจะใช้มาตรวัดทั้งหมด 5 มาตรวัดได้แก่ Accuracy, TP Rate, TN Rate, F*(F-Measure), G*(G-Means) ซึ่งมาตรวัดทั้ง 5 มาตรวัดจะเป็นตัวชี้วัดความแม่นในการจำแนกข้อมูล ถ้าหากมีค่าสูงแสดงว่ามีประสิทธิภาพในการจำแนกประเภทข้อมูลที่ดี

4.2.1 การทดสอบประสิทธิภาพ

การทดสอบประสิทธิภาพในการจำแนกประเภทข้อมูล ในงานวิจัยนี้สนใจประสิทธิภาพในการจำแนกประเภทข้อมูลส่วนน้อย จึงได้กำหนดให้ Positive หมายถึงคลาสของข้อมูลส่วนน้อย และ Negative คือคลาสของข้อมูลส่วนมาก ดังนั้น TP Rate คือ ประสิทธิภาพในการจำแนกข้อมูลส่วนน้อย และ TN Rate คือ ประสิทธิภาพในการจำแนกข้อมูลส่วนมาก จากรูปที่ 4.3 ซึ่งเป็น Confusion Matrix ที่เป็นตัวอย่างในการคำนวนประสิทธิภาพของโมเดลโดยสามารถคำนวนได้ดังต่อไปนี้

TP Rate สูตรในการคำนวนคือ TP / (TP+FN) แทนค่าแล้วจะได้ 68/(68+22) ซึ่งมีค่าเท่ากับ 0.7556 หรือ 75.56% แสดงว่าโมเดลนี้มีประสิทธิภาพในการจำแนกข้อมูลส่วนน้อย 75.56%

TN Rate สูตรในการคำนวนคือ TN / (TN+FP) เมื่อแทนค่าแล้วจะได้ 148/(148+62) ซึ่งมีค่าเท่ากับ 0.7047 หรือ 70.47% แสดงว่าโมเดลนี้มีประสิทธิภาพในการจำแนกข้อมูลส่วนมาก 70.47%

G* สูตรในการคำนวนคือ $\sqrt{TP\ Rate \bullet TN\ Rate}$ เมื่อแทนค่าแล้วจะได้ $\sqrt{0.7556 \bullet 0.7047}$ ซึ่งมีค่าเท่ากับ 0.7297 หรือ 72.97% นั่นหมายความว่าโมเดลนี้มีความแม่นเฉลี่ยในการจำแนกข้อมูลส่วนน้อยและข้อมูลส่วนมากเท่ากับ 72.97%

F* สูตรในการคำนวนคือ (2 * TP Rate * Precision) / (TP Rate + Precision) เมื่อแทนค่าแล้วจะได้ (2 * 0.7556 * 0.5230) / (0.7556 + 0.5230) ซึ่งมีค่าเท่ากับ 0.6181 หรือ 61.81% นั้นหมายความว่าโมเดลนี้ความแม่นเฉลี่ยในการจำแนกข้อมูลส่วนน้อยเท่ากับ 61.81%

Accuracy สูตรในการคำนวนคือ (TP+TN) / (TP+TN+FN+FP) เมื่อแทนค่าแล้วจะได้ (68+148) / (68+148+22+62) มีค่าเท่ากับ 0.72 หรือ 72% แสดงว่าโมเดลนี้มีความแม่นในการจำแนกประเภทข้อมูลเท่ากับ 72%

Confusion Matrix

Prediction Actual	Positive	Negative
Positive	68 (TP)	22 (FN)
Negative	62 (FP)	148 (TN)

รูปที่ 4.3 ตัวอย่าง Confusion Matrix สำหรับการคำนวน

4.3 ผลการทดสอบประสิทธิภาพ

จากการออกแบบวิธีทดสอบประสิทธิภาพ ซึ่งในงานวิจัยในวิทยานิพนธ์ฉบับนี้ได้ใช้ชุดข้อมูลตัวเลขทั้งหมด 10 ชุดข้อมูล โดยเป็นข้อมูลที่ได้จากการสังเคราะห์จำนวน 3 ชุดข้อมูลและข้อมูลจริงที่นำมาจากฐานข้อมูลมาตรฐานจำนวน 7 ชุดข้อมูล ซึ่งจะใช้มาตรวัดในการวัดระยะห่างทั้งหมด 3 มาตรวัดได้แก่ Euclidean, City Block และ Mahalanobis เมื่อใช้ข้อมูลทดสอบในการแบ่งข้อมูลเป็นชุดข้อมูลที่ซ้อนทับและชุดข้อมูลที่ไม่ซ้อนทับด้วยการวัดระยะแบบ Euclidean จะสามารถแสดงได้ดังตารางที่ 4.3 ส่วนการวัดระยะห่างแบบ City Block จะสามารถแสดงได้ดังตารางที่ 4.4 และการวัดระยะห่างแบบ Mahalanobis จะสามารถแสดงได้ดังตารางที่ 4.5 โดยทั้ง 3 ตารางนั้นจะเป็นการแสดงจำนวนข้อมูลส่วนมาก และข้อมูลส่วนน้อยที่มีอยู่ในชุดข้อมูลที่ซ้อนทับและชุดข้อมูลที่ไม่ซ้อนทับ

ตารางที่ 4.3 แสดงข้อมูลทดสอบที่ได้จากการแบ่งด้วยการวัดระยะแบบ Euclidean

ชุดข้อมูล	Ovelap			Non-Overlap			Total	
	Maj	Min	Total	Maj	Min	Total	Maj	Min
D1	14	12	26	246	28	274	260	40
D2	36	12	48	224	28	252	260	40
D3	72	9	81	188	31	219	260	40
Haberman	48	17	65	19	7	26	67	24
Liver	46	35	81	14	8	22	60	33
Pima	122	33	155	28	47	75	150	80
Page-Blocks	31	3	34	102	5	107	133	8
Vehicle1	124	41	165	65	24	89	189	65
Vehicle3	140	36	176	50	28	78	190	64
German	161	60	221	49	30	79	210	90

ตารางที่ 4.4 แสดงข้อมูลทดสอบที่ได้จากการแบ่งด้วยการวัดระยะแบบ City Block

ชุดข้อมูล	Ovelap			Non-Overlap			Total	
	Maj	Min	Total	Maj	Min	Total	Maj	Min
D1	19	12	31	241	28	269	260	40
D2	33	14	47	227	26	253	260	40
D3	80	12	92	180	28	208	260	40
Haberman	50	17	67	17	7	24	67	24
Liver	48	35	73	12	8	20	60	43
Pima	119	30	149	31	50	81	150	80
Page-Blocks	26	3	29	107	5	112	133	8
Vehicle1	125	42	167	64	23	87	189	65
Vehicle3	134	37	171	56	27	83	190	64
German	156	59	215	54	31	85	210	90

ตารางที่ 4.5 แสดงข้อมูลทดสอบที่ได้จากการแบ่งด้วยการวัดระยะแบบ Mahalanobis

ชุดข้อมูล	Ovelap			Non-Overlap			Total	
	Maj	Min	Total	Maj	Min	Total	Maj	Min
D1	20	2	22	240	38	278	260	40
D2	62	2	64	198	38	236	260	40
D3	63	8	71	197	32	229	260	40
Haberman	49	17	66	18	7	25	67	24
Liver	48	26	74	12	17	29	60	43
Pima	68	40	108	82	40	122	150	80
Page-Blocks	NA	NA	NA	NA	NA	NA	NA	NA
Vehicle1	126	13	139	63	52	115	189	65
Vehicle3	121	6	127	69	58	127	190	64
German	133	59	192	77	31	108	210	90

หลังจากนั้นได้ทำการวัดประสิทธิภาพในการจำแนกประเภทข้อมูลด้วยอัลกอริทึม SVM ซึ่งใช้ kernel ทั้งหมด 3 kernel ได้แก่ Linear, Polynomial, Radial Basis Function (RBF) โดยจะใช้มาตรวัดทั้งหมด 5 มาตรวัดได้แก่ TP Rate, G*, F*, Accuracy, TN Rate ซึ่งจะแสดงประสิทธิภาพในการจำแนกประเภทข้อมูลดังตารางที่ 4.6 – 4.10

ตารางที่ 4.6 แสดงค่า TP Rate ที่ได้จากชุดข้อมูลทดสอบ

ชุดข้อมูล	Euclidean			CityBlock			Mahalanobis		
	Linear	Poly	RBF	Linear	Poly	RBF	Linear	Poly	RBF
D1	92.50	87.50	95.00	90.00	95.00	85.00	**100**	**100**	**100**
D2	87.50	80.00	82.50	80.00	82.50	80.00	**97.50**	**97.50**	95.00
D3	**87.50**	82.50	**87.50**	85.00	72.50	85.00	**87.50**	82.50	**87.50**
Haberman	**66.67**	50.00	62.50	45.83	50.00	58.33	54.17	50.00	58.33
Liver	**74.42**	58.14	48.84	69.77	65.21	55.81	**74.42**	55.81	65.12
Pima	**81.25**	55.00	61.25	77.50	60.00	67.50	80.00	56.25	76.25
Page-Blocks	**100**	**100**	62.50	**100**	**100**	62.5	NA	NA	NA
Vehicle1	**86.15**	69.23	56.92	84.62	67.69	53.85	83.08	78.46	67.69
Vehicle3	92.19	71.88	56.25	**93.75**	70.31	56.25	85.94	81.25	76.56
German	**71.11**	52.22	5.56	67.78	51.11	1.11	66.67	45.56	1.11

ตารางที่ 4.7 แสดงค่า G* ที่ได้จากชุดข้อมูลทดสอบ

ชุดข้อมูล	Euclidean			CityBlock			Mahalanobis		
	Linear	Poly	RBF	Linear	Poly	RBF	Linear	Poly	RBF
D1	92.60	90.06	94.61	91.15	94.61	90.41	88.58	99.03	**99.42**
D2	88.36	86.82	87.27	86.82	88.17	86.11	90.62	**92.87**	91.67
D3	85.85	85.80	88.74	86.34	80.78	86.52	87.02	87.08	**89.12**
Haberman	**70.53**	58.59	66.21	61.34	58.59	63.97	66.68	54.64	60.47
Liver	69.55	59.88	60.52	68.20	63.37	65.41	71.31	58.67	**89.88**
Pima	75.77	64.65	70.00	72.24	67.53	72.25	**81.65**	67.92	79.39
Page-Blocks	97.72	98.87	79.06	97.72	**99.25**	79.06	NA	NA	NA
Vehicle1	82.41	77.74	70.71	83.03	77.80	70.21	78.45	**85.23**	80.29
Vehicle3	**87.28**	78.77	70.31	87.17	79.32	70.31	78.14	84.51	84.69
German	**70.31**	63.47	23.51	68.88	62.99	10.54	66.9	60.55	10.52

ตารางที่ 4.8 แสดงค่า F* ที่ได้จากชุดข้อมูลทดสอบ

ชุดข้อมูล	Euclidean			CityBlock			Mahalanobis		
	Linear	Poly	RBF	Linear	Poly	RBF	Linear	Poly	RBF
D1	77.08	74.47	81.72	75.00	81.72	80.95	58.82	94.12	**96.39**
D2	67.96	73.56	70.97	73.56	**75.00**	70.33	65.00	71.56	70.37
D3	60.34	65.35	69.31	64.15	61.05	64.76	63.64	70.21	**70.71**
Haberman	**56.14**	42.11	50.85	46.81	42.11	48.28	53.06	38.01	44.44
Liver	66.67	54.95	53.16	64.52	59.57	59.26	**68.09**	53.33	65.12
Pima	68.78	55.00	61.64	64.92	58.54	64.29	**75.74**	59.21	73.05
Page-Blocks	72.73	84.21	76.92	72.73	**88.89**	76.92	NA	NA	NA
Vehicle1	69.57	67.16	59.20	70.97	68.22	60.34	64.29	**78.46**	74.58
Vehicle3	75.64	67.65	58.54	75.00	69.77	58.54	63.22	74.82	**78.40**
German	**58.72**	50.81	10.42	57.01	50.27	2.20	54.79	47.67	2.17

ตารางที่ 4.9 แสดงค่า Accuracy ที่ได้จากชุดข้อมูลทดสอบ

ชุดข้อมูล	Euclidean			CityBlock			Mahalanobis		
	Linear	Poly	RBF	Linear	Poly	RBF	Linear	Poly	RBF
D1	92.67	92.00	94.33	92.00	94.33	94.67	81.33	98.33	**99.00**
D2	89.00	92.33	91.00	92.33	**92.67**	91.00	86.00	89.67	89.33
D3	84.67	88.33	89.67	87.33	87.67	87.67	86.67	**90.67**	90.33
Haberman	72.53	63.74	68.13	72.53	63.74	67.03	**74.73**	57.14	61.54
Liver	68.93	60.19	64.08	67.96	63.11	67.96	**70.87**	59.22	70.87
Pima	74.35	68.70	73.48	70.87	70.43	73.91	**82.17**	73.04	80.43
Page-Blocks	95.74	97.87	97.87	95.74	**98.58**	97.87	NA	NA	NA
Vehicle1	80.71	82.68	79.92	82.28	83.86	81.89	76.38	**88.98**	88.19
Vehicle3	85.04	82.68	79.92	84.25	84.65	79.92	74.80	86.22	**89.37**
German	70.00	69.67	**71.33**	69.33	69.67	70.33	67.00	70.00	70.00

ตารางที่ 4.10 แสดงค่า TN Rate ที่ได้จากชุดข้อมูลทดสอบ

ชุดข้อมูล	Euclidean			CityBlock			Mahalanobis		
	Linear	Poly	RBF	Linear	Poly	RBF	Linear	Poly	RBF
D1	92.69	92.69	94.23	92.31	94.23	96.15	78.46	98.08	**98.85**
D2	89.23	**94.23**	92.31	**94.23**	**94.23**	92.69	84.23	88.46	88.46
D3	84.23	59.23	90.00	87.69	90.00	88.08	86.54	**91.92**	90.77
Haberman	74.63	68.66	70.15	**82.09**	68.66	70.15	**82.09**	59.70	62.69
Liver	65.00	61.67	75.00	66.67	61.67	**76.67**	68.33	61.67	75.00
Pima	70.67	76.00	80.00	67.33	76.00	77.33	**83.33**	82.00	82.67
Page-Blocks	95.47	97.74	**100**	95.49	98.5	**100**	NA	NA	NA
Vehicle1	78.84	87.30	87.83	81.48	89.42	91.53	74.07	92.59	**95.24**
Vehicle3	82.63	86.32	87.89	81.05	89.47	87.89	71.05	87.89	**93.68**
German	69.52	77.14	99.52	70.00	77.62	**100**	67.14	80.48	99.52

เมื่อได้ประสิทธิภาพในการจำแนกประเภทข้อมูลเรียบร้อยแล้วได้นำผลลัพธ์ไปเปรียบเทียบกับงานวิจัย Piyanoot et al. (2015) ซึ่งจะเปรียบเทียบประสิทธิภาพในการจำแนกข้อมูลด้วย 3 มาตรวัดได้แก่ TP Rate, G* และ F* เนื่องจากค่า TN Rate และ Accuracy ในงานวิจัยที่ใช้ในการเปรียบเทียบไม่ได้ระบุไว้จึงใช้เพียง 3 มาตรวัดดังกล่าว โดยจะใช้ประสิทธิภาพของการจำแนกประเภทข้อมูลด้วยการวัดแบบ Euclidean โดยใช้อัลกอริทึม SVM kernel คือ Linear kernel ซึ่งให้ผลลัพธ์ดีที่สุด ซึ่งสามารถแสดงได้ดังตารางที่ 4.11-4.13 และรูปที่ 4.4-4.6

ตารางที่ 4.11 แสดงการเปรียบเทียบประสิทธิภาพการจำแนกประเภทข้อมูล

ชุดข้อมูล	SVM (linear)			Piyanoot et al. (2015)			วิทยานิพนธ์ฉบับนี้		
	TPR	G*	F*	TPR	G*	F*	TPR	G*	F*
Haberman	33.33	55.09	42.11	54.32	**71.03**	**62.41**	**66.67**	70.53	56.14
Liver	53.49	61.19	54.76	65.52	68.68	64.19	**74.42**	**69.55**	**66.67**
Pima	57.5	69.5	61.33	59.06	71.12	63.58	**81.25**	**75.77**	**68.78**
Page-Blocks	87.5	91.41	66.67	59.19	71.86	58.90	**100**	**97.72**	**72.73**
Vehicle1	78.46	74.03	58.96	47.57	65.06	53.09	**86.15**	**82.41**	**69.57**
Vehicle3	84.38	77.71	62.79	45.38	62.56	48.73	**92.19**	**87.28**	**75.64**
German	66.67	66.19	54.05	54.32	67.26	57.42	**71.11**	**70.31**	**58.72**

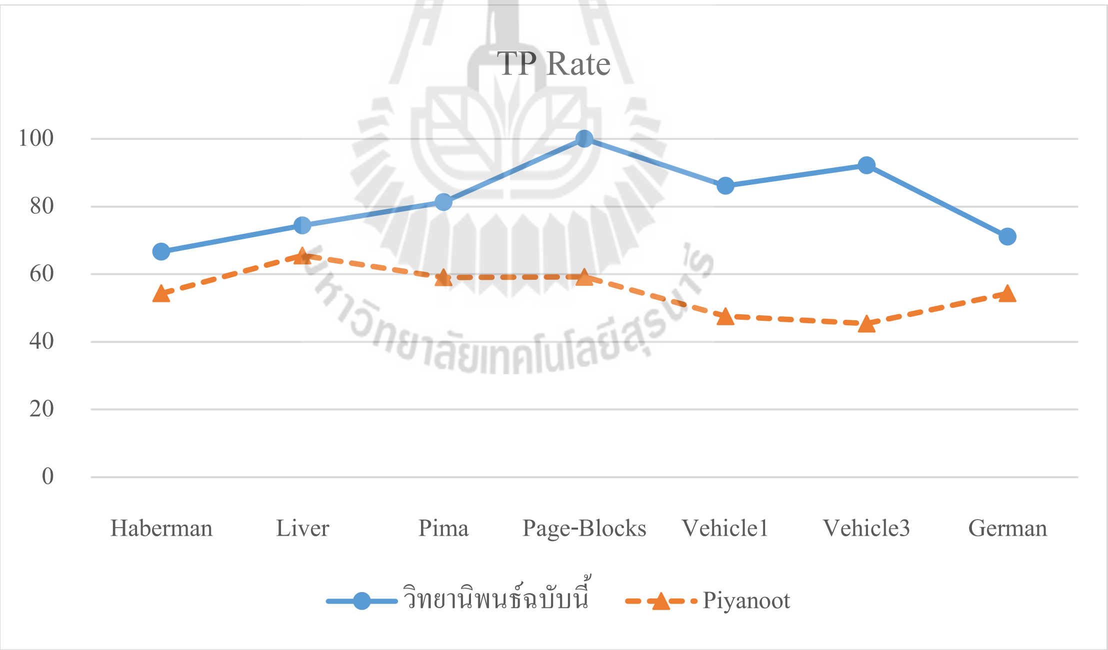
ตารางที่ 4.12 แสดงการเปรียบเทียบประสิทธิภาพการจำแนกประเภทข้อมูลด้วยวิธีสุ่มข้อมูล

ชุดข้อมูล	Random Undersampling			Random Oversampling			วิทยานิพนธ์ฉบับนี้		
	TPR	G*	F*	TPR	G*	F*	TPR	G*	F*
Haberman	38.27	58.35	49.21	65.36	**78.29**	**75.48**	**66.67**	70.53	56.14
Liver	6.21	24.60	11.32	11.45	40.43	27.66	**74.42**	**69.55**	**66.67**
Pima	54.85	70.65	63.91	52.99	68.99	61.61	**81.25**	**75.77**	**68.78**
Page-Blocks	50.00	70.71	66.67	94.75	97.23	**97.14**	**100**	**97.72**	72.73
Vehicle1	18.43	42.73	30.42	10.14	31.72	18.03	**86.15**	**82.41**	**69.57**
Vehicle3	8.49	28.98	15.49	5.66	23.72	10.53	**92.19**	**87.28**	**75.64**
German	49.33	65.86	55.74	55.91	**70.43**	**61.73**	**71.11**	70.31	58.72

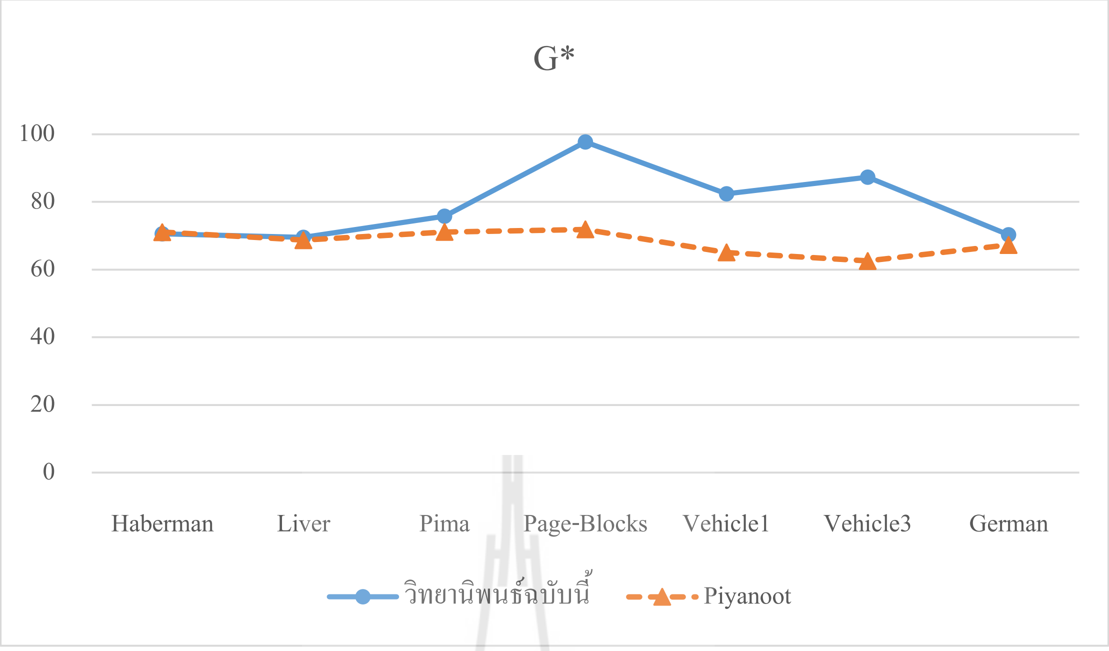
ตารางที่ 4.13 แสดงค่าเฉลี่ยของแต่ละมาตรวัดของชุดข้อมูลจริง

มาตรวัด	Euclidean			CityBlock			Mahalanobis		
	Linear	Poly	RBF	Linear	Poly	RBF	Linear	Poly	RBF
TP Rate	**81.68**	65.21	50.55	77.04	66.33	50.76	74.05	61.22	57.51
G*	**79.08**	71.71	62.90	76.94	72.69	61.68	73.86	68.59	67.54
F*	**66.89**	60.27	52.96	64.57	62.48	52.83	63.20	58.58	56.29
Accuracy	**78.19**	75.08	76.39	77.57	76.29	76.99	74.33	72.43	76.73
TN Rate	76.68	79.26	85.77	77.73	80.19	86.22	74.34	77.39	**84.80**

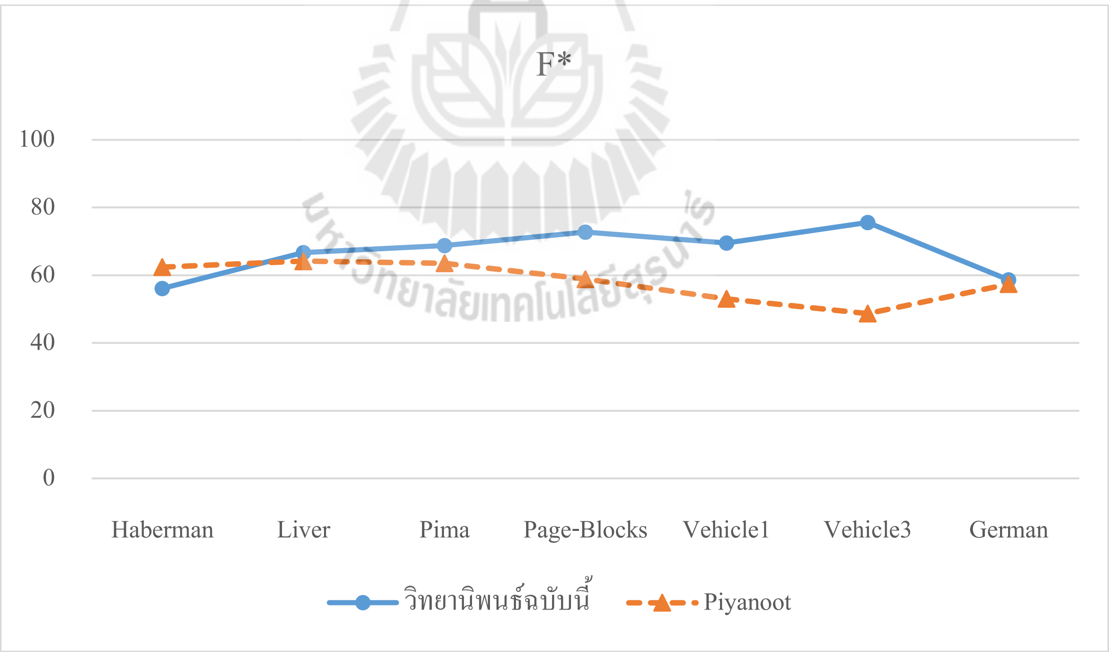
จากตารางที่ 4.13 TP Rate คือ ความแม่นยำในการจำแนกประเภทข้อมูลส่วนน้อย, TN Rate คือความแม่นยำในการจำแนกประเภทข้อมูลส่วนมาก, F* คือความแม่นยำในการจำแนกประเภทข้อมูลโดยรวม



รูปที่ 4.4 แสดงการเปรียบเทียบ TP Rate



รูปที่ 4.5 แสดงการเปรียบเทียบ G*



รูปที่ 4.6 แสดงการเปรียบเทียบ F*

4.4 อภิปรายผล

จากผลการทดสอบประสิทธิภาพการจำแนกประเภทข้อมูลที่แสดงข้างต้น ซึ่งเป็นการทดสอบประสิทธิภาพในการจำแนกข้อมูลทั้งหมด 10 ชุดข้อมูล โดยแบ่งเป็นชุดข้อมูลสังเคราะห์จำนวน 3 ชุดข้อมูล และข้อมูลจริงจากฐานข้อมูลมาตรฐานจำนวน 7 ชุดข้อมูล ซึ่งในงานวิจัยนี้จะมุ่งเน้นในการจำแนกข้อมูลส่วนน้อยให้ถูกต้องมากที่สุด โดยที่จำแนกข้อมูลส่วนมากได้อย่างมีประสิทธิภาพเช่นเดียวกัน จากผลการทดสอบสามารถสรุปได้ดังต่อไปนี้

4.4.1 อภิปรายผลชุดข้อมูลสังเคราะห์

ชุดข้อมูลสังเคราะห์ทั้ง 3 ชุดที่สังเคราะห์ขึ้นมานั้นมีจุดกึ่งกลางของแต่ละคลาสที่ตำแหน่งเดียวกันแต่แตกต่างกันที่การกระจายตัวของข้อมูลส่วนน้อยเท่านั้น ซึ่งทำให้การซ้อนทับ (OR) ไม่เท่ากันโดยที่ชุดข้อมูล D1, D2 และ D3 มีค่าซ้อนทับเท่ากับ 39.99, 51.32 และ 65.97 ตามลำดับ จากผลการทดสอบประสิทธิภาพในการจำแนกประเภทข้อมูลจากตารางที่ 4.6 นั้นทำให้ทราบว่า ชุดข้อมูลที่มีการซ้อนทับกันของข้อมูลที่มีค่าน้อย จะสามารถจำแนกประเภทข้อมูลได้แม่นยำกว่าชุดข้อมูลที่มีการซ้อนทับกันของข้อมูลที่มีค่ามาก และสำหรับชุดข้อมูลสังเคราะห์การวัดระยะแบบ Mahalanobis และ SVM RBF kernel ให้ผลลัพธ์ในการจำแนกข้อมูลดีที่สุด

4.4.2 อภิปรายผลชุดข้อมูลจากฐานข้อมูลมาตรฐาน

ชุดข้อมูลที่นำข้อมูลมาจากฐานข้อมูลมาตรฐานนั้นเป็นข้อมูลตัวเลขทั้งหมด โดยในงานวิจัยนี้ได้นำข้อมูลจากฐานข้อมูลมาตรฐานมาทั้งหมดจำนวน 7 ข้อมูล ได้แก่ Haberman, Liver, Pima, Page-Blocks, Vehicle1, Vehicle3 และ German ซึ่งชุดข้อมูลแต่ละชุดข้อมูลมีรายละเอียดแสดงดังตารางที่ 4.2 และได้ทำการแบ่งข้อมูลด้วยการวัดระยะ 3 แบบคือ Euclidean, City Block และ Mahalanobis โดยแสดงรายละเอียดดังตารางที่ 4.3 - 4.5 ซึ่งข้อมูล Page-Blocks ในการวัดระยะทางแบบ Mahalanobis นั้นไม่สามารถทำได้เนื่องจากจำนวนข้อมูลส่วนน้อยในชุดข้อมูลทดสอบมีจำนวน 8 ข้อมูล ซึ่งมีจำนวนน้อยกว่าคุณสมบัติ(Attributes)ของชุดข้อมูลซึ่งมี 10 คุณสมบัติ ทำให้ไม่สามารถหาระยะห่างระหว่างข้อมูลของชุดข้อมูล Page-Blocks ได้

จากผลการทดสอบ TP Rate จากตารางที่ 4.6 ซึ่งเป็นค่าที่บ่งบอกถึงประสิทธิภาพในการจำแนกประเภทข้อมูลส่วนน้อย นั้นสามารถสรุปได้ว่าการแบ่งข้อมูลด้วยการวัดระยะแบบ Euclidean นั้นให้ประสิทธิภาพในการจำแนกประเภทพข้อมูลส่วนน้อยได้ดีที่สุด โดยใช้ SVM Linear kernel ในการจำแนกประเภทข้อมูล ซึ่งชุดข้อมูลที่สามารถจำแนกประเภทข้อมูลได้ดีที่สุดคือชุดข้อมูล Vehicle3 โดยมีค่า TP Rate เท่ากับ 92.19 % และชุดข้อมูลที่สามารถจำแนกประเภท

ข้อมูลได้ต่ำที่สุดคือ Haberman โดยมีค่า TP Rate เท่ากับ 66.67% ซึ่งเมื่อเทียบกับงานวิจัยที่นำมาเปรียบเทียบในตารางที่ 4.11 แล้วจะเห็นได้ว่าค่า TP Rate ที่ได้จากงานวิจัยของวิทยานิพนธ์ฉบับนี้มีค่าสูงกว่างานวิจัยที่นำมาเปรียบเทียบทุกชุดข้อมูล

จากผลการทดสอบ G* จากตารางที่ 4.7 ซึ่งเป็นค่าความแม่นเฉลี่ยของการจำแนกประเภทข้อมูลส่วนน้อยและข้อมูลส่วนมาก สามารถสรุปได้ว่าการใช้การวัดระยะแบบ Euclidean และใช้ SVM Linear kernel ในการจำแนกประเภทข้อมูลนั้นให้ผลลัพธ์ดีที่สุดโดยจากตารางที่ 4.13 มีค่าเฉลี่ยของค่า G* ของชุดข้อมูลจริงทั้งหมด 7 ชุดข้อมูลมีค่าเท่ากับ 79.08% โดยเมื่อเปรียบเทียบกับงานวิจัยที่นำมาเปรียบเทียบในตารางที่ 4.11 แล้วจะเห็นได้ว่าค่า G* ที่ได้จากงานวิจัยของวิทยานิพนธ์ฉบับนี้มีค่าสูงกว่างานวิจัยที่นำมาเปรียบเทียบทุกชุดข้อมูล

จากผลการทดสอบ F* จากตารางที่ 4.8 ซึ่งเป็นค่าความแม่นเฉลี่ยของการจำแนกประเภทข้อมูลส่วนน้อย โดยสามารถสรุปได้ว่าการใช้การวัดระยะห่างแบบ Euclidean และใช้ SVM Linear kernel ในการจำแนกประเภทข้อมูลให้ผลลัพธ์ที่ดีที่สุด ซึ่งจากตารางที่ 4.13 ค่าเฉลี่ยของค่า F* ของข้อมูลจริงทั้ง 7 ชุดข้อมูลมีค่าเท่ากับ 66.89% เมื่อนำค่า F* มาเปรียบเทียบกับงานวิจัยที่นำมาเปรียบเทียบในตารางที่ 4.11 แล้วจะเห็นได้ว่าค่า F* ที่ได้จากงานวิจัยของวิทยานิพนธ์ฉบับนี้มีค่าสูงกว่างานวิจัยที่นำมาเปรียบเทียบเกือบทุกชุดข้อมูล ยกเว้นชุดข้อมูล Haberman เท่านั้น

จากผลการทดสอบค่า Accuracy จากตารางที่ 4.9 ซึ่งเป็นค่าความแม่นในการจำแนกประเภทข้อมูลโดยรวมของแต่ละชุดข้อมูล นั้นสามารถสรุปได้ว่าการใช้วิธีวัดระยะห่างแบบ Euclidean และใช้ SVM Linear kernel ในการจำแนกประเภทข้อมูลให้ผลลัพธ์ที่ดีที่สุดโดยมีค่าเฉลี่ยของข้อมูลจริงทั้ง 7 ชุดข้อมูลเท่ากับ 78.19% ดังแสดงในตารางที่ 4.13 ซึ่งในงานวิจัยที่นำมาเปรียบเทียบไม่ได้ระบุค่า Accuracy ไว้จึงไม่สามารถนำไปเปรียบเทียบได้

จากผลการทดสอบค่า TN Rate จากตารางที่ 4.10 ซึ่งเป็นค่าความแม่นในการจำแนกประเภทข้อมูลส่วนมาก สามารถสรุปจากตารางที่ 4.13 ได้ว่าวิธีวัดระยะห่างแบบ Cityblock และใช้ SVM RBF kernel นั้นให้ผลลัพธ์ที่ดีที่สุดคือมีค่าเฉลี่ยของค่า TN Rate ของชุดข้อมูลจริงทั้ง 7 ชุดข้อมูลมีค่าเท่ากับ 86.22% ซึ่งในงานวิจัยที่นำมาเปรียบเทียบไม่ได้ระบุค่า TN Rate จึงไม่สามารถทำการเปรียบเทียบความแตกต่างได้

จากผลการเปรียบเทียบในตารางที่ 4.11 ซึ่งเป็นการเปรียบเทียบประสิทธิภาพของงานวิจัยนี้กับการใช้ SVM Linear kernel จำแนกประเภทข้อมูลโดยไม่มีการแบ่งข้อมูลส่วนย่อย และเปรียบเทียบกับงานวิจัยของ Piyanoot et al. (2015) นั้นแสดงให้เห็นว่าวิธีการที่งานวิจัยนี้นำเสนอส่วนมากให้ประสิทธิภาพในการจำแนกประเภทข้อมูลที่ดีกว่า

จากผลการเปรียบเทียบในตารางที่ 4.12 ซึ่งเป็นการเปรียบเทียบประสิทธิภาพของงานวิจัยนี้และเทคนิคการสุ่มเพิ่ม (Random Oversampling) และเทคนิคการสุ่มลด (Random Undersampling) โดยนำผลมาจากงานวิจัยของ Piyanoot et al. (2015) จะสามารสรุปได้ว่าวิธีการที่งานวิจัยนี้นำเสนอมานั้นส่วนมากแล้วให้ประสิทธิภาพในการจำแนกข้อมูลที่ดีกว่าเทคนิคการสุ่ม

จากผลการทดสอบค่าความแม่นยำในการจำแนกประเภทข้อมูลด้วยมาตรวัด 5 มาตรวัดที่กล่าวมาข้างต้นสามารถสรุปได้ว่าการใช้วิธีการแบ่งข้อมูลด้วยวิธีการวัดระยะแบบ Euclidean โดยใช้ SVM Linear kernel เป็นวิธีการที่ดีที่สุดในการจำแนกประเภทข้อมูลส่วนน้อยบนข้อมูลที่ไม่สมดุล ซึ่งการใช้การวัดระยะแบบ Euclidean ในการแบ่งข้อมูลนั้นเป็นวิธีที่แบ่งข้อมูลที่ซ้อนทับ และไม่ซ้อนทับ ออกจากกันได้ดีที่สุด เนื่องจาก เมื่อทำการแบ่งข้อมูลที่ซ้อนทับ และไม่ซ้อนทับ แล้วนำข้อมูลไปสร้างโมเดลเพื่อทดสอบประสิทธิภาพในการจำแนกแล้วให้ผลลัพธ์ที่ดีที่สุด ดังนั้นสิ่งที่ทำให้งานวิจัยนี้มีประสิทธิภาพในการจำแนกที่ดีนั้นมาจากการแบ่งข้อมูลที่ซ้อนทับและข้อมูลที่ไม่ซ้อนทับ ออกจากกันได้อย่างมีประสิทธิภาพ จึงทำให้โมเดลที่ได้มีประสิทธิภาพที่ดีตามไปด้วย โดยวิธีการแบ่งข้อมูลด้วยการวัดระยะแบบ Euclidean จะทำให้การแบ่งข้อมูลที่ซ้อนทับและข้อมูลที่ไม่ซ้อนทับที่ใกล้เคียงกับความจริงมากที่สุด และเมื่อนำมาใช้กับ SVM Linear kernel ซึ่งเป็นอัลกอริทึมในการจำแนกประเภทข้อมูลที่ใช้สมการเชิงเส้นเป็นพื้นฐานในการ จำแนกประเภทข้อมูลด้วยแล้วทำให้ประสิทธิภาพที่ได้ดีกว่าการใช้ SVM kernel อื่น ๆ และ

เมื่อเปรียบเทียบกับงานวิจัยของ Piyanoot et al. (2015) แล้วสิ่งที่แตกต่างกันคือในงานวิจัยของ Piyanoot นั้นจะเป็นการใช้วิธีการแบ่งข้อมูลด้วยการวัดระยะแบบ Hausdorff และแบ่งข้อมูลออกเป็น 3 ส่วนคือ ส่วนที่ซ้อนทับ ส่วนขอบ และส่วนที่ไม่ซ้อนทับ และใช้อัลกอริทึมในการจำแนกข้อมูลในแต่ละส่วนที่แตกต่างกัน แต่ในงานวิจัยนี้จะใช้การแบ่งข้อมูลด้วยการวัดระยะแบบ Euclidean และแบ่งข้อมูลออกเป็น 2 ส่วนคือ ส่วนที่ซ้อนทับ และส่วนที่ไม่ซ้อนทับ และใช้อัลกอริทึมในการจำแนกเพียงอัลกอริทึมเดียวคือ SVM Linear kernel โดยสิ่งที่ทำให้ประสิทธิภาพในการจำแนกที่แตกต่างกันคือการใช้การแบ่งข้อมูลด้วยการวัดระยะแบบ Hausdorff นั้นจะไม่ได้ข้อมูลที่ซ้อนทับกันที่ใกล้เคียงความเป็นจริง โดยจะมีข้อมูลที่ไม่ซ้อนทับกันบางส่วนเข้าไปปะปนกับข้อมูลที่ซ้อนทับกันทำให้ประสิทธิภาพในการจำแนกที่ได้ไม่ดีเท่าที่ควร ซึ่งในงานวิจัยนี้ใช้การแบ่งข้อมูลด้วยการวัดระยะแบบ Euclidean จะทำให้การแบ่งข้อมูลที่ซ้อนทับ และข้อมูลที่ไม่ซ้อนทับใกล้เคียงกับความเป็นจริงมากกว่าจึงทำให้มีประสิทธิภาพในการจำแนกที่ดีกว่า

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

ในปัจจุบันการเก็บข้อมูลสามารถทำได้อย่างง่ายดายและมีประสิทธิภาพ ซึ่งสามารถขุดค้นความรู้ใหม่ได้จากการทำเหมืองข้อมูล เพื่อนำความรู้ใหม่ไปใช้ประโยชน์ในด้านต่าง ๆ เช่น การทำเหมืองข้อมูลกับข้อมูลสภาพอากาศ เพื่อพยากรณ์สภาพอากาศในอนาคต หรือ การทำเหมืองข้อมูลกับข้อมูลการซื้อสินค้า เพื่อจัดโปรโมชั่นสินค้า หรือการสั่งสำรองสินค้าเพื่อที่สินค้าจะไม่ขาดตลาด เป็นต้น ซึ่งในปัจจุบันข้อมูลที่จัดเก็บบางข้อมูลจะเป็นข้อมูลที่ไม่สมดุล คือมีข้อมูลบางชนิดมีจำนวนมาก และข้อมูลอีกชนิดมีจำนวนน้อย ยกตัวอย่างเช่น ข้อมูลการผลิตชิ้นส่วนรถยนต์ซึ่งข้อมูลของชิ้นส่วนที่ปกติจะมีจำนวนมากกว่าข้อมูลของชิ้นส่วนที่ผิดปกติหรือเสียหาย ซึ่งข้อมูลชนิดนี้คือข้อมูลไม่สมดุล โดยในปัจจุบันการทำเหมืองข้อมูลโดยใช้ข้อมูลไม่สมดุลนั้นสามารถทำได้ยาก เนื่องจาก คุณสมบัติของข้อมูลส่วนใหญ่จะบดบังคุณสมบัติของข้อมูลส่วนน้อย ซึ่งทำให้การทำเหมืองข้อมูลด้วยการจำแนกประเภทข้อมูลนั้นมีประสิทธิภาพในการจำแนกข้อมูลส่วนน้อยต่ำ ซึ่งได้มีการแก้ไขปัญหานี้ด้วยวิธีการต่าง ๆ เช่น การลดข้อมูลส่วนมากโดยให้มีจำนวนข้อมูลใกล้เคียงกับข้อมูลส่วนน้อย หรือจะเป็นการเพิ่มข้อมูลส่วนน้อยให้มีปริมาณข้อมูลใกล้เคียงกับข้อมูลส่วนมาก หรืออาจจะใช้ทั้งสองวิธีทั้งวิธีการเพิ่มข้อมูลส่วนน้อย และการลดข้อมูลส่วนมาก โดยให้มีปริมาณที่ใกล้เคียงกัน ซึ่งงานวิจัยในบางงานก็จะมุ่งเน้นไปที่การพัฒนาอัลกอริทึมในการจำแนกข้อมูลโดยมีการปรับค่าน้ำหนักเพื่อเพิ่มประสิทธิภาพในการจำแนกข้อมูลส่วนน้อยให้เพิ่มขึ้น โดยการใช้วิธีดังกล่าวก็สามารถเพิ่มประสิทธิภาพในการจำแนกข้อมูลส่วนน้อยได้ แต่ยังไม่ดีเท่าที่ควร

ดังนั้นวัตถุประสงค์ของงานวิจัยของวิทยานิพนธ์ฉบับนี้คือการเพิ่มประสิทธิภาพในการจำแนกข้อมูลส่วนน้อย และทำการเปรียบเทียบประสิทธิภาพในการจำแนกกับงานวิจัยอื่น โดยวิธีการที่นำเสนอในวิทยานิพนธ์ฉบับนี้คือ เทคนิคการจำแนกประเภทข้อมูลบนข้อมูลไม่สมดุลด้วยวิธีการแบ่งข้อมูล โดยวิธีการแบ่งข้อมูลที่ใช้ในงานวิจัยนี้จะแบ่งข้อมูลออกเป็น 2 กลุ่มคือ กลุ่มข้อมูลที่ไม่ซ้อนทับกัน และกลุ่มข้อมูลที่ซ้อนทับกัน ซึ่งในการแบ่งข้อมูลนั้นจะใช้การวัดระยะ 3 แบบคือ Euclidean, City Block และ Mahalanobis โดยข้อมูลทั้งสองกลุ่มจะถูกนำไปสร้างเป็นโมเดลในการจำแนกข้อมูลด้วยอัลกอริทึมซัพพอร์ตเวกเตอร์แมชชีน (SVM) ซึ่งจะใช้ kernel ทั้งหมด 3 kernel ได้แก่ Linear, Polynomial และ Radial Basis Function (RBF) โดยในแต่ละชุดข้อมูลจะมี 3 โมเดลในการจำแนกประเภทข้อมูลคือ โมเดลสำหรับจำแนกประเภทข้อมูลที่ซ้อนทับ

กัน และโมเดลสำหรับจำแนกประเภทข้อมูลที่ไม่ซ้อนทับกัน ซึ่งในการทดสอบประสิทธิภาพด้วยข้อมูลทดสอบก็จะทำการแบ่งข้อมูลออกเป็น 2 กลุ่มข้อมูลเช่นกัน โดยจะใช้รูปแบบในการแบ่งข้อมูลที่ได้จากการแบ่งข้อมูลฝึกสอน หลังจากนั้นจึงใช้ข้อมูลทั้งสองกลุ่มทดสอบประสิทธิภาพของโมเดลในการจำแนก ซึ่งประสิทธิภาพในการจำแนกประเภทข้อมูลแต่ละชุดข้อมูลที่นำมาทดสอบนั้นจะเป็นประสิทธิภาพโดยรวมจากโมเดลจำแนกประเภทข้อมูลทั้งสองโมเดล

5.1 สรุปผลการวิจัย

จากผลการทดสอบประสิทธิภาพในการจำแนกข้อมูลที่แสดงในบทที่ 4 นั้นซึ่งได้ใช้ข้อมูลไม่สมดุล 2 ชนิดคือชุดข้อมูลสังเคราะห์จำนวน 3 ชุดข้อมูลและชุดข้อมูลจริงที่นำมาจากฐานข้อมูลมาตรฐานจำนวน 7 ชุดข้อมูลสามารถสรุปได้ดังต่อไปนี้

1. จากผลการทดสอบประสิทธิภาพการจำแนกข้อมูลสังเคราะห์ซึ่งแต่ละชุดข้อมูลมีการกระจายตัวของข้อมูลส่วนน้อยที่แตกต่างกัน โดยสามารถสรุปได้ว่าถ้าหากชุดข้อมูลที่มีการกระจายมากจะทำให้เกิดการซ้อนทับของข้อมูลที่มากขึ้นด้วยทำให้ประสิทธิภาพในการจำแนกประเภทข้อมูลส่วนน้อยนั้นมีความแม่นยำในการจำแนกต่ำ กว่าชุดข้อมูลที่มีการซ้อนทับกันของข้อมูลที่น้อยกว่า ซึ่งวิธีการวัดระยะห่างที่ทำให้มีประสิทธิภาพในการจำแนกข้อมูลส่วนน้อยที่ดีที่สุดคือวิธีวัดระยะห่างแบบ Mahalanobis ซึ่งใช้ SVM RBF kernel ในการจำแนกประเภทข้อมูล

2. จากผลการทดสอบประสิทธิภาพในการจำแนกข้อมูลจริงจากข้อมูลมาตรฐานจำนวน 7 ชุดข้อมูลนั้นสามารถสรุปได้ว่า วิธีการแบ่งข้อมูลนั้นสามารถช่วยเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลส่วนน้อยได้อย่างมีประสิทธิภาพ โดยจากการเปรียบเทียบประสิทธิภาพในการจำแนกกับงานวิจัยของ Piyanoot และคณะ (2015) ผลปรากฏว่าค่า TP Rate, G-Means และค่า F-Measure นั้นทุกชุดข้อมูลให้ประสิทธิภาพในการจำแนกข้อมูลส่วนน้อยที่มากกว่า มีเพียงค่า F-Measure ของชุดข้อมูล Haberman เท่านั้นที่มีค่าน้อยกว่างานวิจัยที่นำมาเปรียบเทียบ ซึ่งข้อมูล Haberman นั้นมีค่าความซ้อนทับของข้อมูล (OR) เท่ากับ 89.86% และมีจำนวนคุณสมบัติ (Attribute) เท่ากับ 3 คุณสมบัติ เมื่อทำการจำแนกประเภทข้อมูลส่วนน้อยแล้วได้ค่า TP Rate เท่ากับ 66.67% ส่วนข้อมูล German นั้นมีค่าความซ้อนทับของข้อมูลเท่ากับ 98.45% และมีจำนวนคุณสมบัติเท่ากับ 24 คุณสมบัติ โดยเมื่อทำการจำแนกประเภทข้อมูลมีค่า TP Rate เท่ากับ 71.11% ซึ่งอาจจะเกิดจากข้อมูล Haberman นั้นมีค่าการซ้อนทับกันที่มากและยังมีค่าคุณสมบัติของข้อมูลที่น้อย ทำให้ประสิทธิภาพในการจำแนกประเภทข้อมูลนั้นด้อยกว่าชุดข้อมูล German ซึ่งมีค่าคุณสมบัติที่มากกว่า โดยสรุปได้ว่าหากข้อมูลมีค่าความซ้อนทับที่มากและมีจำนวนคุณสมบัติข้อมูล

ที่น้อยจะทำให้ประสิทธิภาพในการจำแนกประเภทข้อมูลที่ต่ำกว่า ชุดข้อมูลที่มีค่าความซ้อนทับที่น้อย และจำนวนคุณสมบัติที่มาก

3. จากผลการทดสอบประสิทธิภาพในการจำแนกประเภทข้อมูลในบทที่ 4 นั้นสามารถสรุปได้ว่าวิธีการที่ทำให้ประสิทธิภาพในการจำแนกประเภทข้อมูลในงานวิจัยนี้มีผลลัพธ์ที่ดีนั้นมาจากการใช้วิธีการแบ่งข้อมูลที่ซ้อนทับ และข้อมูลที่ไม่ซ้อนทับให้ใกล้เคียงกับความเป็นจริงมากที่สุดโดยการใช้การแบ่งข้อมูลด้วยการวัดระยะแบบ Euclidean และใช้อัลกอริทึมที่มีพื้นฐานที่ใช้การวัดระยะทางแบบ Euclidean และสมการเชิงเส้นในการจำแนกประเภทข้อมูลซึ่งก็คือ SVM Linear kernel ในการจำแนกประเภทข้อมูล ทำให้สามารถเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลได้ และการแบ่งข้อมูลซ้อนทับ และข้อมูลไม่ซ้อนทับให้ผลดีกว่าการไม่แบ่งข้อมูลซึ่งผลแสดงในตารางที่ 4.11 ซึ่งทุกชุดข้อมูลที่ทำการแบ่งข้อมูลซ้อนทับ และไม่ซ้อนทับนั้นให้ผลลัพธ์ที่ดีกว่าทุกชุดข้อมูล

5.2 ปัญหาและข้อเสนอแนะ

การจำแนกประเภทข้อมูลส่วนน้อยบนข้อมูลที่ไม่สมดุลนั้นยังไม่มีวิธีที่แก้ปัญหาได้อย่างแน่นอน ซึ่งจะเห็นได้ว่าวิธีที่นำเสนอในวิทยานิพนธ์ฉบับนี้เมื่อข้อมูลที่มีการซ้อนทับกันของข้อมูลมีค่าที่สูงจะทำให้ประสิทธิภาพในการจำแนกข้อมูลต่ำลง โดยที่สามารถจำแนกประเภทข้อมูลส่วนน้อยกับข้อมูลที่มีค่าซ้อนทับต่ำได้ดีกว่า

ดังนั้นสิ่งที่จะเสนอแนะคือ การเพิ่มกระบวนการลดการซ้อนทับกันของข้อมูลซึ่งอาจจะใช้เทคนิคการเลือกคุณสมบัติ มาช่วยในการลดความซ้อนทับการซ้อนทับกันของข้อมูล ซึ่งอาจจะเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลได้ดียิ่งขึ้น ส่วนการนำเทคนิคการจำแนกประเภทข้อมูลส่วนน้อยบนข้อมูลไม่สมดุลด้วยวิธีการแบ่งข้อมูลไปใช้ประโยชน์นั้นสามารถนำไปใช้ในการจำแนกข้อมูลที่ต้องการโดยที่สิ่งที่สินใจจะจำแนกนั้นมีจำนวนข้อมูลน้อยเมื่อเทียบกับข้อมูลทั้งหมด ยกตัวอย่างเช่น การจำแนกผลิตภัณฑ์ที่เสียหายจากการผลิต ออกจากข้อมูลการผลิต ซึ่งข้อมูลการผลิตข้อมูลส่วนมากจะเป็นข้อมูลผลิตภัณฑ์ที่มีคุณภาพดี และมีผลิตภัณฑ์ที่เสียหายเป็นข้อมูลส่วนน้อย แต่สนใจที่จะจำแนกผลิตภัณฑ์ที่เสียหายเพื่อนำไปปรับปรุงกระบวนการผลิต โดยปัญหาในการจำแนกลักษณะนี้สามารถใช้เทคนิคในการจำแนกประเภทข้อมูลในงานวิจัยนี้เพื่อ เพิ่มประสิทธิภาพในการจำแนกผลิตภัณฑ์ที่เสียหายได้

รายการอ้างอิง

Alhammady, H., and Ramamohanarao, K. (2004). The Application of Emerging Patterns for Improving the Quality of Rareclass Classification. **In Proceedings of PKDD**, pp:207-211.

Alibeigi M., Hashemi S., and Hamzeh A. (2012). DBFS: An effective Density Based Feature Selection scheme for small sample size and high dimensional imbalanced data sets. **Data & Knowledge Engineering**, 81-82:67-103.

Antonelli, M., Ducange, P., and Marcelloni, F. (2014). An experimental study on evolutionary fuzzy classifiers designed for managing imbalanced datasets. **Neurocomputing**, 146:125-136.

Barandela, R., Sánchez, J.S., García, V., and Rangel,E. (2003). Strategies for learning in class imbalance problems. **Pattern Recognition**, 36(3):849-851.

Batuwita, R., and Palade, V. (2010). Efficient resampling methods for training support vector machines with imbalanced datasets. **In Neural Networks (IJCNN), The 2010 International Joint Conference on IEEE**, pp1-8.

Cateni, S., Colla, V., and Vannucci, M. (2014). A method for resampling imbalanced datasets in binary classification tasks for real-world problems. **Neurocomputing**, 135:32-41.

Chawla, N.V. (2002). Data Mining for Imbalanced Datasets: An Overview. **The Data Mining and Knowledge Discovery Handbook**, p853-867.

Chawla, N.V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. (2002). SMOTE: synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, 16:321–357.

Chawla, N. V., Japkowicz, N., amd Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. **ACM Sigkdd Explorations Newsletter**, 6(1):1-6.

Chistianini, N., and Shawe-Taylor J. (2000). An introduction to support vector machines, and other kernel-based learning methods. **Cambridge University Press**.

Cortes C, and Vapnik V. (1995). Support vector network. **Machine Learning**, 20(3):273–297.

Datta, S., and Das, S. (2015). Near-Bayesian Support Vector Machines for imbalanced data classification with equal or unequal misclassification costs. **Neural Networks**, 70:39-52.

Estabrooks, A., and Japkowicz, N. (2001). A mixture-of-experts framework for learning from unbalanced data sets. **In Proceedings of the 2001 Intelligent Data Analysis Conference**, pp34-43.

Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A Multiple Resampling Method for Learning from Imbalanced Data Sets. **Computational Intelligence,** 20:18-36.

Fan, W., Stolfo, S. J., Zhang, J., and P. K. Chan. (1999). AdaCost: misclassification cost-sensitive boosting. **In Proceedings of the Sixteenth International Conference on Machine Learning**, pp99-105.

Farquad, M. A. H., and Bose, I. (2012). Preprocessing unbalanced data using support vector machine. **Decision Support Systems**, 53(1):226-233.

Fischer, A., Suen, Y.C., Frinken, V., Riesen, K., and Bunke, H. (2015). Approximation of graph edit distance based on Hausdorff matching. **Pattern Recognition**, 48:331-343

Gao, M., Hong, X., Chen, S., Harris, J.C., and Khalaf, E. (2014). PDFOS: PDF estimation based over-sampling for imbalanced two-class problems. **Neurocomputing**, 138:248-259.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. **Machine learning**, 46(1-3):389-422

He, H., and Garcia, E. A. (2009). Learning from imbalanced data. **IEEE Transactions on Knowledge and Data Engineering**, 21(9):1263-1284.

He, H., and Ghodsi, A. (2010). Rare class classification by support vector machine. **In Pattern Recognition (ICPR), 2010 20th International Conference on IEEE**, 548-551.

Japkowicz, N., and Stephen, S. (2002). The class imbalance problem: A systematic study. **Intelligent Data Analysis**, 6(5):203-231.

Jayadeva, Khemchandani, R., and Chandra, S. (2007). Twin support vector machine for pattern classification. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 29(5):905–910.

Jo, T., and Japkowicz, N. (2004). Class imbalances versus small disjuncts. **ACM Sigkdd Explorations Newsletter**, 6(1):40-49.

Kong, X., Liu, X., Shi, R., and Lee, K. Y. (2015). Wind speed prediction using reduced support vector machines with feature selection. **Neurocomputing**, 169:449-456.

Kotsiantis, S., and Pintelas, P. (2003). Mixture of Expert Agents for Handling Imbalanced Data Sets, Annals of Mathematics. **Computing & TeleInformatics**, 46-55.

Kotsiantis, S., Kanellopoulos, D., and Pintelas, P. (2006). Handling imbalanced datasets: A review. **GESTS International Transactions on Computer Science and Engineering**, 30:25-36.

Lee, S., Lim, S.J., Kim, J., Yang, J., and Lee, Y. (2014). Classification of normal and epileptic seizure EEG signals using wavelet transform, phase-space reconstruction, and Euclidean distance. **Computer Methods and Programs in Biomedicine**, 116:10-25.

Li, K., Kong, X., Lu, Z., Wenyin, L., and Yin, J. (2014). Boosting weighted ELM for imbalanced learning. **Neurocomputing**, 128:15-21.

Manevitz, L., and Yousef, M. (2007). One-class document classification via neural networks. **Neurocomputing**, 70(7):1466-1481.

Muller, KR., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. (2001). An Introduction to kernel-based learning algorithms. **IEEE Trans Neural Networks**, 12(2):199–222.

Peng, X., and Xu, D. (2012). Twin Mahalanobis distance-based support vector machines for pattern recognition. **Information Sciences**, 200:22–37.

Piyanoot, V., Suwanna, R., Krisana, C., and Chidchanok, L. (2015). Improving classification rate constrained to imbalanced data between overlapped and non-overlapped regions by hybrid algorithms. **Neurocomputing**, 153:429-443.

Pratik, C. P. and Upasna, S. (2013). A Novel Classification Model for Data Theft Detection Using Advanced Pattern Mining. **Digital Investigation**, 10:385-97.

Scholkopf, B., Burges, C., and Smola, A. (1999). Advances in kernel methods—support vector learning. **MIT Press**.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., and Napolitano, A. (2010). RUSBoost: A hybrid approach to alleviating class imbalance. Systems, Man and Cybernetics, **IEEE Transactions on Systems and Humans**, 40(1):185-197.

Shao, Y., Zhang C., Wang, X., and Deng, N. (2011). Improvements on twin support vector machines. **IEEE Transactions on Neural Networks**, 22(6):962-968.

Tomar, D., and Agarwal, S. (2015). Twin Support Vector Machine: A review from 2007 to 2014. **Egyptian Informatics Journal**, 16:55-69.

Vluymans, S., Tarragó, D. S., Saeys, Y., Cornelis, C., & Herrera, F. (2015). Fuzzy rough classifiers for class imbalanced multi-instance data. **Pattern Recognition**, 53:36-45.

Wang, B. X., and Japkowicz, N. (2010). Boosting support vector machines for imbalanced data sets. **Knowledge and Information Systems**, 25(1):1-20.

Weiss, G. M., and Provost, F. (2003). Learning when training data are costly: the effect of class distribution on tree induction. **Journal of Artificial Intelligence Research**, 19:315-354.

Wu, J., Xiong, H., Wu, P., and Chen, J. (2007). Local decomposition for rare class analysis. **In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining**, pp:814-823.

Yang, W. H., Dai, D. Q., and Yan, H. (2008). Feature extraction and uncorrelated discriminant analysis for high-dimensional data. **IEEE Transactions on Knowledge and Data Engineering**, 20(5):601-614.

Yong, Y. (2012). The Research of Imbalanced Data Set of Sample Sampling Method Based on K-Means Cluster and Genetic Algorithm. **Energy Procedia**, 17:164-170.

Zhang, H., and Li, M. (2014). RWO-Sampling: A random walk over-sampling approach to imbalanced data classification. **Information Fusion**, 20:99-116.

Zheng, Z., Wu, X., and Srihari. R. (2004). Feature selection for text categorization on imbalanced data. **SIGKDD Explorations**, 6(1):80-89.

ภาคผนวก ก

รหัสต้นฉบับของโปรแกรมที่ใช้ในวิทยานิพนธ์

```
function [Xtr, Xts, ytr, yts] = imbalanced( X, y, MulVal)
 close all;
clc;
%% ทำการแบ่งข้อมูล train 70% test 30%
[ Xtr, Xts, ytr, yts ] = TrainTestSplit( X, y, 0.7);
%% ทำการจำแนกประเภทข้อมูลด้วยวิธีวัดแบบต่าง ๆ
DistMethod = ['euclidean'];
overlap( Xtr, Xts, ytr, yts, DistMethod, 1);
DistMethod = [cityblock];
overlap( Xtr, Xts, ytr, yts, DistMethod, 1);
DistMethod = ['mahalanobis'];
overlap( Xtr, Xts, ytr, yts, DistMethod, 1);
end

function [ Xtr, Xts, ytr, yts ] = TrainTestSplit( X, y, TrainPercen)
%% กำหนดคลาสเป้าหมาย
Class = unique(y);
yClassA = strcmp(y,Class(1));
yClassB = strcmp(y,Class(2));
disp(['Instances == ',num2str(size(X,1))]);
disp(['NumClass == ',num2str(size(Class,1))]);
disp(['Class : ',char(Class(1)),' have ',num2str(sum(yClassA))]);
disp(['Class : ',char(Class(2)),' have ',num2str(sum(yClassB))]);
XcA = X(yClassA,:);
XcB = X(yClassB,:);
ycA = y(yClassA);
ycB = y(yClassB);
```

```
%% แบ่งข้อมูล train และ test ของคลาสที่ 1
indA = randperm(length(ycA));
splA = round(length(ycA)*TrainPercen);
lenA = length(ycA);
XtrA = XcA(indA(1:splA), :);
XtsA = XcA(indA(splA+1:lenA), :);
ytrA = ycA(indA(1:splA));
ytsA = ycA(indA(splA+1:lenA));

%% แบ่งข้อมูล train และ test ของคลาสที่ 2
indB = randperm(length(ycB));
splB = round(length(ycB)*TrainPercen);
lenB = length(ycB);
XtrB = XcB(indB(1:splB), :);
XtsB = XcB(indB(splB+1:lenB), :);
ytrB = ycB(indB(1:splB));
ytsB = ycB(indB(splB+1:lenB));

%% รวมข้อมูลทั้งสองคลาสเป็นชุดข้อมูล Train และ Test
Xtr = [XtrA; XtrB];
Xts = [XtsA; XtsB];
ytr = [ytrA; ytrB];
yts = [ytsA; ytsB];

disp(['Train : ',num2str(size(Xtr,1)),' ',char(Class(1)),':',num2str(size(XtrA,1)),' |
',char(Class(2)),':',num2str(size(XtrB,1))]);
disp(['Test : ',num2str(size(Xts,1)),' ',char(Class(1)),':',num2str(size(XtsA,1)),' |
',char(Class(2)),':',num2str(size(XtsB,1))]);
end
```

```
function [  ] = overlapV1( Xtr, Xts, ytr, yts, DistMethod, MulVal)
disp(['Distance Method == ',DistMethod]);
disp(['MulVal == ', num2str(MulVal)]);
%% กำหนดคลาสในฟังก์ชัน
Class = unique(ytr);
yClassA = strcmp(ytr,Class(1));
yClassB = strcmp(ytr,Class(2));
yClassAts = strcmp(yts,Class(1));
yClassBts = strcmp(yts,Class(2));

%% กำหนดข้อมูล Train
XcA = Xtr(yClassA,:);
XcB = Xtr(yClassB,:);
ycA = ytr(yClassA);
ycB = ytr(yClassB);

%% กำหนดข้อมูล Test
XcAts = Xts(yClassAts,:);
XcBts = Xts(yClassBts,:);
ycAts = yts(yClassAts);
ycBts = yts(yClassBts);

%% คำนวนข้อมูลที่ซ้อนทับและข้อมูลที่ไม่ซ้อนทับบนข้อมูล Train
%% คำนวนการซ้อนทับในคลาส B
pointXtrA = mean(XcA);
calDistA = quantile(sort(pdist2(XcA,pointXtrA,DistMethod)),0.75);
DistA = calDistA*MulVal;
DistBonA = pdist2(XcB,pointXtrA,DistMethod);
indPureB = DistBonA > DistA;
XPureB = XcB(indPureB,:);
```

```
yPureB = ycB(indPureB,:);
XOverB = XcB(~indPureB,:);
yOverB = ycB(~indPureB,:);

%% คำนวนการซ้อนทับในคลาส A
pointXtrB = mean(XcB);
calDistB = quantile(sort(pdist2(XcB,pointXtrB,DistMethod)),0.75);
DistB = calDistB*MulVal;
DistAonB = pdist2(XcA,pointXtrB,DistMethod);
indPureA = DistAonB > DistB;
XPureA = XcA(indPureA,:);
yPureA = ycA(indPureA,:);
XOverA = XcA(~indPureA,:);
yOverA = ycA(~indPureA,:);

%% ข้อมูลที่ซ้อนทับกันและไม่ซ้อนทับกันของทั้งสองคลาสของข้อมูล Train
XPureTr = [XPureA; XPureB];
XOverTr = [XOverA; XOverB];
yPureTr = [yPureA; yPureB];
yOverTr = [yOverA; yOverB];

%% คำนวนข้อมูลที่ซ้อนทับและข้อมูลที่ไม่ซ้อนทับบนข้อมูล Test
%% คำนวนการซ้อนทับในคลาส B
DistBonAts = pdist2(XcBts,pointXtrA,DistMethod);
indPureBts = DistBonAts > DistA;
XPureBts = XcBts(indPureBts,:);
yPureBts = ycBts(indPureBts,:);
XOverBts = XcBts(~indPureBts,:);
yOverBts = ycBts(~indPureBts,:);
```

```
%% คำนวนการซ้อนทับในคลาส A
DistAonBts = pdist2(XcAts,pointXtrB,DistMethod);
indPureAts = DistAonBts > DistB;
XPureAts = XcAts(indPureAts,:);
yPureAts = ycAts(indPureAts,:);
XOverAts = XcAts(~indPureAts,:);
yOverAts = ycAts(~indPureAts,:);

%% ข้อมูลที่ซ้อนทับกันและไม่ซ้อนทับกันของทั้งสองคลาสของข้อมูล Test
XPureTs = [XPureAts; XPureBts];
XOverTs = [XOverAts; XOverBts];
yPureTs = [yPureAts; yPureBts];
yOverTs = [yOverAts; yOverBts];

%% ทำการสร้างโมเดลและแสดงผลลัพธ์
nCa = sum(yClassA);
nCb = sum(yClassB);
if (nCa < nCb) choseC = 1; else choseC = 2; end;
printResult( XPureTr, XOverTr, yPureTr, yOverTr, XPureTs, XOverTs, yPureTs, yOverTs,
choseC );
end
```

```
%% ฟังก์ชันที่ใช้สร้างโมเดลและแสดงผลลัพธ์
function [] = printResult( XPure, XOver, yPure, yOver, XPure1, XOver1, yPure1, yOver1,
choseC )
mi = statset('MaxIter',500000);

%% สร้างโมเดลด้วย SVM Linear Kernel
disp ('############ Linear ############');
disp ('------------ Over ------------');

%% สร้างโมเดลด้วยข้อมูลที่ซ้อนทับกัน
cl = svmtrain(XOver,yOver,'Kernel_Function', 'linear','option',mi);
predict = svmclassify(cl,XOver);
predict = svmclassify(cl,XOver1);
cfmts = confusionmat(yOver1,predict);
tmp1 = cfmts;
tst = cfmts(1,1)+cfmts(2,2);
tsf = cfmts(1,2)+cfmts(2,1);
accts = tst/(tst+tsf);
disp ('Test Metric');
disp ([num2str(cfmts)]);
disp (['Test accuracy = ',num2str(tst),' or ',num2str(round(accts*10000)/100),'%']);
disp (['TP Rate Class 1 = ',num2str(cfmts(1,1)),' or
',num2str(round((cfmts(1,1)/(cfmts(1,1)+cfmts(1,2)))*10000)/100),'%']);
disp (['TP Rate Class 2 = ',num2str(cfmts(2,2)),' or
',num2str(round((cfmts(2,2)/(cfmts(2,1)+cfmts(2,2)))*10000)/100),'%']);
disp (' ');
%% สร้างโมเดลด้วยข้อมูลที่ไม่ซ้อนทับกัน
disp ('------------ Pure ------------');
if (length(unique(yPure))>1 && ~isempty(yPure1))
  cl = svmtrain(XPure,yPure,'Kernel_Function', 'linear','option',mm);
```

```
    predict = svmclassify(cl,XPure);
    predict = svmclassify(cl,XPure1);
    if (length(unique(yPure1))>1)
        cfmts = confusionmat(yPure1,predict);
        tmp1 = tmp1 + cfmts;
        tst = cfmts(1,1)+cfmts(2,2);
        tsf = cfmts(1,2)+cfmts(2,1);
        accts = tst/(tst+tsf);
        disp ('Test Metric');
        disp ([num2str(cfmts)]);
        disp (['Test accuracy = ',num2str(tst),' or ',num2str(round(accts*10000)/100),'%']);
        disp (' ');
        disp ('$$$$$ Summary $$$$$');
        disp ([num2str(tmp1)]);
        if (choseC == 1)
        disp (['TP Rate Class 1 = ',num2str(tmp1(1,1)),' or
',num2str(round((tmp1(1,1)/(tmp1(1,1)+tmp1(1,2)))*10000)/100),'%']);
        else
        disp (['TP Rate Class 2 = ',num2str(tmp1(2,2)),' or
',num2str(round((tmp1(2,2)/(tmp1(2,1)+tmp1(2,2)))*10000)/100),'%']);
        end
        Measure(tmp1,choseC);
        disp (' ');
    else
        accts = sum(strcmp(yPure1,predict))/length(yPure1);
        disp (['Test accuracy = ',num2str(tst),' or ',num2str(round(accts*10000)/100),'%']);
    end
else
    if isempty(yPure1)
        disp('Null Pure test');
```

```
   else
      disp(['Acc Class',unique(yPure),' = ',[num2str(sum(strcmp(unique(yPure),yPure1))),' from
',num2str(length(yPure1))]]);
   end
   Measure(tmp1,choseC);
end

%% สร้างโมเดลด้วย SVM Polynimial Kernel
disp ('############ Polynomial #############');

%% สร้างโมเดลด้วยข้อมูลที่ซ้อนทับกัน
disp ('------------ Over ------------');
cl = svmtrain(XOver,yOver,'Kernel_Function', 'polynomial','option',mm);
predict = svmclassify(cl,XOver);
predict = svmclassify(cl,XOver1);
cfmts = confusionmat(yOver1,predict);
tmp1 = cfmts;
tst = cfmts(1,1)+cfmts(2,2);
tsf = cfmts(1,2)+cfmts(2,1);
accts = tst/(tst+tsf);
disp ('Test Metric');
disp ([num2str(cfmts)]);
disp (['Test accuracy = ',num2str(tst),' or ',num2str(round(accts*10000)/100),'%']);
disp (['TP Rate Class 1 = ',num2str(cfmts(1,1)),' or
',num2str(round((cfmts(1,1)/(cfmts(1,1)+cfmts(1,2)))*10000)/100),'%']);
disp (['TP Rate Class 2 = ',num2str(cfmts(2,2)),' or
',num2str(round((cfmts(2,2)/(cfmts(2,1)+cfmts(2,2)))*10000)/100),'%']);
disp (' ');
```

```
%% สร้างโมเดลด้วยข้อมูลที่ไม่ซ้อนทับกัน
disp ('------------ Pure ------------');
if (length(unique(yPure))>1 && ~isempty(yPure1))
   cl = svmtrain(XPure,yPure,'Kernel_Function', 'polynomial','option',mm);
   predict = svmclassify(cl,XPure);
   predict = svmclassify(cl,XPure1);
   if (length(unique(yPure1))>1)
      cfmts = confusionmat(yPure1,predict);
      tmp1 = tmp1 + cfmts;
      tst = cfmts(1,1)+cfmts(2,2);
      tsf = cfmts(1,2)+cfmts(2,1);
      accts = tst/(tst+tsf);
      disp ('Test Metric');
      disp ([num2str(cfmts)]);
      disp (['Test accuracy = ',num2str(tst),' or ',num2str(round(accts*10000)/100),'%']);
      disp (' ');
      disp ('$$$$$ Summary $$$$$');
      disp ([num2str(tmp1)]);
      if (choseC == 1)
      disp (['TP Rate Class 1 = ',num2str(tmp1(1,1)),' or
',num2str(round((tmp1(1,1)/(tmp1(1,1)+tmp1(1,2)))*10000)/100),'%']);
      else
      disp (['TP Rate Class 2 = ',num2str(tmp1(2,2)),' or
',num2str(round((tmp1(2,2)/(tmp1(2,1)+tmp1(2,2)))*10000)/100),'%']);
      end
      Measure(tmp1,choseC);
      disp (' ');
   else
      accts = sum(strcmp(yPure1,predict))/length(yPure1);
      disp (['Test accuracy = ',num2str(tst),' or ',num2str(round(accts*10000)/100),'%']);
```

```
  end
else
  if isempty(yPure1)
    disp('Null Pure test');
  else
    disp(['Acc Class',unique(yPure),' = ',[num2str(sum(strcmp(unique(yPure),yPure1))),' from
',num2str(length(yPure1))]]);
  end
  Measure(tmp1,choseC);
End

%% สร้างโมเดลด้วย SVM Radial Basis Function Kernel
disp ('############ RBF ############');

%% สร้างโมเดลด้วยข้อมูลที่ซ้อนทับกัน
disp ('------------ Over ------------');
cl = svmtrain(XOver,yOver,'Kernel_Function', 'rbf');
predict = svmclassify(cl,XOver);
predict = svmclassify(cl,XOver1);
cfmts = confusionmat(yOver1,predict);
tmp1 = cfmts;
tst = cfmts(1,1)+cfmts(2,2);
tsf = cfmts(1,2)+cfmts(2,1);
accts = tst/(tst+tsf);
disp ('Test Metric');
disp ([num2str(cfmts)]);
disp (['Test accuracy = ',num2str(tst),' or ',num2str(round(accts*10000)/100),'%']);
disp (['TP Rate Class 1 = ',num2str(cfmts(1,1)),' or
',num2str(round((cfmts(1,1)/(cfmts(1,1)+cfmts(1,2)))*10000)/100),'%']);
disp (['TP Rate Class 2 = ',num2str(cfmts(2,2)),' or
```

```
',num2str(round((cfmts(2,2)/(cfmts(2,1)+cfmts(2,2)))*10000)/100),'%']);
disp (' ');

%% สร้างโมเดลด้วยข้อมูลที่ไม่ซ้อนทับกัน
disp ('------------ Pure ------------');
if (length(unique(yPure))>1 && ~isempty(yPure1))
   cl = svmtrain(XPure,yPure,'Kernel_Function', 'rbf');
   predict = svmclassify(cl,XPure);
   predict = svmclassify(cl,XPure1);
   if (length(unique(yPure1))>1)
      cfmts = confusionmat(yPure1,predict);
      tmp1 = tmp1 + cfmts;
      tst = cfmts(1,1)+cfmts(2,2);
      tsf = cfmts(1,2)+cfmts(2,1);
      accts = tst/(tst+tsf);
      disp ('Test Metric');
      disp ([num2str(cfmts)]);
      disp (['Test accuracy = ',num2str(tst),' or ',num2str(round(accts*10000)/100),'%']);
      disp (' ');
      disp ('$$$$$ Summary $$$$$');
      disp ([num2str(tmp1)]);
      if (choseC == 1)
      disp (['TP Rate Class 1 = ',num2str(tmp1(1,1)),' or
',num2str(round((tmp1(1,1)/(tmp1(1,1)+tmp1(1,2)))*10000)/100),'%']);
      else
      disp (['TP Rate Class 2 = ',num2str(tmp1(2,2)),' or
',num2str(round((tmp1(2,2)/(tmp1(2,1)+tmp1(2,2)))*10000)/100),'%']);
      end
      Measure(tmp1,choseC);
      disp (' ');
```

```
  else
    accts = sum(strcmp(yPure1,predict))/length(yPure1);
    disp (['Test accuracy = ',num2str(tst),' or ',num2str(round(accts*10000)/100),'%']);
  end
else
  if isempty(yPure1)
    disp('Null Pure test');
  else
    disp(['Acc Class',unique(yPure),' = ',[num2str(sum(strcmp(unique(yPure),yPure1))),' from
',num2str(length(yPure1))]]]);
  end
  Measure(tmp1,choseC);
end
end
```

ภาคผนวก ข

บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

Kittipong Chomboon, Nuntawut Kaoungku, Nittaya Kerdprasop and Kittisak Kerdprasop (2014). **Data Mining in Semantic Web Data.** International Journal of Computer Theory and Engineering, Vol. 6, No. 6, p472-475.

Kittipong Chomboon, Pasapitch Chujai, Pongsakorn Teerarassamee, Kittisak Kerdprasop and Nittaya Kerdprasop (2015). **Ensemble Learning For Imbalanced Data Classification Problem.** ICIAE'2015 the 3rd International Conference on Industrial Application Engineering 2015, The Institute of Industrial Applications Engineers, Japan. 28-31 March 2015.

International Journal of Computer Theory and Engineering, Vol. 6, No. 6, December 2014

Data Mining in Semantic Web Data

K. Chomboon, N. Kaoungku, K. Kerdprasop, and N. Kerdprasop

***Abstract*—This research aims at studying the data mining role in semantic web data. Semantic web is popular in a variety of different applications, but research in data mining in semantic web data, appears less. As open source software for data mining in semantic web open source is minimal, and data model of the semantic web requires RDF or OWL format. These specific formats cannot be used directly in most data mining tools. We thus propose a methodology to mine data that appear in an RDF format. The mining process has been demonstrated through the use of R packages.**

***Index Terms*—Data mining, semantic web, R language.**

I. INTRODUCTION

Current data is not stored on a single computer, because the current is the era of information technology and social media, data can be stored in many computers on the internet, is difficult for them to access data quickly and easily. The researchers presented the technology to help manage these data called semantic web. The data in the format or the same specification as RDF/XML, N3, Turtle, N-Triples and OWL.

Semantic web [1], [2] has been used in various fields such as Information Systems, Search Engine etc. Large data technology to handle with this is data mining, because the large data analyzed find patterns or relationships of data is an advantage of data mining. Research in the field of data mining in semantic web data is not yet widely, since there is a management tool for data mining of semantic web is less, and data from the semantic web is stored in a format that cannot be used directly in data mining. The research in data mining has appeared very little.

Research in the field of data mining in semantic web data applied to various algorithms of data mining, such as data classification, association rule mining etc. Most research using the licensed software such as Microsoft Data Mining Extension (DMX) which is Microsoft SQL Server.

From the above it can be seen that the present data are not stored on a single computer always, is difficult to put that information in the internet is analyzed find patterns or relationships with the data mining. This research has proposed methods for data mining in semantic web data.

II. BACKGROUND

A. *Semantic Web*

Semantic web, have been developed since the storage is only human to understand the meaning, but the machine cannot understand it, because data without structure. Semantic web has been developed to provide useful data on the Internet that can be analyzed and applied to various tasks. The language used for defining the data structure is RDF [3]-[5] (Resource Description Framework). Which is written in the form of sentences consists of the subject, predicate and objects show in Fig. 1.

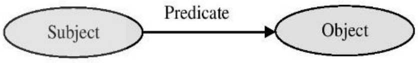


Fig. 1. RDF triples.

The standardization for semantic web in the context of web 3.0 shows in Fig. 2.

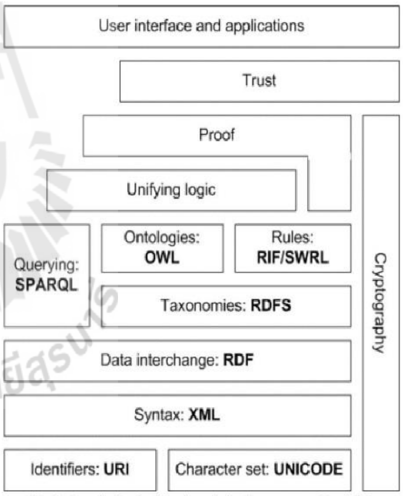


Fig. 2. Standard of semantic web in the context of web 3.0.

The components of semantic web are as follows:

- XML stands for Extensible Markup Language: XML is a markup language much like HTML, but XML was designed to transport and store data, not to display data. XML provides an elemental syntax for content structure within documents.
- XML Schema: XML Schema is a language for providing and restricting the structure and content of elements contained within XML documents.
- RDF stands for Resource Description Framework: RDF is a language for expressing data models. RDF was designed to provide a common way to describe information so it can be read and understood by computer applications.
- RDF Schema: RDF Schema extends RDF and is a

Manuscript received December 13, 2013; revised March 14, 2014. This work was supported in part by grant from Suranaree University of Technology through the funding of Data Engineering Research Unit.

The authors are with the School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand (e-mail: chomboon.k@gmail.com).

DOI: 10.7763/IJCTE.2014.V6.912

vocabulary for describing properties and classes of RDF-based resources.

- OWL stands for Web Ontology Language: OWL was designed to provide a common way to process the content of web information. OWL and RDF are much of the same thing, but OWL is a stronger language with greater machine interpretability than RDF. OWL comes with a larger vocabulary and stronger syntax than RDF.
- SPARQL: SPARQL is a protocol and query language for semantic web data sources.

B. XML

XML stands for Extensible Markup Language. XML was designed to carry data, not to display data. Tags are not predefined. The components of XML are as follows:

1) Tree structure

XML documents must contain a root element. This element is "the parent" of all other elements. All elements can have sub elements. The example shows as Fig. 3.

```
<root>
<child>
<subchild>…….</subchild>
</child>
</root>
```

Fig. 3. Show tree structure of XML documents.

2) Syntax

The syntax rules of XML are very simple are shows as follows:

- All XML elements must have a closing tag:
  <p> This is a paragraph </p>
- XML tags are case sensitive:
  <Note>This is incorrect<note>
  <note>This is correct<note>
- XML elements must be properly nested:
  <i>This text is bold and italic</i>
- XML documents must have a root element:
  <book>
  <title>XML Manual</title>
  <price>25.00</price
  </book>
  The root element is <book>.
- XML attribute values must be quoted:
  <book category="MANUAL">
  <title>XML Manual</title>
  <price>25.00</price
  </book>
- White-space is Preserved in XML:
  Input = Hello   World
  HTML output: Hello World

With XML, the white-space in a document is not truncated.

- Entity References.
  There are 5 predefined entity references in XML.
  < <less than
  > >greater than
  & &ampersand
  ' 'apostrophe
  "" quotation mark

3) Elements

An XML document contains XML elements. An XML element is everything from (including) the element's start tag to (including) the element's end tag. Example shows as Fig. 4.

```
<bookstore>
<book category="CHILDREN">
<title>Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="WEB">
<title>Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
```

Fig. 4. Show elements of XML documents.

In the Fig. 4. <bookstore> and <book> have element contents, because they contain other elements. <book> also has an attribute (category="CHILDREN"). <title>, <author>, <year>, and <price> have text content because they contain text.

XML elements must follow these naming rules:

- Names can contain letters, numbers, and other characters.
- Names cannot start with a number or punctuation character.
- Names cannot start with the letters xml (or XML, or Xml, etc).
- Names cannot contain spaces.

4) Namespace

XML Namespaces provide a method to avoid element name conflicts. Example show as follows:

- This XML carries book properties:
  <book>
  <weight>150</weight>
  <length>29.7</length>
  <width>21</width>
  </book>
- This XML carries book information:
  <book>
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
  </book>

If these XML fragments were added together, there would be a name conflict. Both contain a <book> element, but the elements have different content and meaning. An XML parser will not know how to handle these differences. Name conflicts in XML can easily be avoided using a name prefix show as follows:

<p:book>
<p:weight>150</p:weight>
<p:length>29.7</p:length>
<p:width>21</p:width>

```
</p:book>
<i:book>
< i:title>Harry Potter</ i:title>
< i:author>J K. Rowling</ i:author>
< i:year>2005</ i:year>
< i:price>29.99</ i:price>
</ i:book>
```

There will be no conflict because the two <book> elements have different names. When using prefixes in XML, a so-called namespace for the prefix must be defined. The namespace is defined by the xmlns attribute in the start tag of an element show as follows:

```
<root>
<p:book xmlns:p="http://www.ex.com/properties">
<p:weight>150</p:weight>
<p:length>29.7</p:length>
<p:width>21</p:width>
</p:book>
<i:book xmlns:i="http://www.ex.com/information">
< i:title>Harry Potter</ i:title>
< i:author>J K. Rowling</ i:author>
< i:year>2005</ i:year>
< i:price>29.99</ i:price>
</ i:book>
</root>
```

C. SPARQL Language

SPARQL [6] is a query language for the semantic web, which is format in RDF / XML or OWL. SPARQL language to access data through a Triple (Basic Graph Pattern) consists subject predicate and object. The main structure consists of a "SELECT" to define a variable to store the results of the query and "WHERE" as a condition for the query. Table I show example data and query with SPARQL language.

TABLE I: EXAMPLE DATA AND QUERY WITH SPARQL LANGUAGE

Data	**Query**

@prefix foaf:
<http://xmlns.com/foaf/0.1/> .
_:a foaf:nameb "Alice" .
_:a foaf:mbox
<mailto:Alice@example.com> .
_:b foaf:name "Bob" .
_:b foaf:mbox <mailto:
Bob@example.org> .
_:c foaf:name "Peter" .
	PREFIX foaf:
	<http://xmlns.com/foaf/0.1/>
	SELECT ?name
	WHERE {
	?x foaf:name ?name .
	}

Table I shows example data and query with sparql language are as follows:

- PREFIX is defines a resource that is defined in the head.
- SELECT is set the variable to store the results of query, by define variable need the "?" before variable name.
- WHERE is the conditions used for the query (e.g. ?x foaf:name ?name etc.)

D. R Language

R language [7], [8] is a functional and object-oriented language that was developed to replace the S language for statistical, developed in 1995 from the Department of Statistics, University of Auckland, New Zealand. R language has been applied to various fields, the data mining applied R language in the research, Because of strength of the R language for data mining is to analyze large data and open-source software. In this research study the data mining in semantic web data. The command in R language for data mining in semantic web data as follows.

Rrdf is a package on R language, handling triples on ontology within "RDF/XML" format. Now we use rrdf package to transformation data on "RDF/XML" to data frame format on R. Then we easily to use data on data frame format to mining.

- load.rdf(filename, format = "RDF/XML") is command to load data from file format RDF / XML.
- sparql.rdf(model, sparql) is command to query data with SPARQL language.

III. METHODOLOGY

In this section we present the process of data mining on semantic web dataset. We use R language for implementing our method. The overview our techniques show as Fig. 5.

Step 1: use library rrdf on R language to import dataset from RDF file as data frame

Step 2: use data on R language to classification.

Step 3: use model to predict data

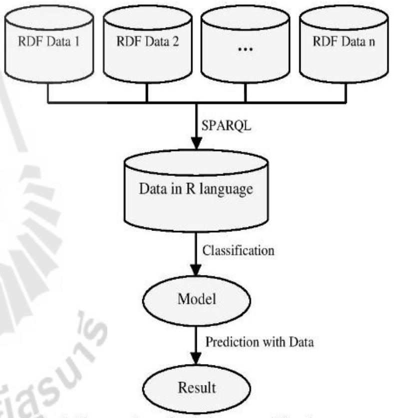


Fig. 5. The overview of techniques to mining dataset.

IV. EXPERIMENT RESULT

This research experimentation used Iris dataset from UCI Machine Learning Repository. Iris dataset has 5 attributes and 150 data instances.

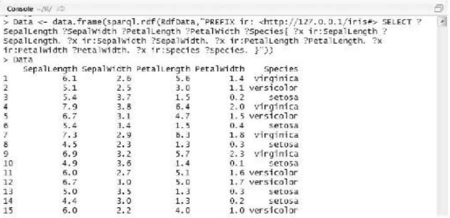
Step 1: use library rrdf on R language to import iris dataset from RDF file as data frame. We can use this command to load data from RDF file.

"RdfData <- load.rdf("iris.rdf")".

Then convert data to data frame can use this command.

```
"Data <- data.frame(sparql.rdf(RdfData,
  "PREFIX ir: <http://127.0.0.1/iris#>
  SELECT ?SepalLength ?SepalWidth
  ?PetalLength ?PetalWidth
  ?Species { ?x ir:SepalLength ?SepalLength.
  ?x ir:SepalWidth ?SepalWidth.
  ?x ir:PetalLength ?PetalLength.
  ?x ir:PetalWidth ?PetalWidth.
  ?x ir:Species ?Species.}))"
```

Convert iris data to data frame on R language (show in Fig. 6).

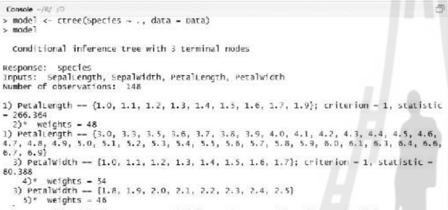


```
> Data <- data.frame(sparql.rdf(RdfData,"PREFIX ir: <http://127.0.0.1/iris#> SELECT ?
SepalLength ?SepalWidth ?PetalLength ?PetalWidth ?Species{ ?x ir:SepalLength ?
SepalLength. ?x ir:SepalWidth ?SepalWidth. ?x ir:PetalLength ?PetalLength. ?x
ir:PetalWidth ?PetalWidth. ?x ir:Species ?Species. }"))
> Data
   SepalLength SepalWidth PetalLength PetalWidth    Species
1          6.1        2.6         5.6        1.4  virginica
2          5.1        2.5         3.0        1.1 versicolor
3          5.4        3.7         1.5        0.2     setosa
4          7.9        3.8         6.4        2.0  virginica
5          6.7        3.1         4.7        1.5 versicolor
6          5.4        3.4         1.5        0.4     setosa
7          7.3        2.9         6.3        1.8  virginica
8          4.5        2.3         1.3        0.3     setosa
9          6.9        3.2         5.7        2.3  virginica
10         4.9        3.6         1.4        0.1     setosa
11         6.0        2.7         5.1        1.6 versicolor
12         6.7        3.0         5.0        1.7 versicolor
13         5.0        3.5         1.3        0.3     setosa
14         4.4        3.0         1.3        0.2     setosa
15         6.0        2.2         4.0        1.0 versicolor
```

Fig. 6. Convert iris data to data frame on R language.

Step 2: then use data on R language to classification. First we can generate model form data with this command.

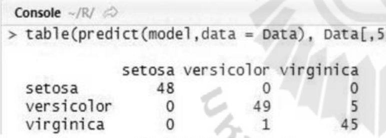
"model <- ctree(Species ~ ., data = Data)", use column Species to decision show in Fig. 7.



```
> model <- ctree(Species ~ ., data = Data)
> model

  Conditional inference tree with 3 terminal nodes

Response:  Species
Inputs:  SepalLength, SepalWidth, PetalLength, PetalWidth
Number of observations:  148

1) PetalLength == {1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.9}; criterion = 1, statistic
= 266.364
  2)*  weights = 48
1) PetalLength == {3.0, 3.3, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6,
4.7, 4.8, 4.9, 5.0, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6.0, 6.1, 6.3, 6.4, 6.6,
6.7, 6.9}
  3) PetalWidth == {1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7}; criterion = 1, statistic =
80.388
    4)*  weights = 54
  3) PetalWidth == {1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5}
    5)*  weights = 46
```

Fig. 7. Generate model from data use Species to decision.

Step 3: we use model to predict data and view the result of model with this command "table (predict(model, data = Data, Data[,5])", show in Fig. 8.



```
> table(predict(model,data = Data), Data[,5])

             setosa versicolor virginica
  setosa         48          0         0
  versicolor      0         49         5
  virginica       0          1        45
```

Fig. 8. Result of the model.

V. CONCLUSION

This research aims to study how to use dataset from semantic web in format RDF/XML and apply to mining with R language. Because R language is open source program, we can use library is already in R and easy to create function for mining data. We can use semantic web dataset or open linked dataset to improve performance of data mining.

REFERENCES

[1] P. Hitzler, M. Krötzsch, and S. Rudolph, "Foundations of semantic web technologies," *Textbooks in Computing*, Chapman and Hall/CRC Press, 2009.

[2] V. Nebot and R. Berlanga, "Finding association rules in semantic web data," *Knowledge-Based Systems*, vol. 25, no. 1, pp. 51-62, 2012.

[3] E. Willighagen. (2013). RRDF - support for the resource description framework. [Online]. Available: http://cran.r-project.org/web/packages/rrdf/rrdf.pdf

[4] W3C. (2004). Resource Description Framework (RDF): Concepts and abstract syntax. [Online]. Available: http://www.w3.org/TR/rdf-concepts/

[5] W3C. (2008). SPARQL query language for RDF. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[6] C. Kiefer, A. Bernstein, and A. Locher, "Adding data mining support to SPARQL via statistical relational learning methods," *The Semantic Web: Research and Applications*, vol. 5021, pp. 478-492, 2008.

[7] K. Kerdprasop. (2012). Data mining with R. [Online]. Available: https://sites.google.com/site/kittisakthailand55/home/datamining2-55

[8] E. Paradis, J. Claude, and K. Strimmer. "APE: Analyses of phylogenetics and evolution in R language," *Bioinformatics*, vol. 20, no. 2, pp. 289-290, 2004.

Kittipong Chomboon is currently a doctoral student with the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in computer engineering from Suranaree University of Technology, Thailand in 2012, and master degree in computer engineering from Suranaree University of Technology, Thailand in 2013. His current research includes ontology and classification.

Nuntawut Kaoungku is currently a doctoral student with the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in computer engineering from Suranaree University of Technology, Thailand in 2012, and master degree in computer engineering from Suranaree University of Technology, Thailand in 2013. His current research includes semantic web and association.

Nittaya Kerdprasop is an associate professor at the School of Computer Engineering, Suranaree University of Technology, Thailand. She received her bachelor degree in radiation techniques from Mahidol University, Thailand, in 1985, master degree in Computer Science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in computer science from Nova Southeastern University, U.S.A, in 1999. She is a member of ACM and IEEE Computer Society. Her research of interest includes knowledge discovery in databases, artificial intelligence, logic programming, and intelligent databases.

Kittisak Kerdprasop is an associate professor and chair of the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in mathematics from Srinakarinwirot University, Thailand in 1986, master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in computer science from Nova Southeastern University, U.S.A. in 1999. His current research includes data mining, artificial intelligence, functional and logic programming languages, computational statistics.

An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm

Kittipong Chomboon*, Pasapitch Chujai, Pongsakorn Teerarassamee, Kittisak Kerdprasop, Nittaya Kerdprasop

School of Computer Engineering, Institute of Engineering, Suranaree University of Technology,
Nakhorn Ratchasima 3000, Thailand.

*Corresponding Author: chomboon.k@gmail.com

Abstract

This research aims at studying the performance of k-nearest neighbor classification when applying different distance measurements. In this work, we comparatively study 11 distance metrics including Euclidean, Standardized Euclidean, Mahalanobis, City block, Minkowski, Chebychev, Cosine, Correlation, Hamming, Jaccard, and Spearman. A series of experimentations has been performed on eight synthetic datasets with various kinds of distribution. The distance computations that provide highly accurate prediction consist of City block, Chebychev, Euclidean, Mahalanobis, Minkowski, and Standardize Euclidean techniques.

Keywords: Data Classification, Synthetic Data, Distance Metrics, k-Nearest Neighbors.

1. Introduction

Data mining is the extraction of knowledge hidden in the data. Data mining is often done with the large datasets. The knowledge from data mining has been used in various fields, such as prediction over future situation, assisting in medical diagnosis, forecasting relation of chronology.

Current data mining methodology has been classified into several tasks, such as classification, clustering, and association mining. Data mining each for task will have a different purpose. Classification task will be trying to classify data with high accuracy for classifying future example, such as trying to distinguish between patients with heart disease and those who are healthy. Clustering task will try to categorize groups of data such that data in the same group look similar, whereas they are dissimilar to others in different groups. Association mining task will try to find rules that represent relation between data with some support and confident values.

Classification task of data mining can be done with many algorithms such as k-nearest neighbor. Beyer(1) explained the significance and origin of the nearest neighbor. Cover(2) used k-nearest neighbor to classify data. Dudani(3) did research about weighting of distance matrix values with k-nearest neighbor. Fukunaga(4) developed techniques for running k-nearest neighbor faster. Keller(5) developed new algorithm named "Fuzzy K-Nearest Neighbor" based on k-nearest neighbor with the purpose to use it with fuzzy task. Köhn(6) used city-block distance matric to increase performance of k-nearest neighbor algorithm.

This research also studies classification technique with a specific interest in the k-nearest algorithm. We aim to analyze the performance of different distance metrics to finally choose a proper metric that makes a good classification performance. In this research use 8 synthetic datasets with different distribution, and a dataset for each distribution has 2 classes but has different amount of data in each class. This is to test the impact about amount in each class on the performance of classification.

The rest of this research is organized as follows: Section 2 gives details of the k-Nearest Neighbor and the computation of each distance metric. Section 3 gives details of our proposed method. The experimental results and analysis will be presented in Section 4. Finally, the research is concluded in Section 5.

DOI: 10.12792/iciae2015.051

2. Background

2.1 k-Nearest Neighbor

The k-nearest neighbor is a semi-supervised learning algorithm such that it requires training data and a predefined k value to find the k nearest data based on distance computation. If k data have different classes, the algorithm predicts class of the unknown data to be the same as the majority class. For example, to find the appropriate class of new datum using the k-nearest neighbor algorithm with a Euclidean distance metric, the concept can be shown in Fig. 1.

Fig. 1 shows the classification of iris data. The point to be classified is (5, 1.45), which is shown with "X". When applying k-nearest neighbor algorithm with k = 8 using Euclidean distance computation, the result is shown with a radius of dot line. It has two possible classes: virginica class with two instances and versicolor class with six instances. This algorithm will classify mark "X" to the class of versicolor because versicolor class is the majority of data within the radius.

2.2 Distance Metrics

Distance metrics are a method to find distance between a new data point and existing training dataset. In this research, we experiment with 11 distance metrics, which can be explained as follows.

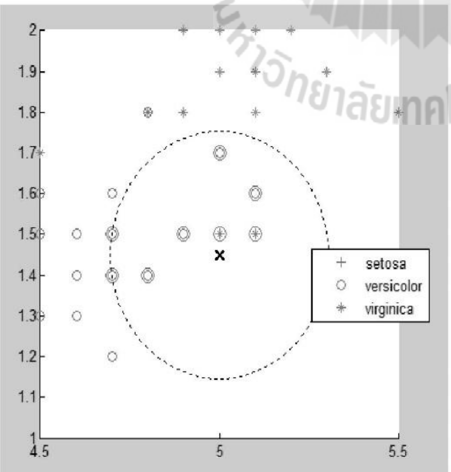


Fig. 1. The k-nearest neighbor prediction with k = 8.

Given an *mx*-by-*n* data matrix X, which is treated as *mx* (1-by-*n*) row vectors x_1, x_2, ..., xm_x, and *my*-by-*n* data matrix Y, which is treated as *my*(1-by-*n*) row vectors y_1, y_2,...,ym_y, the various distances between the vectors x_s and y_t are defined as follows:

1. Euclidean Distance

The Euclidean distance is a measure to find distance between two points, defined by Eq. (1)

$$d_{st}^2 = (x_s - y_t)(x_s - y_t)' \tag{1}$$

The Euclidean distance is a special case of the Minkowski metric, where p = 2.

2. Standardized Euclidean Distance

The standardized Euclidean distance is used to optimize the problem of finding the distance, defined by Eq. (2)

$$d_{st}^2 = (x_s - y_t)V^{-1}(x_s - y_t)' \tag{2}$$

where V is the n-by-n diagonal matrix whose jth diagonal element is $S(j)^2$, S is the vector containing the inverse weights.

3. Mahalanobis Distance

The Mahalanobis distance is a measure between a point and a distribution of data, defined by Eq. (3)

$$d_{st}^2 = (x_s - y_t)C^{-1}(x_s - y_t)' \tag{3}$$

where C is the covariance matrix.

4. City Block Distance

The city block distance between two points is the sum of the absolute difference of Cartesian coordinates, defined by Eq. (4)

$$d_{st} = \sum_{j=1}^{n} |x_{sj} - y_{tj}| \tag{4}$$

The city block distance is a special case of the Minkowski metric, where p = 1.

5. Minkowski Distance

The Minkowski distance is a method to find distance based on Euclidean space, defined by Eq. (5)

$$d_{st} = \sqrt[p]{\sum_{j=1}^{n} |x_{sj} - y_{tj}|^{p}} \qquad (5)$$

For the special case of Minkowski distance

p = 1, the Minkowski metric gives the city block distance,

p = 2, the Minkowski metric gives the Euclidean distance, and

p = ∞, the Minkowski metric gives the Chebychev distance.

6. Chebychev Distance

The Chebychev distance is a measure to find distance between two vectors or points with standard coordinates, defined by Eq. (6)

$$d_{st} = max_j\{|x_{sj} - y_{tj}|\} \qquad (6)$$

The Chebychev distance is a special case of the Minkowski metric, where p = ∞.

7. Cosine Distance

The Cosine distance is computed from one minus the cosine of the included angle between points, defined by Eq. (7)

$$d_{st} = \left(1 - \frac{x_s {y'}_t}{\sqrt{(x_s {x'}_s)(y_t {y'}_t)}}\right) \qquad (7)$$

8. Correlation Distance

Distance based on correlation is a measure of statistical dependence between two vectors, defined by Eq. (8)

$$d_{st} = \left(1 - \frac{(x_s - \bar{x}_s)(y_t - \bar{y}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'}\sqrt{(y_t - \bar{y}_t)(y_t - \bar{y}_t)'}}\right) \qquad (8)$$

where

$$\bar{x}_s = \frac{1}{n}\sum_j x_{sj}$$

$$\bar{y}_t = \frac{1}{n}\sum_j y_{tj}$$

9. Hamming Distance

Hamming distance, which is the percentage of coordinates that differ, can be defined by Eq. (9)

$$d_{st} = \left(\frac{\#(x_{sj} \neq y_{tj})}{n}\right) \qquad (9)$$

10. Jaccard Distance

Jaccard distance is computed from one minus the Jaccard coefficient, which is the percentage of nonzero coordinates that differ, defined by Eq. (10)

$$d_{st} = \left(\frac{\#\left[(x_{sj} \neq y_{tj}) \cap \left((x_{sj} \neq 0) \cup (y_{tj} \neq 0)\right)\right]}{\#\left[(x_{sj} \neq 0) \cup (y_{tj} \neq 0)\right]}\right) \qquad (10)$$

11. Spearman Distance

Spearman distance is computed from one minus the sample Spearman's ranked correlation between observations, defined by Eq. (11)

$$d_{st} = 1 - \frac{(r_s - \bar{r}_s)(r_t - \bar{r}_t)'}{\sqrt{(r_s - \bar{r}_s)(r_s - \bar{r}_s)'}\sqrt{(r_t - \bar{r}_t)(r_t - \bar{r}_t)'}} \qquad (11)$$

Where

r_{sj} is the rank of x_{sj} taken over x_1j, x_2j, $...xmx,j$.

r_{tj} is the rank of y_{tj} taken over y_1j, y_2j, $...ymy,j$.

r_s and r_t are the coordinate-wise rank vectors of x_s and y_t,

i.e., $r_s = (r_{s1}, r_{s2}, \ldots r_{sn})$ and $r_t = (r_{t1}, r_{t2}, \ldots r_{tn})$.

$$\bar{r}_s = \frac{1}{n}\sum_j r_{sj} = \frac{(n+1)}{2}$$

$$\bar{r}_t = \frac{1}{n}\sum_j r_{tj} = \frac{(n+1)}{2}$$

3. Empirical Study Methodology

In this section, we present our study framework using k-nearest neighbor algorithm with various distance metrics. The framework is shown in Fig. 2.

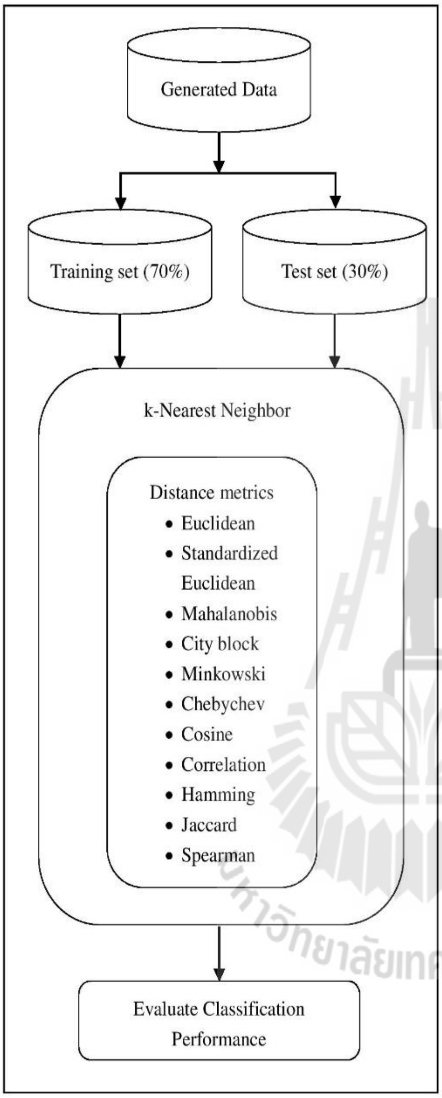


Fig. 2. The framework of our empirical study.

From Fig. 2 the detail of each step can be explained as follows:

Step 1: Generate binary data set with different distribution and different amount of data in each class. Then split data around 70% for training set and 30% for test set, which will be used for testing the performance of classification.

Step 2: Use data from step 1 for data classification by applying the k-nearest neighbor algorithm with various distance metrics to compute the k-nearest data points for making classification.

Step 3: Analyze the results and conclude about the performance of classification using various distance metrics.

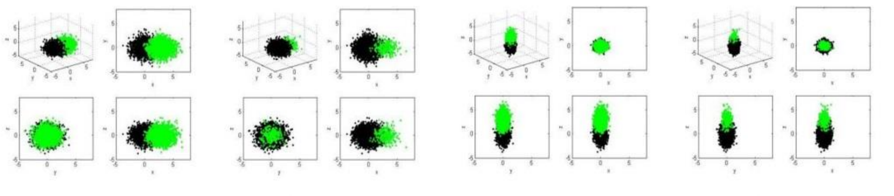
4. Experimental Results

4.1 Datasets

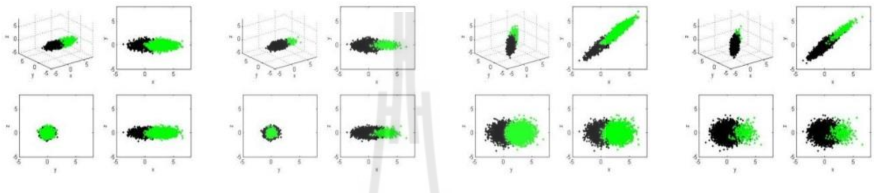
For our experiment, the proposed framework has been applied for classifying binary synthetic datasets. We generate eight synthetic datasets, each dataset has four different distributions, and each distribution has two of data in which class the amount of data in each class is varied. Each dataset has in total 5000 instances, and three features. We use MATLAB program to generate synthetic datasets. Details of the synthetic datasets are given in Table 1. Fig. 3 illustrates an overview of synthetic datasets.

Table 1. Details of synthetic datasets.

Dataset	Mean	SD	Class 1	Class 2	Total
1	[0 0 0; 3 0 0]	[1 0 0; 0 1 0; 0 0 1]	2500	2500	5000
2	[0 0 0; 3 0 0]	[1 0 0; 0 1 0; 0 0 1]	4750	250	5000
3	[0 0 0; 0 0 3]	[0.2 0 0; 0 0.2 0; 0 0 1]	2500	2500	5000
4	[0 0 0; 0 0 3]	[0.2 0 0; 0 0.2 0; 0 0 1]	4750	250	5000
5	[0 0 0; 3 0 0]	[1 0 0; 0 0.2 0; 0 0 0.2]	2500	2500	5000
6	[0 0 0; 3 0 0]	[1 0 0; 0 0.2 0; 0 0 0.2]	4750	250	5000
7	[0 0 0; 3 3 0]	[1 0.9 0; 0.9 1 0; 0 0 1]	2500	2500	5000
8	[0 0 0; 3 3 0]	[1 0.9 0; 0.9 1 0; 0 0 1]	4750	250	5000



(a) Dataset 1 (b) Dataset 2 (c) Dataset 3 (d) Dataset 4



(e) Dataset 5 (f) Dataset 6 (g) Dataset 7 (h) Dataset 8

Fig. 3. Distribution of the eight synthetic datasets, each one has four kinds of distribution.

4.2 Experimental Results

The results from the proposed study framework for eight synthetic datasets have been shown in Figs. 4 and 5. The data classification has been performed with the same algorithm (that is, k-Nearest Neighbor) and the same parameter setting. The only varied factor is a distance measurement. It turns out that the Hamming and Jaccard distance metrics perform badly on 4 out of 8 datasets.

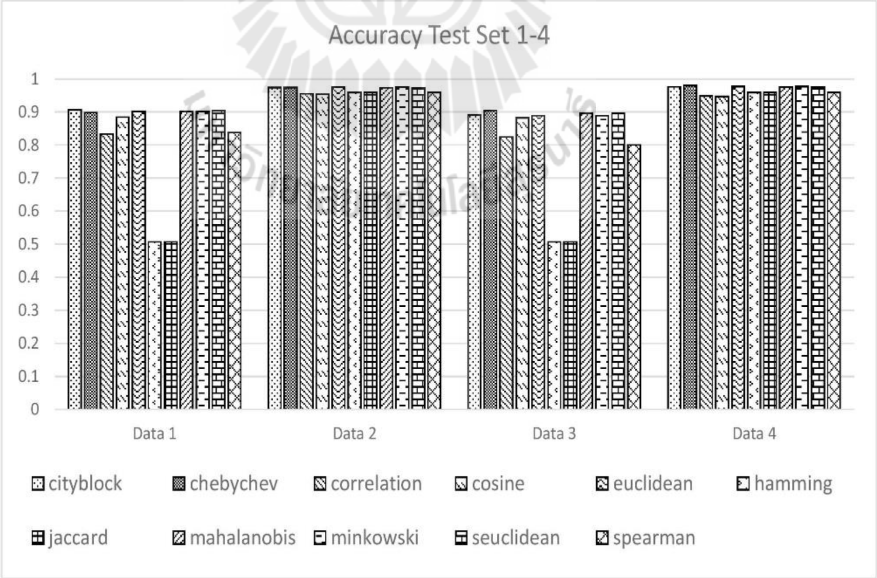


Fig. 4. Accuracy of synthetic datasets from no. 1 to no. 4.

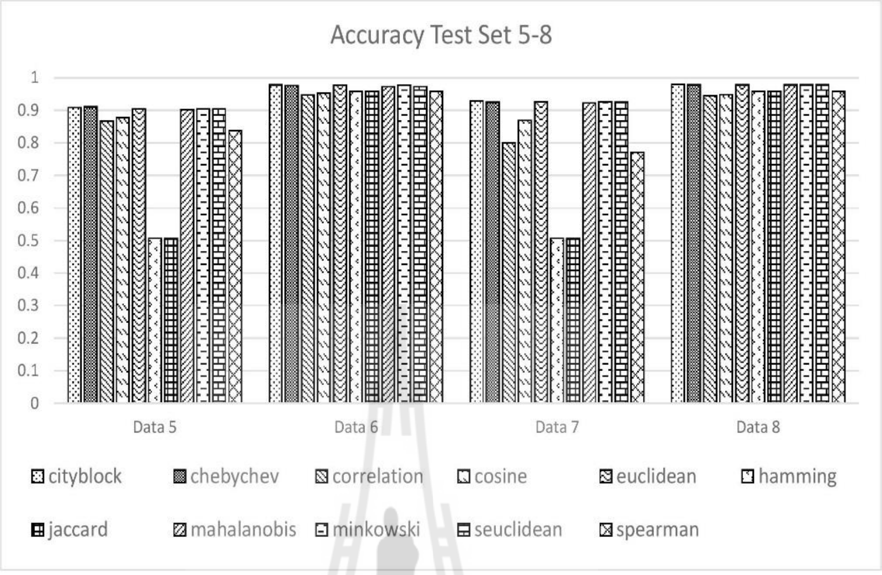


Fig. 5. Accuracy of synthetic datasets from no. 5 to no. 8.

5. Conclusions

The results of this research showed accuracy of k-nearest neighbor classification algorithm with different distance metrics. Experiments had been performed on eight synthetic datasets generated by MATLAB. The synthetic datasets have four distributions and have been split 70% to training set and 30% to test set. The results of classification over datasets in which amount of data in each class is equal showed that the Hamming and Jaccard techniques are low accuracy, while the other distance computation techniques have similar accuracy. The synthetic datasets in which amount of data in each class is different such as dataset 2, 4, 6 and 8 showed that the Hamming and Jaccard techniques are increasing in their classification accuracy. We can conclude that Hamming and Jaccard techniques are affected by the ratio of members in each class, while the other techniques are not affected by such phenomenon. The highest accuracy on classify data with k-Nearest Neighbor is obtained from the six distance metrics, that are City-block, Chebychev, Euclidean, Mahalanobis, Minkowski, and Standardized Euclidean techniques.

References

(1) Beyer Kevin, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft, When is "nearest neighbor" meaningful?, in Database Theory—ICDT'99. 1999, Springer. p. 217-235.aaaa

(2) Cover Thomas and Peter Hart: "Nearest neighbor pattern classification", Information Theory, IEEE Transactions on, Vol. 13, No. 1, pp. 21-27, 1967

(3) Dudani Sahibsingh A: "The distance-weighted k-nearest-neighbor rule", Systems, Man and Cybernetics, IEEE Transactions on, Vol. No. 4, pp. 325-327, 1976

(4) Fukunaga Keinosuke and Patrenahalli M Narendra: "A branch and bound algorithm for computing k-nearest neighbors", Computers, IEEE Transactions on, Vol. 100, No. 7, pp. 750-753, 1975

(5) Keller James M, Michael R Gray, and James A Givens: "A fuzzy k-nearest neighbor algorithm", Systems, Man and Cybernetics, IEEE Transactions on, Vol. No. 4, pp. 580-585, 1985

(6) Köhn Hans-Friedrich: "Combinatorial individual differences scaling within the city-block metric", Computational Statistics & Data Analysis, Vol. 51, No. 2, pp. 931-946, 2006

ประวัติผู้เขียน

นายกิตติพงศ์ ชมบุญ เกิดเมื่อวันที่ 6 กันยายน พ.ศ. 2532 ที่อำเภอบัวเชด จังหวัดสุรินทร์ เริ่มเข้าศึกษาระดับชั้นอนุบาล 1 ถึงชั้นประถมศึกษาปีที่ 5 ที่โรงเรียนบ้านสวาท อำเภอบัวเชด จังหวัดสุรินทร์ หลังจากนั้นได้ย้ายไปศึกษาต่อในระดับชั้นประถมศึกษาปีที่ 6 ที่โรงเรียนเมือง อำเภอเมือง จังหวัดสุรินทร์ จากนั้นได้เข้าศึกษาต่อในระดับมัธยมศึกษาตอนต้นและตอนปลาย ที่โรงเรียนสุรวิทยาคาร อำเภอเมือง จังหวัดสุรินทร์ ในปีการศึกษา 2551 ได้เข้าศึกษาต่อในระดับปริญญาตรีในสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่มหาวิทยาลัยเทคโนโลยีสุรนารี และสำเร็จการศึกษาในปีการศึกษา 2554 ภายหลังจากสำเร็จการศึกษาในระดับปริญญาตรี ได้เข้าศึกษาและสำเร็จการศึกษาระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปีการศึกษา 2555 หลังจากนั้นได้เข้าศึกษาระดับปริญญาเอก สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารีในปีการศึกษา 2556

ในระหว่างการศึกษาได้รับความอนุเคราะห์อย่างดีจากอาจารย์ที่ปรึกษาและอาจารย์ประจำวิชา Database Systems และได้รับความไว้วางใจให้เป็นผู้ช่วยสอนปฏิบัติการในรายวิชา Database Systems และ Knowledge Discovery and Data Mining หลังจากนั้นได้รับการตีพิมพ์เผยแพร่บทความวิจัยซึ่งรายละเอียดสามารถดูได้ที่ภาคผนวก ข