การประยุกต์ใช้ซัพพอร์ตเวคเตอร์แมชชีนแบบหนึ่งต่อหนึ่งบนข้อมูลแบบหลายฉลากโดยใช้สปาร์ค

นายสุทธิพงษ์ แดงด้วง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

Application of One-Versus-One Support Vector Machines to Classify Multi-Label Datasets Using Spark

Mr. Suthipong Daengduang

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2016

หัวข้อวิทยานิพนธ์	การประยุกต์ใช้ซัพพอร์ตเวคเตอร์แมชชีนแบบหนึ่งต่อหนึ่งบนข้อมูลแบบหลายฉลากโดยใช้สปาร์ค
โดย	นายสุทธิพงษ์ แดงด้วง
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ดร.พีรพล เวทีกูล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ดร.พีรพล เวทีกูล)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ)

.....กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.กฤษณะ ไวยมัย)

สุทธิพงษ์ แดงด้วง : การประยุกต์ใช้ซัพพอร์ตเวคเตอร์แมชชีนแบบหนึ่งต่อหนึ่งบนข้อมูลแบบหลายฉลากโดยใช้สปาร์ค (Application of One-Versus-One Support Vector Machines to Classify Multi-Label Datasets Using Spark) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ดร.พีรพล เวทีกูล, 62 หน้า.

การจำแนกข้อมูลแบบหลายฉลาก เป็นวิธีการที่มีการใช้ความรู้ที่มีอยู่ก่อนหน้าในการจำแนกข้อมูล โดยในหนึ่งตัวอย่างนั้นสามารถปรากฏได้ในหลายกลุ่มข้อมูล ในกรณีของวิธีซัพพอร์ตเวคเตอร์แมชชีน วิธีการจำแนกแบบหนึ่งต่อทั้งหมดนั้นเป็นที่นิยมอย่างมากในการแก้ปัญหา อย่างไรก็ตาม วิธีการนี้มีข้อจำกัดในเรื่องของความแม่นยำในการทำนาย เพราะวิธีการนี้มักจะเกิดปัญหาเรื่องความไม่สมดุลของข้อมูลเสมอ วิธีจำแนกแบบหนึ่งต่อหนึ่งกำลังเป็นที่น่าสนใจเนื่องมาจากมีงานวิจัยจำนวนน้อยที่ได้นำเอาวิธีนี้ประยุกต์ใช้งานในงานด้านการจำแนกแบบหลายฉลาก แม้ว่าวิธีนี้จะได้รับการพิสูจน์ว่ามีประสิทธิภาพในการจำแนกมากกว่าวิธีหนึ่งต่อทั้งหมดในงานด้านการจำแนกแบบหลายประเภท อย่างไรก็ตาม วิธีนี้จำเป็นต้องใช้ระยะเวลาในการประมวลผลเป็นอย่างมาก เมื่อทำการทดลองกับข้อมูลที่มีจำนวนของกลุ่มข้อมูลเป็นจำนวนมาก งานวิจัยชิ้นนี้ได้เสนอวิธีการที่ใช้ในการแก้ปัญหางานด้านการจำแนกแบบหลายฉลากด้วยการประยุกต์ใช้งานวิธีการจำแนกแบบหนึ่งต่อหนึ่ง ซึ่งได้ทำการแก้ไขปัญหาความไม่สมดุลของข้อมูลที่เกิดขึ้นด้วยวิธีอันเดอร์แซมพลิ่ง และทำการประยุกต์ใช้งานระบบประมวลผลแบบกระจายสปาร์คด้วยวิธีการแบ่งงานออกเป็นหลาย ๆ ส่วนและทำการกระจายงานเพื่อให้งานแต่ละส่วนทำงานพร้อมกัน ซึ่งระบบนี้สามารถเพิ่มความเร็วในการประมวลผลให้กับวิธีซัพพอร์ตเวคเตอร์แมชชีนแบบหนึ่งต่อหนึ่ง ในขณะที่ยังสามารถคงประสิทธิภาพในการจำแนกข้อมูลไว้ได้แม้ว่าจะใช้งานร่วมกับชุดข้อมูลที่มีกลุ่มข้อมูลจำนวนมากก็ตาม งานวิจัยนี้ได้ทำการทดลองกับข้อมูลแบบหลายฉลากพื้นฐาน 6 ชุดข้อมูล ซึ่งผลของการทดลองนั้นแสดงให้เห็นว่าระบบที่ผู้วิจัยเสนอนั้นสามารถลดระยะเวลาประมวลผลของการใช้วิธีการจำแนกแบบหนึ่งต่อหนึ่งเป็นอย่างมาก ในขณะที่มีประสิทธิภาพในการจำแนกสูงกว่าวิธีการจำแนกแบบหนึ่งต่อทั้งหมดอีกด้วย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2559

5770476621 : MAJOR COMPUTER SCIENCE
KEYWORDS: MULTI-LABEL / SUPPORT VECTOR MACHINE / SPARK / MAP-REDUCE / ONE-VERSUS-ONE

SUTHIPONG DAENGDUANG: Application of One-Versus-One Support Vector Machines to Classify Multi-Label Datasets Using Spark. ADVISOR: PEERAPON VATEEKUL, 62 pp.

Multi-label classification is a supervised learning, where one example can belong to several classes. In the case of Support Vector Machine (SVM), One-versus-All (OVA) is the most common approach to tackle this problem. However, the accuracy is very limited due to extremely imbalanced training set. It is interesting that there have been only very few works that applied One-versus-One (OVO) in the multi-label domain even though it has been shown to provide better accuracy than OVA in the multiclass domain. Anyway, OVO requires an extremely high computational cost when there is a large number of labels. This research propose a multi-label classification framework that employs OVO incorporating with the undersampling, technique to alleviate the imbalanced issue. Spark framework along with a mechanism was applied to split a job to a set of small jobs and then processed them in parallel. The framework can induce OVO SVMs very fast, while maintaining the prediction accuracy even though, there is a large number of classes. The experiment was conducted on 6 standard multi-label datasets. The result indicate that our framework can really reduce computing time on Spark environment, while significantly outperforms OVA in terms of F1 on all data.

Department: Computer Engineering Student's Signature .....
Field of Study: Computer Science Advisor's Signature .....
Academic Year: 2016

กิตติกรรมประกาศ

ตลอดระยะเวลาในการจัดทำวิทยานิพนธ์ฉบับนี้ ได้มีอุปสรรคต่าง ๆ เกิดขึ้น มากมาย ซึ่งเป็นบทเรียนที่ทรงคุณค่ายิ่งแก่ผู้จัดทำซึ่งทำให้สามารถเพิ่มพูนความรู้ประสบการณ์ และความสามารถต่าง ๆ อีกมากมาย ซึ่งทั้งหมดนี้ล้วนเป็นปัจจัยที่ช่วย ส่งเสริมและผลักดันศักยภาพให้แก่ผู้จัดทำเป็นอย่างมาก อย่างไรก็ตาม วิทยานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี ถ้าขาดแรงสนับสนุนจากบุคคลหลายฝ่าย ซึ่งผู้จัดทำซาบซึ้งในการสนับสนุนเหล่านั้นเป็นอย่างมาก และขอใช้เนื้อหาในกิตติกรรมประกาศของวิทยานิพนธ์ฉบับนี้เป็นสื่อในการแสดงความขอบคุณอย่างสุดซึ้งจากผู้จัดทำ

ประการแรก ขอบพระคุณอาจารย์ที่ปรึกษาวิทยานิพนธ์ฉบับนี้ ดร. พีรพล เวทีกูล ผู้ซึ่งอบรมสั่งสอน และให้คำแนะนำในเรื่องต่าง ๆ เสมอมา ไม่ว่าจะเป็นเรื่องของการทำงาน และการใช้ชีวิตประจำวัน ซึ่งเป็นปัจจัยหลักที่ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ฉบับนี้ ที่ให้ข้อคิดและข้อเสนอแนะต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการพัฒนาวิทยานิพนธ์ฉบับนี้ ซึ่งคณะกรรมการ สอบวิทยานิพนธ์นั้น ประกอบไปด้วย ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล รองศาสตราจารย์ ดร. กฤษณะ ไวยมัย และผู้ช่วยศาสตราจารย์ ดร. โชติรัตน์ รัตนามหัทธนะ

ขอขอบคุณ พี่ ๆ เพื่อน ๆ และน้อง ๆ ในห้องปฏิบัติการทุกคนที่ช่วยให้ชีวิตในการทำวิจัยมีสีสัน และมีความหมายมากยิ่งขึ้น รวมทั้งช่วยเสนอแนวคิดต่าง ๆ ในการแก้ไขปัญหาในงานวิทยานิพนธ์ฉบับนี้

สุดท้าย ขอบพระคุณครอบครัวของผู้จัดทำ ที่เป็นกำลังใจ และให้การสนับสนุนทุกสิ่งทุกอย่างด้วยดีเสมอมา

สารบัญ

หน้า

หน้า

หน้า

หน้า

สารบัญรูป

สารบัญตาราง

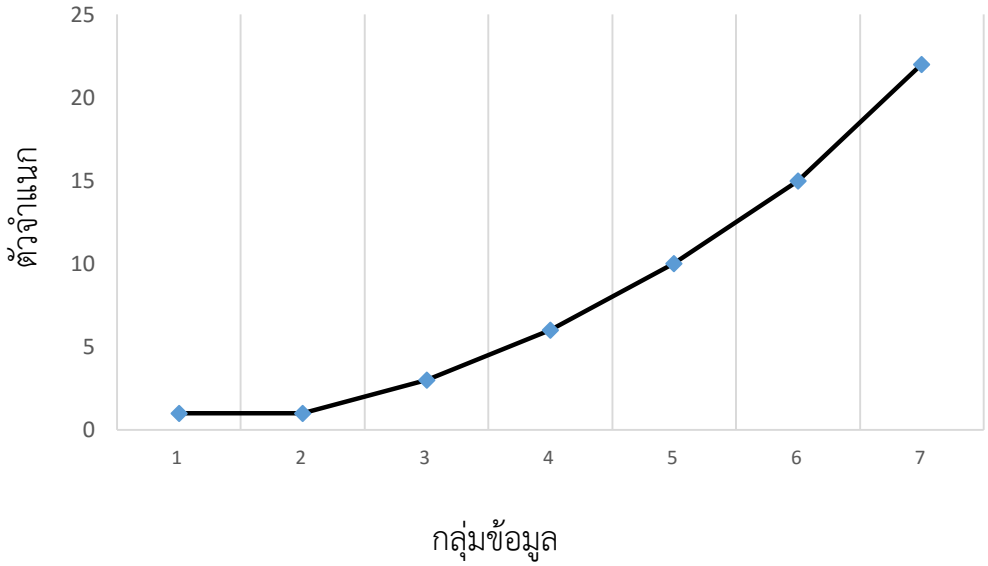
บทที่ 1
ที่มาและความสำคัญ

1.1 ที่มาและความสำคัญของปัญหา

การจำแนกประเภทแบบหลายฉลาก (Multi-label classification, ML) เป็นการจัดกลุ่มข้อมูล โดยใช้งานความรู้ที่มีอยู่ก่อน ช่วยในการจัดกลุ่ม ซึ่งข้อมูลหนึ่งชุดสามารถอยู่ได้มากกว่าหนึ่งกลุ่ม [1-4] ปัจจุบัน วิธีการนี้เป็นที่นิยมใช้อย่างมาก ในด้านของการจำแนกประเภทเอกสาร (Text Categorization) การจำแนกความหมายของภาพ (Semantic Image Labeling) การทำชีวสารสนเทศ (Bioinformatics) การจำแนกชนิดเพลง (Music Categorization) และอื่น ๆ โดยการจำแนกประเภทแบบหลายฉลากนั้นแบ่งออกเป็น 2 วิธี ประกอบด้วย วิธีดัดแปลงอัลกอริทึม (Algorithm Adaptation, AA) และวิธีแปลงปัญหา (Problem Transformation, PT) [5] ซึ่งการดัดแปลงอัลกอริทึมมีเป้าหมายอยู่ที่การสร้างวิธีการใหม่ ซึ่งมีความเฉพาะเจาะจงกับงานด้านการจำแนกแบบหลายฉลาก และวิธีแปลงปัญหาเป็นการเปลี่ยนแปลงปัญหาการจำแนกประเภทแบบหลายฉลาก ให้เป็นกลุ่มของงานด้านการจำแนกแบบสองประเภท (Binary Classification, BC) ซึ่งวิธีนี้เป็นที่นิยมอย่างมากในปัจจุบัน อีกทั้งยังสามารถประยุกต์ใช้งานกับวิธีการจำแนกอื่น ๆ และมีประสิทธิภาพสูงอีกด้วย วิธีแปลงปัญหานั้นจะมีอยู่ 2 วิธี คือ หนึ่งต่อทั้งหมด (One-versus-All, OVA) และหนึ่งต่อหนึ่ง (One-versus-One, OVO)

โดยที่วิธีหนึ่งต่อทั้งหมด [3] จะทำการสร้างตัวจำแนก (Classifier) ขึ้นตามจำนวนของกลุ่ม โดยที่ข้อมูลที่อยู่ในกลุ่มใด ๆ จะถูกเลือกเป็นข้อมูลชุดตัวอย่างบวก (Positive) และข้อมูลที่เหลือทั้งหมดจะถูกเลือกเป็นข้อมูลชุดตัวอย่างลบ (Negative) ยกตัวอย่างเช่น มีกลุ่มข้อมูลอยู่ทั้งหมด 10 กลุ่มคือ C1-C10 ในการสร้างตัวจำแนก C1 วิธีนี้จะทำการเลือกข้อมูลทั้งหมดที่มีความเกี่ยวข้องกับ C1 ให้เป็นข้อมูลชุดตัวอย่างบวก และข้อมูลอื่น ๆ ที่ไม่มีความเกี่ยวข้องกับ C1 จะถูกเลือกให้เป็นข้อมูลชุดตัวอย่างลบ (C2-C10) ซึ่งวิธีการนี้ง่ายต่อการประยุกต์ใช้งานเป็นอย่างมาก แต่อย่างไรก็ตามวิธีนี้สามารถนำไปสู่ปัญหาความไม่สมดุลของข้อมูล (Imbalance Issue) โดยที่ปัญหานี้จะเกิดขึ้นก็ต่อเมื่อข้อมูลชุดตัวอย่างลบมีมากกว่าข้อมูลชุดตัวอย่างบวก จากตัวอย่างข้างต้นข้อมูลตัวอย่างบวกมีเพียงแค่ 10% และข้อมูลตัวอย่างลบมีมากถึง 90% ด้วยเหตุนี้วิธีการนี้มักจะเกิดปัญหาประสิทธิภาพในการจำแนกต่ำเสมอ

วิธีหนึ่งต่อหนึ่ง [6-8] ทำการเลือกจับคู่กลุ่มข้อมูลที่เป็นไปได้ทั้งหมด เพื่อสร้างตัวจำแนกสองประเภท ตัวอย่าง เช่น มีกลุ่มข้อมูลอยู่ 3 กลุ่มคือ C1-C3 วิธีนี้จะสร้างตัวจำแนกสองประเภท โดยจับคู่กลุ่มที่มีอยู่ทั้งหมด จากตัวอย่างข้างต้นจะได้ 3 ตัวจำแนก คือ C1vsC2 C1vsC3 และ C2vsC3 วิธีนี้จะใช้การลงคะแนนเลือกกลุ่มที่มีการตอบมากที่สุดเป็นคำตอบวิธีหนึ่งต่อหนึ่งนิยมใช้ในการจำแนกหลายประเภท (Multiclass Classification) และวิธีนี้ยังได้รับการพิสูจน์ว่ามีความสามารถสูงกว่าวิธีหนึ่งต่อทั้งหมด แต่อย่างไรก็ตาม วิธีนี้ถูกใช้งานน้อยมากในงานด้านการจำแนกประเภทแบบหลายฉลาก เนื่องจากวิธีนี้สามารถตอบได้เพียงแค่กลุ่มเดียวเท่านั้น อีกทั้งการใช้วิธีหนึ่งต่อหนึ่งกับงานที่มีกลุ่มของข้อมูลจำนวนมากนั้นไม่สามารถทำได้ในการใช้งานคอมพิวเตอร์เครื่องเดียว เพราะการใช้วิธีนี้นั้นจำเป็นต้องสร้างตัวจำแนกจำนวนมาก จากรูปที่ 1 จะแสดงให้เห็นถึงอัตราการเพิ่มจำนวนของตัวจำแนกต่อประเภทของข้อมูล



รูปที่ 1 อัตราการเพิ่มจำนวนของตัวจำแนก

อย่างไรก็ตาม งานในด้านการจัดกลุ่มแบบหลายประเภทนั้น วิธีหนึ่งต่อหนึ่งเป็นหนึ่งในวิธีการจำแนกประเภทที่ได้รับความนิยมมากที่สุด ซึ่งให้ความถูกต้องและแม่นยำสูงกว่าวิธีการจำแนกประเภทอื่น ๆ โดยเฉพาะอย่างยิ่งในงานด้านการจัดหมวดหมู่เอกสาร วิธีนี้ทำการสร้างระนาบการแบ่งข้อมูลที่เหมาะสม เพื่อใช้ในการแยกกลุ่มของข้อมูลออกเป็น 2 กลุ่ม โดยข้อมูลที่ปรากฏอยู่บนระนาบจะถูกเรียกว่า “ซัพพอร์ตเวคเตอร์ (Support Vector)” ระนาบนี้จะถูกเรียกว่า “ไฮเปอเพลน (Hyperplane)”

ปัจจุบันข้อมูลต่าง ๆ ได้มีแนวโน้มว่าจะเพิ่มขึ้นอย่างมหาศาล [8, 9] เป็นเหตุให้เวลาทำงานในด้านต่าง ๆ จำเป็นต้องใช้เวลานาน และต้องมีอุปกรณ์ที่มีประสิทธิภาพสูง จึงได้มีการพัฒนาระบบประมวลผลแบบกระจาย (Distributed Computing) [9-14] เพื่อใช้ในการแก้ปัญหา ซึ่งระบบประมวลผลแบบกระจายเป็นการนำเอาเครื่องคอมพิวเตอร์หลาย ๆ เครื่องมาช่วยในการประมวลผล โดยมองว่ากลุ่มของเครื่องนั้นเป็นคอมพิวเตอร์เครื่องเดียว ซึ่งจะแบ่งส่วนข้อมูลที่มีอยู่ให้เป็นหลาย ๆ ส่วน และเก็บลงในแต่ละเครื่อง และประมวลผลโดยระบบประมวลผลแบบกระจายที่นิยมใช้กันเป็นอย่างมากจะมี 2 ระบบ คือ ฮาร์ดูป (Hadoop) และ สปาร์ค (Spark) โดยที่ฮาร์ดูปนั้นจะทำงานอยู่บนพื้นฐานของฮาร์ดดิสก์ (Hard Disk) สปาร์คนั้นทำงานอยู่บนพื้นฐานของหน่วยความจำ (Memory) ซึ่งได้มีการพิสูจน์แล้วว่าสปาร์คนั้นมีความเร็วในการทำงานมากกว่าฮาร์ดูปเป็นอย่างมาก

อย่างไรก็ตาม ปัญหาความไม่สมดุลของข้อมูล [4, 9, 15] นั้นมักจะเกิดขึ้นเสมอ ไม่ว่าใช้วิธีการจัดกลุ่มแบบใดก็ตาม ซึ่งในปัจจุบันมีวิธีแก้ไขอยู่หลายวิธี แต่วิธีที่นิยมใช้งานมากที่สุดคือ โอเวอร์แซมพลิ่ง (Oversampling) และ อันเดอร์แซมพลิ่ง (Undersampling) [9, 16] โดยที่วิธีการอันเดอร์แซมพลิ่งจะทำการสร้างสมดุลของข้อมูลโดยการสุ่มลบข้อมูลที่มีเสียงข้างมาก (Majority Class) ให้จำนวนข้อมูลที่มีเสียงข้างมากทั้งหมดมีจำนวนเท่ากับข้อมูลที่มีเสียงข้องน้อย (Minority Class) ตัวอย่าง เช่น มีข้อมูลตัวอย่างบวกอยู่ 50 ตัว และมีข้อมูลตัวอย่างลบอยู่ 200 ตัว วิธีการนี้จะทำการสุ่มลบข้อมูลตัวอย่างลบออก 150 ตัว เพื่อสร้างความสมดุลระหว่างข้อมูล วิธีโอเวอร์แซมพลิ่งจะทำการสุ่มเพิ่มข้อมูลที่มีเสียงข้างน้อยที่มีอยู่แล้วให้มีจำนวนเท่ากับข้อมูลที่มีเสียงข้างมาก ตัวอย่าง เช่น มีข้อมูลตัวอย่างบวก 50 ตัว ข้อมูลตัวอย่างลบ 200 ตัว วิธีนี้จะทำการเพิ่มข้อมูลตัวอย่างบวกจำนวน 150 ตัว เพื่อให้เกิดความสมดุลระหว่างข้อมูลตัวอย่างบวกและข้อมูลตัวอย่างลบ โดยที่วิธีอันเดอร์แซมพลิ่งนั้นเป็นวิธีที่นิยมใช้งานเป็นอย่างมากในปัจจุบัน

งานวิจัยนี้มีเป้าหมายในการพัฒนาวิธีการจำแนกประเภทแบบหลายฉลาก โดยใช้วิธีหนึ่งต่อหนึ่งด้วยการประยุกต์ใช้ระบบการประมวลผลแบบกระจาย เพื่อแก้ไขปัญหาในการทำงานกับชุดข้อมูลที่มีกลุ่มจำนวนมาก อีกทั้งยังสามารถเพิ่มประสิทธิภาพของการจำแนกได้อีกด้วย โดยงานวิจัยนี้จะประยุกต์ใช้วิธีอันเดอร์แซมพลิ่งเพื่อใช้ในการแก้ปัญหาความไม่สมดุลของข้อมูลแล้วทำการเลือกตอบข้อมูลที่เป็นชุดตัวอย่างลบ เพื่อประยุกต์ใช้กับงานด้านการจำแนกประเภทแบบหลายฉลาก โดยใช้วิธีการจำแนกแบบหนึ่งต่อหนึ่ง

1.2 วัตถุประสงค์

1) เพื่อเพิ่มประสิทธิภาพในการทำนายของ วิธีการจำแนกแบบหลายฉลาก โดยใช้วิธีการแบบหนึ่งต่อหนึ่ง ด้วยการประยุกต์ใช้ซัพพอร์ตเวคเตอร์แมชชีน
2) ประยุกต์ใช้ระบบประมวลผลแบบกระจาย เพื่อทำการลดระยะเวลาในการประมวลผลของระบบทั้งหมด

1.3 ขอบเขตของงานวิจัย

1) พัฒนาวิธีการจำแนกข้อมูลแบบหลายฉลาก โดยประยุกต์ใช้งานวิธีการจำแนกข้อมูลแบบหนึ่งต่อหนึ่ง
2) ใช้งานอัลกอลิทึมซัพพอร์ตเวคเตอร์แมชชีนในการประยุกใช้วิธีการจำแนกข้อมูลแบบหนึ่งต่อหนึ่ง
3) ประยุกต์ใช้ระบบประมวลผลแบบกระจาย เพื่อทำการลดระยะเวลาที่ใช้ในการประมวลผลข้อมูลของวิธีการจำแนกข้อมูลแบบหนึ่งต่อหนึ่ง
4) ทำการทดลองโดยใช้งานชุดข้อมูลแบบหลายฉลาก โดยในหนึ่งตัวอย่างนั้นสามารถปรากฏอยู่ได้หลายกลุ่มข้อมูล
5) ทำการแก้ไขปัญหาความไม่สมดุลของข้อมูลที่ปรากฏขึ้นในบางตัวจำแนก ด้วยการประยุกต์ใช้งานวิธีแซมพลิ่ง

1.4 ประโยชน์ที่ได้รับ

เพิ่มความสามารถของการจำแนกประเภทแบบหลายฉลากด้วยวิธีหนึ่งต่อหนึ่งให้สามารถทำงานกับชุดข้อมูลที่มีกลุ่มจำนวนมากได้ และเพิ่มประสิทธิภาพของการจำแนกให้สูงขึ้น อีกทั้งสามารถลดระยะเวลาการทำงานร่วมกับข้อมูลที่มีขนาดใหญ่และมีความซับซ้อนสูงได้อีกด้วย

1.5 วิธีดำเนินการวิจัย

1) ศึกษาวิธีการจำแนกแบบหลายฉลาก
2) ศึกษางานวิจัยที่เกี่ยวข้องกับวิธีการจำแนกหลายฉลาก
3) ศึกษาวิธีการจำแนกแบบหลายฉลากที่มีการประยุกต์ใช้วิธีการจำแนกหนึ่งต่อหนึ่ง
4) ศึกษาระบบประมวลผลแบบกระจาย

5) ศึกษางานวิจัยที่มีการนำระบบประมวลผลแบบกระจายมาประยุกต์ใช้งาน

6) ทำการจำแนกปัญหาที่พบในการใช้งานวิธีการจำแนกแบบหลายฉลากแบบหนึ่งต่อหนึ่ง และการประยุกใช้ระบบประมวลผลแบบกระจาย

7) ศึกษาและพัฒนาวิธีการแก้ไขปัญหาที่พบ

8) ออกแบบวิธีการประยุกต์ใช้งานระบบประมวลผลแบบกระจาย

9) ทดสอบประสิทธิภาพของระบบที่ได้นำเสนอในด้านของความสามารถในการจำแนกและความเร็วในการประมวลผล โดยนำมาเปรียบเทียบกับวิธีพื้นฐานอื่น ๆ

10) สรุปและวิเคราะห์ผลการทดลอง

11) เรียบเรียงและรายงานวิทยานิพนธ์

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในหัวข้อนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับงานวิจัยชิ้นนี้ ประกอบด้วย ซัพพอร์ตเวกเตอร์แมตชีน การจำแนกประเภทแบบหลายฉลาก วิธีการแก้ปัญหาความไม่สมดุล และระบบประมวลผลแบบกระจาย

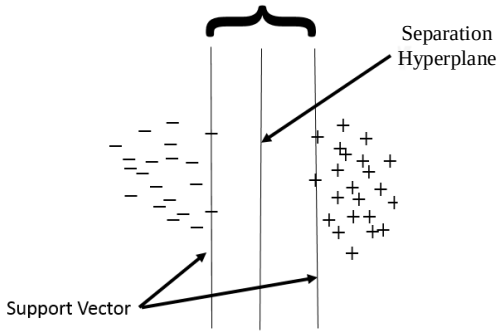
2.1 ซัพพอร์ตเวคเตอร์แมชชีน (Support Vector Machine, SVM)

วิธีการนี้เป็นหนึ่งในวิธีการจำแนกประเภทที่ได้รับความนิยมมากที่สุด ซึ่งให้ความถูกต้องและแม่นยำสูงกว่าวิธีการจำแนกประเภทอื่น ๆ โดยเฉพาะอย่างยิ่งในงานด้านการจัดหมวดหมู่เอกสาร วิธีนี้ทำการสร้างระนาบการแบ่งข้อมูลที่เหมาะสม เพื่อใช้ในการแยกกลุ่มของข้อมูลออกเป็น 2 กลุ่ม โดยข้อมูลที่ปรากฏอยู่บนเส้น จะถูกเรียกว่าซัพพอร์ตเวคเตอร์ ระนาบนี้จะถูกเรียกว่าไฮเปอเพลน [3-5, 8, 15, 17] ซึ่งในการสร้างระนาบการแบ่งข้อมูลที่เหมาะสม สามารถสร้างได้โดยใช้สมการที่ (1)

$$h(\vec{w}, b) = \vec{w} \times (x + b) \quad (1)$$

$\vec{w}$ คือ เวคเตอร์ที่ตั้งฉากกับไฮเปอร์เพลน

b คือ ค่าคงที่ซึ่งกำหนดตำแหน่งของเวคเตอร์



รูปที่ 2 ซัพพอร์ตเวคเตอร์แมชชีน

อย่างไรก็ตาม ซัพพอร์ตเวกเตอร์แมชชีน ไม่สามารถแยกแยะข้อมูลได้ถูกต้องทั้งหมด จึงทำให้มีการกำหนดตัวแปรเพื่อใช้ในการยอมรับค่าความผิดพลาด ซึ่งตัวแปรนั้นจะถูกเรียกว่าตัวแปรผ่อนผัน

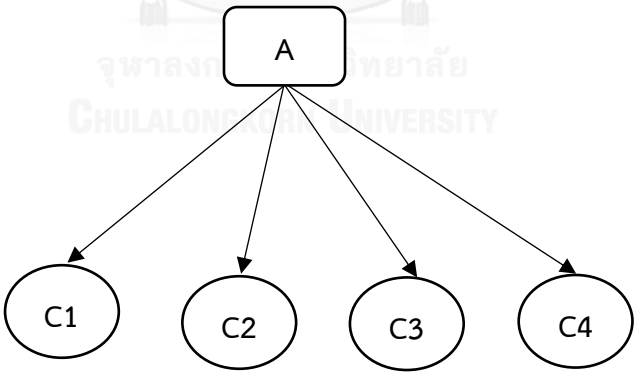
(Slack Variable) โดยตัวแปรนี้จะมีค่ามากกว่า 0 เสมอ โดยที่ค่าของตัวแปรนั้นสามารถหาได้จากสมการที่ 2

$$Minimize_{w,b,\xi} \ \frac{1}{2}w^T w + C\sum_{i=1}^{|D|}\xi_i \qquad (2)$$

C คือ ค่าที่กำหนดเพื่อปรับความสมดุลระหว่างการให้ความสำคัญของระยะแยกแยะสูงสุดหรือให้ความสำคัญกับค่าความผิดพลาด โดยปกติค่า ***C*** จะกำหนดให้มีค่ามากกว่า 1

2.2 การจำแนกประเภทแบบหลายฉลาก (Multi-Label Classification)

การจำแนกรูปแบบนี้ข้อมูลอาจถูกจำแนกให้มีมากกว่าหนึ่งประเภท [1-3, 8, 15, 18] จากประเภทข้อมูลที่มีมากกว่าหรือเท่ากับสองประเภทดังรูปที่ 3 โดยตัวอย่าง A สามารถอยู่ได้หลายกลุ่มข้อมูล ประกอบไปด้วยกลุ่มข้อมูลดังนี้ C1 C2 C3 และ C4 โดยที่การจำแนกประเภทแบบหลายฉลากนั้นแบ่งออกเป็น 2 วิธี คือ วิธีดัดแปลงอัลกอริทึม และวิธีแปลงปัญหาซึ่งการดัดแปลงอัลกอริทึม มีเป้าหมายในการสร้างวิธีการใหม่ที่มีความเฉพาะกับงานด้านการจำแนกแบบหลายฉลาก และวิธีแปลงปัญหาเป็นวิธีที่ใช้ในการเปลี่ยนแปลงการจำแนกแบบหลายฉลาก เป็นกลุ่มของงานด้านการจำแนกสองประเภท ซึ่งวิธีนี้เป็นที่นิยมอย่างมากในปัจจุบัน อีกทั้งยังสามารถประยุกต์ใช้งานกับวิธีการจำแนกอื่น ๆ และมีประสิทธิภาพสูงอีกด้วย วิธีแปลงปัญหานั้นจะมีวิธีการอยู่ 2 วิธี ประกอบด้วย วิธีหนึ่งต่อทั้งหมด และหนึ่งต่อหนึ่ง



รูปที่ 3 การจำแนกประเภทแบบหลายฉลาก

2.2.1 วิธีหนึ่งต่อทั้งหมด (One-versus-All, OVA)

โดยที่วิธีหนึ่งต่อทั้งหมด [1, 3] นั้นจะทำการสร้างตัวจำแนกขึ้นตามจำนวนของกลุ่ม โดยที่ข้อมูลที่อยู่ในกลุ่มใด ๆ จะถูกเลือกเป็นข้อมูลชุดตัวอย่างบวก และข้อมูลที่เหลือทั้งหมดจะถูกเลือกเป็นข้อมูลชุดตัวอย่างลบยกตัวอย่าง เช่น มีกลุ่มข้อมูลอยู่ 10 กลุ่มคือ C1-C10 ในการสร้างตัวจำแนก C1 วิธีนี้จะทำการเลือกข้อมูลทั้งหมดที่มีความเกี่ยวข้องกับ C1 ให้เป็นข้อมูลชุดตัวอย่างบวก และข้อมูลอื่น ๆ ที่ไม่มีความเกี่ยวข้องกับ C1 จะถูกเลือกให้เป็นข้อมูลชุดตัวอย่างลบ (C2-C10) ซึ่งวิธีการนี้เป็นวิธีที่ง่ายต่อการประยุกต์ใช้งานเป็นอย่างมาก แต่อย่างไรตามวิธีนี้สามารถนำไปสู่ปัญหาความไม่สมดุลของข้อมูลได้

2.2.2 วิธีหนึ่งต่อหนึ่ง (One-versus-One, OVO)

วิธีหนึ่งต่อหนึ่ง [8, 18] จะทำการเลือกจับคู่ข้อมูลที่เป็นไปได้ทั้งหมด เพื่อสร้างตัวจำแนกสองประเภท ตัวอย่าง เช่น มีกลุ่มข้อมูลอยู่ 3 กลุ่ม C1-C3 ซึ่งวิธีนี้จะสร้างตัวจำแนกจากการจับคู่กลุ่มที่มีอยู่ทั้งหมด จากตัวอย่างจะได้สามตัวจำแนกคือ C1vsC2 C1vsC3 และ C2vsC3 วิธีนี้นั้นจะใช้การลงคะแนนเลือกกลุ่มที่มีการตอบมากที่สุดเป็นคำตอบ อย่างไรก็ตามวิธีหนึ่งต่อหนึ่ง นิยมใช้ในการจำแนกหลายประเภท อีกทั้งวิธีนี้ยังได้รับการพิสูจน์แล้วว่ามีความสามารถสูงกว่าวิธีหนึ่งต่อทั้งหมด [6, 7] แต่อย่างไรก็ตามวิธีนี้นั้นมีการใช้งานน้อยมากในงานของการจำแนกแบบหลายฉลาก เนื่องจากวิธีนี้สามารถตอบได้เพียงแค่กลุ่มเดียวเท่านั้น โดยที่การจำแนกประเภทแบบหลายฉลาก นิยมใช้งานวิธีซัพพอร์ตเวเตอร์แมตชีนเป็นอย่างมาก เนื่องจากวิธีการเป็นวิธีการที่มีประสิทธิภาพสูง อีกทั้งยังสามารถนำมาประยุกต์ใช้กับวิธีการอื่น ๆ ได้ง่าย

2.3 วิธีแก้ปัญหาความไม่สมดุล (Imbalance Solution)

ปัญหาความไม่สมดุลของข้อมูล เป็นปัญหาที่เกิดขึ้นเสมอในการจัดกลุ่มข้อมูล ซึ่งในปัจจุบันนั้นได้มีการพัฒนาวิธีที่ใช้ในการแก้ปัญหาเรื่องความไม่สมดุลของข้อมูลอยู่ 4 วิธี คือ การสุ่มตัวอย่าง การรู้จำ ค่าอ่อนไหว และรวมทั้งหมด [8, 9, 18, 19]

2.3.1 การสุ่มตัวอย่าง (Sampling)

วิธีการนี้เป็นการสุ่มเพิ่มหรือลดจำนวนข้อมูลแต่ละตัวจำแนกโดยแบ่งออกเป็นวิธี 2 วิธี คือ โอเวอร์แซมพลิ่ง (Oversampling) และ อันเดอร์แซมพลิ่ง (Undersampling)

วิธีโอเวอร์แซมพลิ่ง จะทำการสุ่มเพิ่มข้อมูลเสียงข้างน้อย ให้มีจำนวนเท่ากับข้อมูลที่มีเสียงข้างมาก ยกตัวอย่างเช่น มีข้อมูลตัวอย่างบวก 50 ตัว ช้อมูลตัวอย่างลบ 200 ตัว วิธีนี้จะทำการเพิ่มข้อมูลตัวอย่างบวกจำนวน 150 ตัว เพื่อให้เกิดความสมดุลระหว่างข้อมูล อย่างไรก็ตามวิธีนี้มีข้อเสียคืออาจจะนำไปสู่ปัญหาความแม่นยำที่เกินความจริง (Overfiting) ได้

วิธีอันเดอร์แซมพลิ่ง จะทำการสร้างสมดุลของข้อมูลโดยการสุ่มลบข้อมูลเสียงข้างมากให้เท่ากับข้อมูลเสียงข้างน้อย ยกตัวอย่าง เช่น มีข้อมูลตัวอย่างบวกอยู่ 50 ตัว และมีข้อมูลตัวอย่างลบ 200 ตัว วิธีการนี้จะสุ่มลบข้อมูลตัวอย่างลบออก 150 ตัว เพื่อสร้างความสมดุลระหว่างข้อมูล ซึ่งวิธีการนี้เป็นวิธีที่นิยมใช้อย่างกว้างขวางในการแก้ไขปัญหาเรื่องความไม่สมดุลของข้อมูล แต่อย่างไรก็ตาม วิธีนี้นั้นได้ทำการลบข้อมูลบางอย่างออกไป ซึ่งข้อมูลที่ถูกลบไปนั้นอาจจะเป็นข้อมูลที่มีนัยยะสำคัญที่ใช้ในการจำแนกอาจทำให้นำไปสู่ปัญหาจำแนกผิดพลาดได้

2.3.2 การรู้จำ (Recognition)

วิธีการนี้จะถูกเรียกว่าการเรียนรู้เพียงกลุ่มเดียว เนื่องจากวิธีการนี้สนใจเรียนรู้เพียงกลุ่มข้อมูลที่เป็นเป้าหมายเท่านั้นตัวอย่างเช่น มีกลุ่มข้อมูลทั้งหมด 3 กลุ่มประกอบไปด้วยกลุ่ม C1 C2 และ C3 หากเราสนใจในกลุ่ม C1 วิธีการนี้จะทำกรเรียนรู้เพียงแค่กลุ่ม C1 เท่านั้น โดยไม่สนใจกลุ่มอื่น แต่อย่างไรก็ตาม วิธีนี้จะไม่สามารถสร้างตัวจำแนกได้โดยใช้วิธีอื่น ๆ

2.3.3 ค่าอ่อนไหว (Cost Sensitive)

วิธีการนี้จะทำการกำหนดค่าของกลุ่มข้อมูลตัวชุดอย่างบวก ที่มีโอกาสในการจัดกลุ่มผิดให้มีค่าสูง และทำการสร้างแบบจำลองในการจัดกลุ่มโดยกำหนดให้มีค่าต่ำ อย่างไรก็ตาม วิธีนี้มีข้อเสียคืออาจจะนำไปสู่ปัญหาเข้ากันได้มากเกินไปเช่น ต้องการจัดกลุ่ม C1 แต่ในกลุ่ม C1 นั้นมีโอกาสที่ตัวจำแนกจะทำการจัดกลุ่มผิดพลาดสูงมากวิธีการนี้จำทำการกำหนดค่าความสำคัญของกลุ่ม C1 ไว้สูงมาก และทำการสร้างโครงสร้างของการจำแนก โดยกำหนดให้มีค่าต่ำเพื่อลดปัญหาการทำนายผิดพลาด

2.3.4 การรวมทั้งหมด (Ensemble)

วิธีนี้จะทำการสร้างตัวจำแนกหลาย ๆ ชนิดขึ้นมา เพื่อทำการเปรียบเทียบหาตัวที่ดีที่สุด ซึ่งใช้ในการเพิ่มประสิทธิภาพของการทำนาย แต่อย่างไรก็ตาม วิธีนี้อาจจะนำไปสู่ปัญหาความแม่นยำที่เกินความจริง และยังใช้เวลาในการทำงานนานอีกด้วย

2.4 ระบบประมวลผลแบบกระจาย (Distributed Computing)

ระบบประมวลผลแบบกระจาย [10-14] ประกอบไปด้วยคอมพิวเตอร์หลายเครื่องร่วมกันทำงานในงานชิ้นเดียวกันโดยมีลักษณะเป็นลูกข่ายแม่ข่าย (Client/Server) ซึ่งเครื่องแม่ข่ายจะมีหน้าที่ในการจัดการงาน และส่งต่องานให้เครื่องลูกข่าย และเครื่องลูกข่ายมีหน้าที่ในการประมวลผลงานที่ได้รับจากเครื่องแม่ข่าย โดยใช้งานคอมพิวเตอร์ทุกเครื่องที่เชื่อมต่อกันเป็นเครือข่ายคอมพิวเตอร์ ปัจจุบันวิธีการประมวลผลแบบกระจายเป็นวิธีการกระมวลผลที่ให้ประสิทธิภาพสูงและคุ้มทุนที่สุด เนื่องจากวิธีการประมวลผลแบบกระจายสามารถเพิ่มประสิทธิภาพ และความสามารถของการประมวลผลง่ายตามจำนวนคอมพิวเตอร์ในเครือข่าย นอกจากนั้นยังมีเฟรมเวิร์คที่ใช้บริหารจัดการคอมพิวเตอร์ที่ทำงานแบบกระจายต่าง ๆ เกิดขึ้นมากมายทำให้ง่ายต่อการพัฒนา และประมวลผล โดยที่จะมีระบบที่นิยมใช้งานอยู่ 2 ระบบ คือ ฮาร์ดูป และสปาร์ค

2.4.1 ฮาร์ดูป (Hadoop)

ฮาร์ดูป [11-14] เป็นระบบที่ใช้ในการเก็บข้อมูล และประมวลผลแบบกระจาย ด้วยการใช้คอมพิวเตอร์หลายเครื่อง ซึ่งฮาร์ดูปถูกออกแบบมาโดยคำนึงถึง 3 ปัจจัยหลัก

1) ความเสถียรของการทำงาน
2) การขยายหรือเพิ่มประสิทธิภาพ
3) การทำงานแบบกระจาย

ซึ่งทำให้ฮาร์ดูปนั้นสามารถเพิ่มจำนวนคอมพิวเตอร์ในเครือข่ายคอมพิวเตอร์ได้ถึงหลักแสนเครื่องในเครือข่ายเดียวกัน โดยแต่ละคอมพิวเตอร์มีพื้นที่เก็บข้อมูล และส่วนประมวลผลของตัวเครื่องเอง องค์ประกอบหลักของฮาร์ดูปประกอบด้วย 3 ส่วนหลัก คือ Hadoop Distributed Filesystem System (HDFS) Yet Another Resource Manager (YARN) และ MapReduce

2.4.1.1 HDFS

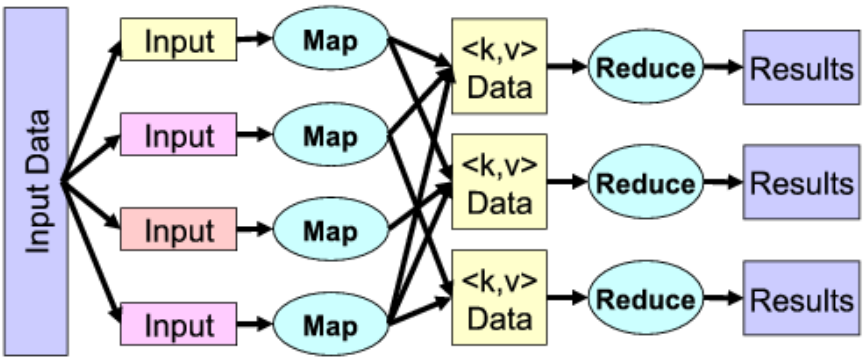
HDFS ทำหน้าที่เก็บข้อมูลแบบกระจายบนฮาร์ดดิสของเครื่องคอมพิวเตอร์ โดยแบ่งข้อมูลขนาดใหญ่ออกเป็นส่วนย่อยๆ และส่งไปเก็บยังเครื่องลูกข่ายนอกจากนั้น HDFS ยังทำสำเนาข้อมูลเป็นจำนวน 2 ชุด รวมกับไฟล์ต้นฉบับ 1 ชุด เป็น 3 ชุดเก็บลงในคอมพิวเตอร์ในเครือข่ายต่างเครื่องกันเพื่อป้องกันการเกิดปัญหาต่าง ๆ ซึ่งอาจจะทำให้มีข้อมูลสูญหายหรือเสียหาย

2.4.1.2 YARN

YARN มีหน้าที่ในการจัดการทรัพยากรของเครื่อง เพื่อใช้ในการประมวลผล และจัดการการทำงานต่าง ๆ อีกทั้งยังช่วยให้สามารถประมวลผลงานข้อมูลขนาดใหญ่ได้ง่ายอีกด้วย

2.4.1.3 แมพรีดิวซ์ (MapReduce)

แมพรีดิวซ์ [10-14] คือการพัฒนาโปรแกรมแบบกระจาย ซึ่งใช้งานกับข้อมูลที่มีขนาดใหญ่บนคอมพิวเตอร์หลายเครื่องบนเครือข่ายเดียวกัน ซึ่งแมพรีดิวซ์ประกอบด้วย 2 ส่วนการทำงานคือ 1 แมพ 2 รีดิวซ์ แมพที่ทำหน้าที่กรองและปรับปรุงข้อมูลให้อยู่ในรูปของ คีย์/แวลู (k,v) และรีดิวซ์มีหน้าที่นำข้อมูลที่อยู่ในรูปแบบของ คีย์/แวลูมาประมวลหาคำตอบ



รูปที่ 4 ตัวอย่างการทำงานของวิธีแมพรีดิวซ์

2.4.2 สปาร์ค (Spark)

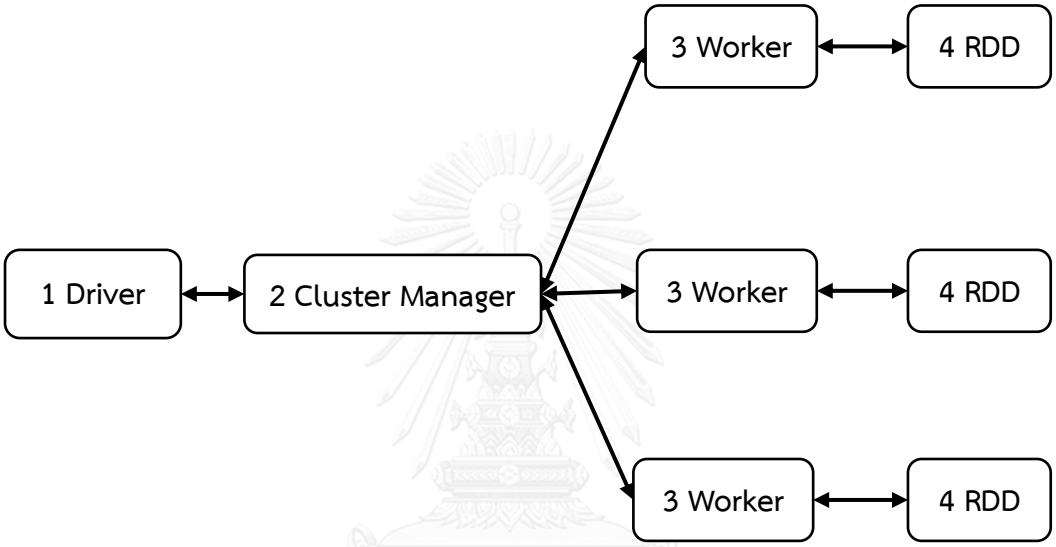
สปาร์ค [10, 20] เป็นระบบประมวลผลแบบกระจายที่พัฒนาต่อเนื่องมาจากฮาร์ดูป โดยระบบนี้มีการประมวลผลอยู่บนหน่วยความจำ ซึ่งสปาร์คนั้นได้รับการพิสูจน์จากผู้พัฒนาว่ามีความสามารถในการประมวลผลได้เร็วกว่า ฮาร์ดูป เป็นอย่างมากโดยสปาร์คออกแบบวิธีการเก็บข้อมูลโดยคำนึงถึง 3 ปัจจัยหลัก ประกอบด้วย

1) ความยืดหยุ่น (Resilient) เป็นปัจจัยที่มีเพื่อใช้ในการป้องการปัญหา เมื่อข้อมูลเกิดการสูญหายหรือ เครื่องลูกข่ายมีความเสียหายเกิดขึ้นจนไม่สามารถทำงานต่อได้

2) ความสามารถในการกระจาย (Distributed) ปัจจัยนี้เกิดขึ้นเพื่อทำให้ข้อมูลสามารถกระจายออกไปยังเครื่องลูกข่ายหลาย ๆ เครื่องได้

3) ความสามารถในการเก็บชุดข้อมูล (Datasets) เป็นปัจจัยที่เกิดขึ้น เพื่อใช้ในการเก็บรวบรวมข้อมูลที่ถูกกระจายออกเป็นหลาย ๆ ส่วนให้อยู่ในรูปแบบดั้งเดิมโดยที่ไม่มีการเปลี่ยนแปลงใด ๆ

โครงสร้างของระบบประมวลผลแบบกระจายสปาร์คประกอบด้วย 4 ส่วน ดังรูปที่ 5



รูปที่ 5 ตัวอย่างโครงสร้างการทำงานของสปาร์ค

2.4.2.1 เครื่องแม่ข่าย (Driver)

เครื่องแม่ข่ายทำหน้าที่ในการจัดการแบ่งและกระจายงานเพื่อส่งต่อไปยังส่วนของตัวจัดการกลุ่มข้อมูล อีกทั้งยังมีหน้าที่ในการตรวจสอบสถานะการทำงานของเครื่องลูกข่ายและทำหน้าในการจัดการลำดับในการทำงานของระบบทั้งหมดอีกด้วย

2.4.2.2 ตัวจัดการกลุ่มข้อมูล (Cluster manager)

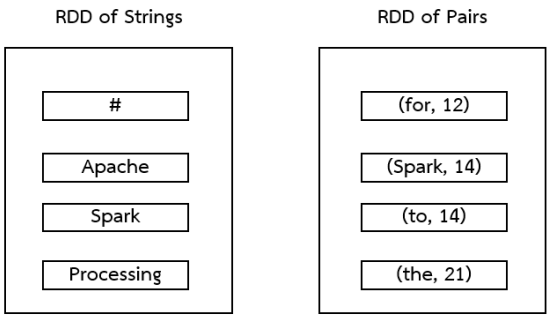
ตัวจัดการข้อมูลทำหน้าที่ในการจัดสรรและกำหนดทรัพยากรในการทำงานของระบบทั้งหมด โดยจะคัดกรองจากขนาดของงานที่ได้รับจากเครื่องแม่ข่าย และทำการกำหนดงานให้เครื่องลูกข่ายแล้วส่งรายละเอียดของเครื่องลูกข่ายกลับไปยังเครื่องแม่ข่าย เพื่อใช้ในการติดตามสถานะการทำงานของเครื่องลูกข่าย

2.4.2.3 เครื่องลูกข่าย (Worker)

เครื่องลูกข่ายทำหน้าที่ในการประมวลผลงานที่ได้รับจากเครื่องแม่ข่าย โดยจะทำการประมวลผลข้อมูลที่ถูกบันทึกอยู่ใน RDD และแสดงผลตามงานที่เครื่องแม่ข่ายกำหนด

*2.4.2.4 RDD (*Resilient Distributed Dataset*)*

RDD เป็นส่วนที่ใช้ในการเก็บบันทึกข้อมูลในหน่วยความจำ เพื่อนำไปใช้ในการประมวลผลต่าง ๆ โดยข้อมูลที่เก็บอยู่ใน RDD นั้นสามารถเก็บได้อยู่ในรูปของ คีย์/แวลู หรือตัวอักษรได้ อีกทั้งยังสามารถใช้ในการเชื่อมต่อกับข้อมูลที่อยู่ในฮาร์ดูป เพื่อนำข้อมูลมาใช้งาน ตัวอย่างในการจัดเก็บข้อมูลของ RDD เป็นดังรูปที่ 6



รูปที่ 6 ลักษณะการจัดเก็บข้อมูลใน RDD

ลักษณะการเก็บข้อมูลของ RDD มี 7 ประการ [20] ดังนี้

1) ข้อมูลที่ถูกเก็บอยู่ใน RDD อยู่ในหน่วยความจำ
2) ข้อมูลที่อยู่ใน RDD เป็นข้อมูลที่สามารถอ่านได้อย่างเดียวเท่านั้น ถ้าต้องการเปลี่ยนรูปร่างของข้อมูลใน RDD จำเป็นต้องสร้าง RDD ใหม่ขึ้นมาแทน
3) ข้อมูลที่เก็บอยู่ใน RDD ไม่สามารถเปลี่ยนรูปร่างได้
4) ข้อมูลใน RDD สามารถสร้างหน่วยเก็บข้อมูล เพื่อใช้ในการเก็บข้อมูลในหน่วยความจำได้
5) ข้อมูลที่อยู่ใน RDD สามารถกระจายไปยังหลาย ๆ เครื่องได้
6) ข้อมูลที่อยู่ใน RDD นั้นสามารถกำหนดลักษณะของตัวแปรเพื่อใช้เก็บข้อมูลได้
7) ข้อมูลที่อยู่ใน RDD นั้นสามารถแบ่งส่วนออกเป็นหลาย ๆ ส่วนได้

2.5 การวัดประสิทธิภาพการทำงาน (Performance Evaluation)

ในหัวข้อนี้จะกล่าวถึงตัวชี้วัดที่ใช้เพื่อวัดประสิทธิภาพของงานวิจัย ซึ่งประกอบด้วย ตัวชี้วัดประสิทธิภาพการจำแนกแบบสองประเภท ตัวชี้วัดประสิทธิภาพการจำแนกประเภทแบบหลาย และตัวชี้วัดประสิทธิภาพการเลือกตอบตามตัวอย่าง

2.5.1 ตัวชี้วัดประสิทธิภาพการจำแนกแบบสองประเภท

การวัดประสิทธิภาพการจำแนกประเภทข้อมูลสองประเภท [4, 5, 8, 15] เป็นการวัดข้อมูลที่มีประเภทเพียงสองประเภทเท่านั้น เช่น ใช่หรือไม่ใช่ อยู่ในกลุ่มนั้นหรือไม่อยู่ในกลุ่มนั้นหรือไม่ เป็นต้น ซึ่งในการวัดประสิทธิภาพการทำงานของข้อมูลสองประเภทนั้นจะใช้ตัววัดหลายตัวในการวัดประสิทธิภาพ เนื่องจากตัวชี้วัดแต่ละตัวมีหน้าที่ในการวัดประสิทธิภาพที่แตกต่างกัน โดยที่ตัวชี้วัดนั้นสามารถคำนวณได้จากค่าต่อไปนี้ TP, FP, TN, FN โดย TP, FP, TN, FN มีนิยาม ดังนี้

True (T)	หมายถึง ผลลัพธ์จากการทำนายถูกต้อง
False (F)	หมายถึง ผลลัพธ์จากการทำนายไม่ถูกต้อง
Positive (P)	หมายถึง ผลลัพธ์จากการทำนายทำนายว่าเป็นบวก
Negative (N)	หมายถึง ผลลัพธ์จากการทำนายทำนายว่าเป็นลบ
True Positive (TP)	หมายถึง จำนวนผลของการทำนายเป็นค่าบวกและผลการทำนายถูกต้อง
False Positive (FP)	หมายถึง จำนวนของผลลัพธ์จากการทำนายเป็นบวกซึ่งทำนายไม่ถูกต้องเนื่องจากข้อมูลดังกล่าวเป็นลบ
True Negative (TN)	หมายถึง จำนวนของผลลัพธ์จากการทำนายเป็นลบและทำนายถูกต้อง
False Negative (FN)	หมายถึง จำนวนของผลลัพธ์จากการทำนายเป็นลบซึ่งทำนายไม่ถูกต้องเนื่องจากข้อมูลดังกล่าวเป็นบวก

ตารางที่ 1 เกณฑ์การทำนาย

		Actual Class	
		A	***Not A***
Predicted Class	***A***	True Positive (TP)	False Positive (FP)
	Not A	False Negative (FN)	True Negative (TN)

$$Precision(Pr) = \frac{TP}{TP+FP} \quad (3)$$

Precision (Pr) หมายถึง ผลลัพธ์จากการทำนายเป็นบวกจากข้อมูลที่ทำนาย

$$Recall(Re) = \frac{TP}{TP+FN} \quad (4)$$

Recall (Re) หมายถึง ผลลัพธ์จากการทำนายเป็นบวกจากข้อมูลที่เป็นบวก

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

Accuracy (Acc) หมายถึง ความแม่นยำในการทำนายของระบบ

$$F_1 = \frac{2*Precision*Recall}{Precision+Recall} \quad (6)$$

F_1 หมายถึง ประสิทธิภาพในการทำนายโดยรวมของระบบ

2.5.2 ตัวชี้วัดประสิทธิภาพการจำแนกประเภทแบบหลายฉลาก (Label-based Measurement)

การวัดประสิทธิภาพการจำแนกประเภทข้อมูลแบบหลายฉลาก [4, 5, 11, 13, 15] เป็นการสร้างตัวชี้วัดโดยการปรับปรุงการวัดประสิทธิภาพการจำแนกประเภทสองประเภท เพื่อให้สามารถวัดประสิทธิภาพการจำแนกประเภทแบบหลายฉลาก โดยการวัดประสิทธิภาพแบบนี้มีสองวิธี คือ Macro average และ Micro average

Macro average เป็นตัววัดประสิทธิภาพของระบบการจำแนกประเภทแบบหลายฉลาก โดยให้ความสำคัญต่อทุกกลุ่มข้อมูลเท่า ๆ กัน โดยจะทำการคำนวณค่า Precision Recall F1 ของแต่ละกลุ่มข้อมูลแล้วนำมาหาค่าเฉลี่ย

$$Precision_{Macro}\ (Pr_{Ma}) = \frac{1}{|C|}\sum_{i=1}^{|C|} \mathrm{Pr}_i \quad (7)$$

$$Recall_{Macro}\ (Re_{Ma}) = \frac{1}{|C|}\sum_{i=1}^{|C|} Re_i \quad (8)$$

$$F_{1,Ma} = \frac{1}{|C|}\sum_{i=1}^{|C|} F_{1,i} \quad (9)$$

Micro average เป็นตัววัดประสิทธิภาพของการจำแนกประเภทแบบหลายฉลาก โดยให้ความสำคัญต่อกลุ่มที่มีจำนวนมากกว่า โดยจะทำการหาผลรวมของค่าเฉลี่ยของ TP FP TN FN ก่อนแล้วนำมาคำนวณหาค่าประสิทธิภาพ โดยสามารถคำนวณได้ตามสมการที่ 10-12

$$Precision_{Micro}\ (Pr_{Mi}) = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} \quad (10)$$

$$Recall_{Micro}\ (Re_{Mi}) = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)} \quad (11)$$

$$F_{1,Mi} = \frac{2 * Pr_{Mi} * Re_{Mi}}{Pr_{Mi} + Re_{Mi}} \quad (12)$$

2.5.3 ตัวชี้วัดประสิทธิภาพการเลือกตอบตามตัวอย่าง (Example-based Measurement)

ตัววัดประสิทธิภาพนี้ [11] ใช้ในการวัดความสามารถในการเลือกจัดกลุ่มของตัวจำแนก โดยที่สามารถคำนวณได้จากสมการที่ 13-15

$$Precision_i\ (Pr_i) = \frac{|\hat{P}_i \cap \hat{T}_i|}{|\hat{P}_i|} \quad (13)$$

$$Recall_i\ (Re_i) = \frac{|\hat{P}_i \cap \hat{T}_i|}{|\hat{T}_i|} \quad (14)$$

$$F_{1,i}\ (F_{1,i}) = \frac{2 * Pr_i * Re_i}{Pr_i + Re_i} \quad (15)$$

P_i แทนเซตของประเภทที่ทำนายได้ของข้อมูลตัวที่ i

T_i แทนเซตของประเภทที่ถูกต้องของข้อมูลตัวที่ i

$\hat{P}_i$ แทนเซตของ P_i รวมกับเซตของประเภทบรรพบุรุษทั้งหมดของเซต P_i

$\hat{T}_i$ แทนเซตของ T_i รวมกับเซตของประเภทบรรพบุรุษทั้งหมดของเซต T_i

บทที่ 3
งานวิจัยที่เกี่ยวข้อง

3.1 งานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทแบบหลายฉลาก

รีแซมพลิ่งซัพพอร์ตเวคเตอร์แมชชีน (Resampling Support Vector Machine, R-SVM) ได้ถูกพัฒนาโดย P. Vateekul, S. Dendamrongvit และ M. Kubat [4] ซึ่งใช้งานซัพพอร์ตเวคเตอร์แมชชีนร่วมกับ วิธีการปรับค่าขีดแบ่งเพื่อที่จะแก้ปัญหาเรื่องความไม่สมดุลของข้อมูล โดยที่งานวิจัยชิ้นนี้จัดเก็บค่าขีดแบ่งที่ดีที่สุดจากโครงสร้างการจำแนก และนำไปปรับค่าขีดแบ่ง เพื่อนำไปใช้งานร่วมกับข้อมูลชุดสอนที่สร้างมาจากข้อมูลทั้งหมด จากนั้นทำการจัดเก็บค่าขีดแบ่งที่ดีที่สุดเพื่อนำมาหาค่าเฉลี่ย และนำไปใช้ในการปรับปรุงโครงสร้างของตัวจำแนก ซึ่งวิธีนี้สามารถเพิ่มประสิทธิภาพของการจำแนกประเภทแบบหลายฉลากได้อย่างมากและสามารถแก้ปัญหาความไม่สมดุลของข้อมูลได้อีกด้วย โดยผลการทดลองของงานวิจัยนี้แสดงให้เห็นว่า วิธีนี้สามารถเพิ่มประสิทธิภาพของการจำแนกประเภทแบบหลายฉลากได้เป็นอย่างมาก อีกทั้งยังสามารถแก้ปัญหาความไม่สมดุลของข้อมูลได้อีกด้วย อย่างไรก็ตามเนื่องจากงานวิจัยชิ้นนี้ไม่ได้ใช้ วิธีแซมพลิ่งในการแก้ไขปัญหาความไม่สมดุลของข้อมูล แต่ใช้วิธีการปรับค่าขีดแบ่ง จึงไม่นำไปใช้เป็นตัวเปรียบเทียบในการทดลอง

S. Dendamrongvit และ M. Kubat [19] ใช้วิธีเลือกลักษณะ เพื่อเลือกลักษณะที่มีความสำคัญของข้อมูลมาใช้ และใช้วิธีอันเดอร์แซมพลิ่งสุ่มลบข้อมูลเพื่อแก้ปัญหาความไม่สมดุลของข้อมูล โดยงานวิจัยชิ้นนี้ได้ทำการเปรียบเทียบการทำงาน 3 รูปแบบ ประกอบด้วย การจำแนกประเภทแบบหลายฉลาก การจำแนกประเภทแบบหลายฉลากประยุกต์ใช้งานวิธีเลือกลักษณะ และการจำแนกประเภทแบบหลายฉลากที่ประยุกต์ใช้ วิธีการเลือกลักษณะร่วมกับ วิธีอันเดอร์แซมพลิ่ง โดยที่ผลของการทดลองนั้นสามารถแสดงให้เห็นว่า เมื่อใช้งานวิธีอันเดอร์แซมพลิ่งจะสามารถเพิ่มประสิทธิภาพในการทำงานของระบบได้เป็นอย่างมาก ซึ่งวิธีการนี้จะถูกใช้เป็นตัวเปรียบเทียบในการทดลอง

ทวินซัพพอร์ตเวคเตอร์แมชชีน (Twin-Support Vector Machine, Twin-SVM) ได้ถูกนำเสนอโดย B. Zhang, X. Xu, และ J. Su [18] เป็นการสร้างซัพพอร์ตเวคเตอร์แมชชีน 2 ตัว และนำผลของการจำแนกที่ได้มารวมกัน เพื่อเลือกผลการจัดกลุ่ม เช่น มีข้อมูลประกอบด้วยกลุ่ม C1 และ C2 โดยวิธีการนี้จะทำการสร้างตัวจำแนกสำหรับตัวจำแนกกลุ่ม C1 จะให้ข้อมูลที่อยู่ใน C1 เป็นกลุ่มข้อมูลตัวอย่างบวกและ ให้ข้อมูลที่อยู่ในกลุ่ม C1-C2 เป็นข้อมูลตัวอย่างลบและตัวจำแนกกลุ่ม C2 จะ

ให้ข้อมูลที่อยู่ในกลุ่ม C2 เป็นข้อมูลกลุ่มตัวอย่างบวกและ ให้ข้อมูลที่อยู่ใน C2-C1 เป็นข้อมูลตัวอย่างลบ จากนั้นทำการเก็บผลการจำแนกของทั้งสองตัวจำแนก เพื่อทำการเลือกว่าข้อมูลนั้นอยู่ในกลุ่มใดอีกครั้ง แต่อย่างไรก็ตาม วิธีนี้มีข้อเสียคือ เมื่อนำตัวจำแนกมาใช้งานร่วมกับข้อมูลชุดทดสอบนั้น จะมีข้อมูลที่ไม่ได้อยู่ในกลุ่มเดียวกับตัวจำแนกนั้น ๆ เข้ามาอยู่ด้วย จึงเป็นสาเหตุทำให้เกิดเป็นปัญหาจำแนกกลุ่มผิดพลาดขึ้น อีกทั้งวิธีนี้ยังใช้เวลาในการทำงานเป็นสองเท่าอีกด้วย ต่อมาทวินซัพพอร์ตเวกเตอร์แมชชีนได้ถูกประยุกต์ใช้งานร่วมกับวิธีการเรียนรู้เบย์แบบง่าย (Naïve Bayes, NB) เพื่อลดระยะเวลาในการทำงานของระบบ โดยในขั้นตอนการสอนตัวจำแนกนั้น จะทำการสอนตัวจำแนกแบบวิธีการเรียนรู้เบย์แบบง่ายทั้งหมดที่เป็นไปได้ และสอนตัวจำแนกแบบทวินซัพพอร์ตเวกเตอร์แมชชีนทุกคู่ที่มีความเป็นไปได้ และทำการทดสอบกับข้อมูลชุดทดสอบ โดยเริ่มทดสอบกับตัวจำแนกแบบวิธีการเรียนรู้เบย์แบบง่ายก่อน ด้วยการกำหนัดค่าขีดแบ่งขึ้นมา เพื่อทำการเลือกตัวจำแนกโดยวิธีการเรียนรู้เบย์แบบง่ายที่มีค่าความเป็นไปได้สูงกว่าค่าขีดแบ่งมาใช้งาน และในส่วนของตัวจำแนกที่ไม่ถูกเลือกนั้นจะใช้งานตัวจำแนกแบบทวินซัพพอร์ตเวคเตอร์แมชชีน ซึ่งวิธีนี้นั้นสามารถทำงานได้เร็วกว่าวิธีทวินซัพพอร์ตเวคเตอร์แมชชีน เป็นอย่างมากและมีประสิทธิภาพสูงอีกด้วย อย่างไรก็ตาม งานวิจัยนี้ใช้วิธีรวมทั้งหมด จึงไม่นำไปใช้เป็นตัวเปรียบเทียบในผลการทดลอง

S. Daengduang และ P. Vateekul [8] ได้พัฒนาวิธีการจำแนกหลายประเภทด้วยวิธีการจำแนกแบบหนึ่งต่อหนึ่ง โดยใช้งานซัพพอร์ตเวคเตอร์แมชชีน ซึ่งงานวิจัยนี้ได้จำแนกปัญหาเป็นสองประเภท คือ ปัญหาที่มีข้อมูลที่ไม่เกี่ยวข้องปรากฏขึ้นเมื่อมีการใช้งานวิธีการจำแนกแบบหนึ่งต่อหนึ่งที่มีการประยุกต์ใช้ ซัพพอร์ตเวคเตอร์แมชชีนแล้วนำไปใช้กับข้อมูลหลายฉลาก และปัญหาความไม่สมดุลของข้อมูลที่มักจะปรากฏขึ้นในบางตัวจำแนก ซึ่งงานวิจัยชิ้นนี้ทำการแก้ไขปัญหาการพบข้อมูลที่ไม่เกี่ยวข้องปรากฏขึ้น โดยการให้ความสนใจผลของการจำแนกกลุ่มข้อมูลชุดตัวอย่างลบ เพื่อนำไปใช้ในการลดจำนวนของข้อมูลที่ไม่เกี่ยวข้องซึ่งมักจะปรากฏขึ้น และทำการประยุกต์ใช้วิธีอันเดอร์แซมพลิ่งเพื่อทำการแก้ปัญหาความไม่สมดุลของข้อมูลที่เกิดขึ้นในบางตัวจำแนก

J. Fürnkranz E. Hüllermeier E. L. Mencía และ K. Brinker [22] ได้พัฒนาวิธีการจำแนกประเภทแบบหลายฉลากโดยทำการประยุกต์ใช้งานวิธีการจำแนกแบบหนึ่งต่อหนึ่ง โดยที่งานวิจัยชิ้นนี้จะทำการจับคู่กลุ่มข้อมูลทั้งหมดที่เป็นไปได้เพื่อนำมาใช้ในการสร้างตัวจำแนกสองประเภท โดยที่ผลการจำแนกที่ได้รับจากแต่ละตัวจำแนกจะถูกจำแนกให้อยู่ในรูปของข้อมูลแบบหลายฉลาก โดยใช้ค่าขีดแบ่งในการเลือกตอบข้อมูลแบบหลายฉลากได้ ซึ่งวิธีการนี้จะถูกใช้เป็นตัวเปรียบเทียบในการทดลอง

3.2 งานวิจัยที่เกี่ยวข้องในด้านการประยุกต์ใช้งานระบบประมวลผลแบบกระจาย

F. O. Catak และ M. E. Balaban [21] ได้ประยุกต์ใช้งานระบบประมวลผลแบบกระจายร่วมกับวิธีการจำแนกสองประเภทโดยใช้งานซัพพอร์ตเวกเตอร์แมชชีน ซึ่งงานวิจัยนี้ได้นำระบบประมวลผลแบบกระจายมาประยุกต์ใช้โดยการใช้วิธีการแมพ เพื่อทำการแบ่งข้อมูลชุดสอนออกเป็นหลาย ๆ ชุด แล้วทำการสร้างโครงสร้างการจำแนกแบบสองประเภทด้วยซัพพอร์ตเวคเตอร์แมชชีนและนำไปทดสอบกับข้อมูลชุดทดสอบ เพื่อใช้ในการหาโลคอลซัพพอร์ตเวคเตอร์ (Local Support Vector) เพื่อนำไปใช้ในการคำนวณาหาค่าโกบอลซัพพอร์ตเวคเตอร์ (Global Support Vector) โดยการหาค่าเฉลี่ย ซึ่งงานวิจัยนี้มีเป้าหมายเพื่อเพิ่มความเร็วในการประมวลผลข้อมูล โดยข้อมูลที่ใช้ในการทดลองนั้นจะเป็นข้อมูล Hand written Digits ในชุดข้อมูลนี้มีข้อมูลทั้งหมด 5,620 ตัวอย่าง

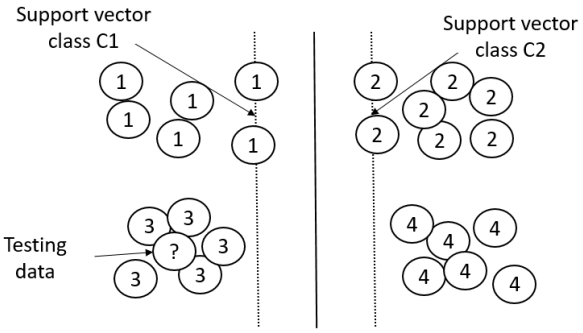
C. Y. Lin C. H. Tsai C. P. Lee และ C. J. Lin [12] ได้ประยุกต์ใช้งานระบบประมวลผลแบบกระจายร่วมกับวิธีการจำแนกแบบสองประเภท โดยนำวิธีซัพพอร์ตเวคเตอร์แมชชีนมาประยุกต์ใช้งาน งานวิจัยชิ้นนี้ได้นำระบบประมวลผลแบบกระจายมาใช้ในการแบ่งส่วนข้อมูลชุดสอนออกเป็นข้อมูลชุดสอนชุดเล็ก ๆ หลาย ๆ ชุด และนำวิธีการซัพพอร์ตเวคเตอร์แมชชีนมาใช้ในการสร้างโครงสร้างการจำแนก แล้วนำข้อมูลชุดทดสอบมาใช้งานกับโครงสร้างการจำแนก เพื่อใช้ในการหา โลคอลเวต (Local Weight) ของซัพพอร์ตเวคเตอร์ และนำมาใช้ในการคำนวณหาค่าโกบอลเวต (Global Weight) ของซัพพอร์ตเวคเตอร์ งานวิจัยชิ้นนี้นั้นมีเป้าหมายเพื่อเพิ่มความเร็วในการประมวลผล โดยใช้ระบบประมวลผลแบบกระจายเท่านั้น ข้อมูลที่ใช้ในการทดลองนั้น ประกอบด้วยข้อมูลทั้งหมด 55,000 ตัวอย่าง โดยแบ่งออกเป็นข้อมูลชุดสอน 50,000 ตัวอย่าง และข้อมูลชุดทดสอบ 5,000 ตัวอย่าง

J. Maillo I. Triguero และ F. Herrera [13] ได้ประยุกต์ใช้งานระบบประมวลผลแบบกระจายกับวิธีเพื่อนบ้านใกล้สุดจำนวน K ตัว (K-Nearest Neighbors, KNN) งานวิจัยชิ้นนี้ได้นำระบบประมวลผลแบบกระจายมาใช้ในการแบ่งส่วนข้อมูลชุดสอนออกเป็นข้อมูลชุดสอนชุดเล็ก ๆ หลาย ๆ ชุด และนำข้อมูลชุดสอนที่ได้ไปสร้างโครงสร้างการจำแนกด้วยวิธีเพื่อนบ้านใกล้สุดจำนวน K ตัว จากนั้นนำโครงสร้างการจำแนกที่ได้ไปทำการทดสอบกับข้อมูลชุดทดสอบ แล้วทำการรามผลทั้งหมดให้อยู่ในรูปแบบของไฟล์ ๆ เดียว งานวิจัยที่มีเป้าหมายเพื่อเพิ่มความเร็วในการทำงานของวิธีที่มีอยู่ก่อนหน้า ซึ่งการทดลองนี้ได้ทำการทดลองกับชุดข้อมูล Poker hand ซึ่งข้อมูลชุดนี้มีตัวอย่างทั้งหมด 1,025,010 ตัวอย่าง

บทที่ 4
แนวคิดและวิธีการดำเนินงาน

ในปัจจุบันข้อมูลมีการเพิ่มขนาด และความซับซ้อนมากขึ้นจนกระทั่งไม่สามารถทำการประมวลผลได้ด้วยการใช้คอมพิวเตอร์เพียงเครื่องเดียว เป็นเหตุให้ผู้วิจัยนำเสนอในงานวิจัยนี้ประยุกต์ใช้การจำแนกเประเภทแบบหนึ่งต่อหนึ่งด้วยการใช้วิธีซัพพอร์ตเวคเตอร์แมชชีนร่วมกับระบบประมวลผลแบบกระจายเพื่อใช้ในการจำแนกชุดข้อมูลหลายฉลาก งานวิจัยชิ้นนี้ได้มุ่งเน้นไปที่วิธีการประยุกต์ใช้งานระบบประมวลผลแบบกระจาย สปาร์ค เพื่อลดระยะเวลาในการประมวลผลของชุดข้อมูลที่มีกลุ่มข้อมูลขนาดใหญ่และมีความซับซ้อนสูง โดยที่ระบบประมวลผลแบบกระจาย สปาร์คนั้นได้รับการพิสูจน์จากผู้พัฒนาระบบว่าเป็นระบบประมวลผลแบบกระจายที่มีประสิทธิภาพในด้านความเร็วสูงที่สุดในปัจจุบัน

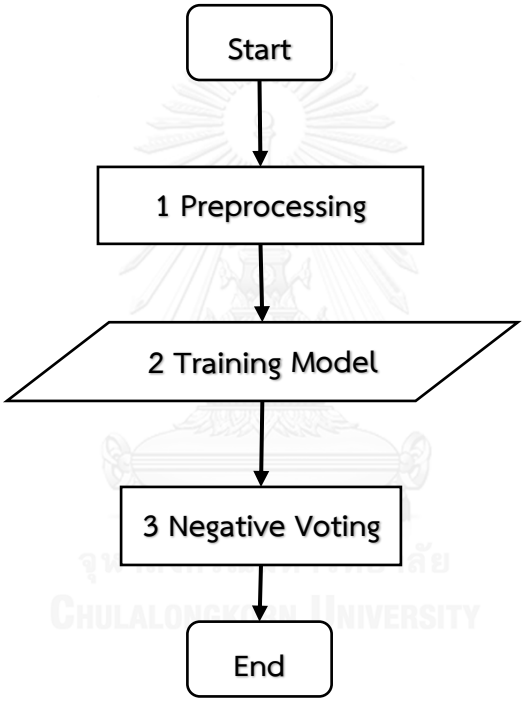
การประยุกต์ใช้ วิธีการจำแนกประเภทแบบหลายฉลากแบบหนึ่งต่อหนึ่งกับซัพพอร์ตเวคเตอร์แมชชีนนั้น ผู้วิจัยพบว่าเมื่อทำการทดสอบโครงสร้างการจำแนกแบบหนึ่งต่อหนึ่ง ข้อมูลทดสอบที่ไม่มีความเกี่ยวข้องกับโครงสร้างตัวจำแนกนั้นอาจจะปรากฏขึ้นมาในตัวจำแนก ซึ่งอาจจะนำไปสู่ปัญหาการจำแนกผิดพลาดได้ดังรูปที่ 7 โดยผู้วิจัยได้นำเสนอวิธีการแก้ไขปัญหาที่เกิดขึ้นนี้โดยทำการเลือกกลุ่มข้อมูลชุดตัวอย่างลบ เพื่อใช้ในการตัดข้อมูลที่ไม่เกี่ยวข้องออกไป เพื่อแก้ปัญหาการจำแนกผิดพลาด และใช้ค่าขีดแบ่งเพื่อทำให้สามารถเลือกตอบได้หลายกลุ่มข้อมูล โดยที่งานวิจัยชิ้นนี้จะแบ่งออกเป็น 2 ส่วน ดังนี้ การสร้างตัวจำแนกแบบหนึ่งต่อหนึ่ง และการประยุกต์ใช้งานระบบประมวลผลแบบกระจาย



รูปที่ 7 ตัวจำแนก C1vsC2 ที่ปรากฏข้อมูลกลุ่มอื่นเข้ามา

4.1 การสร้างตัวจำแนกแบบหนึ่งต่อหนึ่งและการจำแนกประเภท

ด้วยข้อมูลที่พบในปัจจุบันนั้น จะมีข้อมูลอยู่ในกลุ่มแต่ละกลุ่มจำนวนไม่เท่ากัน ซึ่งมักจะทำให้เกิดปัญหาความไม่สมดุลของข้อมูลขึ้น ซึ่งปัญหานี้มักจะนำไปสู้ปัญหาการจำแนกข้อมูลผิดพลาดได้อีกด้วย โดยที่งานวิจัยนี้ได้ประยุกต์ใช้วิธีแซมพลิ่งเพื่อใช้ในการแก้ปัญหาความไม่สมดุลของข้อมูล โดยวิธีแซมพลิ่งนั้นสามารถแบ่งออกเป็น 2 ประเภท ประกอบด้วย วิธีอันเดอร์แซมพลิ่ง และวิธีโอเวอร์แซมพลิ่ง วิธีอันเดอร์แซมพลิ่งจะทำการสุ่มลบข้อมูลเสียงข้างมากให้เหลือจำนวนเท่ากับข้อมูลเสียงข้างน้อย และวิธีโอเวอร์แซมพลิ่งจะทำการสุ่มเพิ่มจำนวน ข้อมูลเสียงข้างน้อยให้มีจำนวนเท่ากับข้อมูลเสียงข้างมาก ซึ่งขั้นตอนนี้ประกอบด้วย 3 ขั้นตอนดังรูปที่ 8



รูปที่ 8 แผนภาพการทำงานในการสร้างตัวจำแนกแบบหนึ่งต่อหนึ่งและจำแนกประเภท

4.1.1 การจัดเตรียมข้อมูล (Preprocessing)

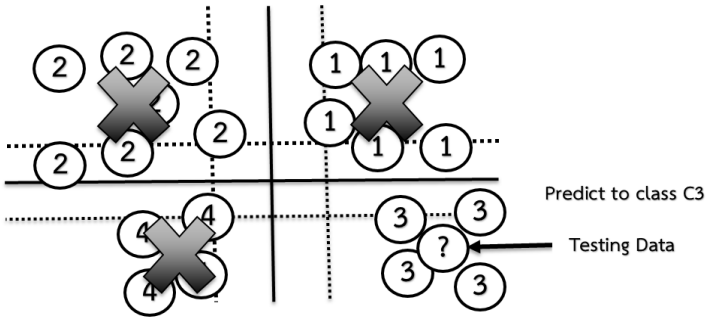
ในขั้นตอนนี้จะเป็นขั้นตอนที่ใช้ในการสร้างข้อมูลชุดสอน เพื่อนำไปใช้สร้างโครงสร้างตัวจำแนก โดยขั้นตอนนี้จะทำการจับคู่ข้อมูลที่อยู่ในกลุ่มแต่ละกลุ่ม เพื่อสร้างเป็นข้อมูลชุดสอน ซึ่งงานวิจัยชิ้นนี้เลือกใช้วิธีการเตรียมข้อมูลด้วย วิธีอันเดอร์แซมพลิ่งเพื่อนำไปใช้ในการเลือกข้อมูลชุดตัวอย่างบวก และข้อมูลชุดตัวอย่างลบด้วยการสุ่มลบข้อมูลที่อยู่ในกลุ่มเสียงข้างมาก จนกระทั่งมีจำนวนข้อมูลเท่ากับกลุ่มข้อมูลเสียงข้างน้อย เนื่องจากวิธีการนี้ใช้ระยะเวลาในการประมวลผลน้อยกว่าวิธีการโอเวอร์แซมพลิ่ง และมีประสิทธิภาพใกล้เคียงกันสามารถอ้างอิงได้จากผลการทดลองที่ 5.2

4.1.2 การสร้างโครงสร้างการจำแนก (Training Model)

ขั้นตอนการสร้างโครงสร้างการจำแนก จะมีการนำข้อมูลชุดสอนที่ถูกสร้างขึ้นในขั้นตอนที่ 4.1.1 มาใช้สร้างโครงสร้างการจำแนกด้วยวิธีการซัพพอร์ตวคเตอร์แมชชีน ซึ่งเลือกใช้งานเคอร์เนล RBF ในการสร้างโครงสร้างการจำแนก เนื่องจากเป็นเคอร์เนลที่มีประสิทธิภาพในการจำแนกสูงที่สุด โดยอ้างอิงจากผลการทดลองที่ 5.1 โดยการเลือกใช้เคอร์เนล RBF จำเป็นต้องทำการปรับค่าพารามิเตอร์ เพื่อให้ได้ค่าที่เหมาะสมในการสร้างตัวโครงสร้างการจำแนก โดยที่ค่าที่ใช้ในการปรับนั้นคือ ค่า gamma ซึ่งในการปรับค่านั้นจะเลือกใช้ค่าพื้นฐาน ซึ่งสามารถคำนวณได้จาก 1ต่อจำนวนลักษณะข้อมูล โดยทำการปรับค่าเพิ่มขึ้นจากค่าพื้นฐาน 4 ค่า ดังนี้ ค่าพื้นฐาน +0.01 ค่าพื้นฐาน +0.1 ค่าพื้นฐาน +0.5 และค่าพื้นฐาน +1

4.1.3 การคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้อง (Unrelated class filtering)

ขั้นตอนการคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้องข้อมูลชุดทดสอบจะถูกมาใช้ในการทดสอบโครงสร้างการจำแนก โดยที่ผลของการจำแนกที่ได้จากแต่ละตัวจำแนกนั้นจะถูกรวบรวมไว้ในที่เดียวกัน แบ่งผลการจำแนกออกเป็นกลุ่ม ๆ ตามกลุ่มข้อมูลทั้งหมดที่มี ซึ่งผลการจำแนกนั้นจะต้องเป็นผลการทดลองของตัวจำแนกที่มีความเกี่ยวข้องกับกลุ่มข้อมูลใด ๆ โดยจัดเก็บผลของการจำแนกเฉพาะข้อมูลตัวอย่างลบเท่านั้น และทำการหาค่าผลรวมของตัวจำแนกที่มีการเลือกตอบเป็นข้อมูลตัวอย่างลบ จากนั้นทำการเปรียบเทียบผลรวมของข้อมูลตัวอย่างลบกับค่าขีดแบ่ง (θ) ถ้าผลรวมมีค่ามากกว่าค่าขีดแบ่งจะจำแนกเป็นตัวอย่างบวก ดังรูปที่ 9



รูปที่ 9 ตัวอย่างการจำแนกด้วยวิธี Unrelated class filtering

ถ้าค่าขีดแบ่งมีค่าน้อยกว่าผลรวมข้อมูลจะไม่ถูกจำแนกเป็นตัวอย่างบวก ตัวอย่าง เช่น มีกลุ่มข้อมูล 4 กลุ่มคือ C1-C4 ทำการสร้างตัวจำแนกแบบหนึ่งต่อหนึ่งของกลุ่ม C1 จะได้ตัวจำแนกดังนี้ C1vsC2 C1vsC3 C1vsC4 ถ้าหากตัวจำแนก C1vsC2 และ C1vsC3 ถูกจำแนกเป็นข้อมูลตัวอย่างลบ และ C1vsC4 ถูกจำแนกเป็นข้อมูลตัวอย่างบวก วิธีการนี้จะทำการเก็บผลของการจำแนกที่เป็นข้อมูล

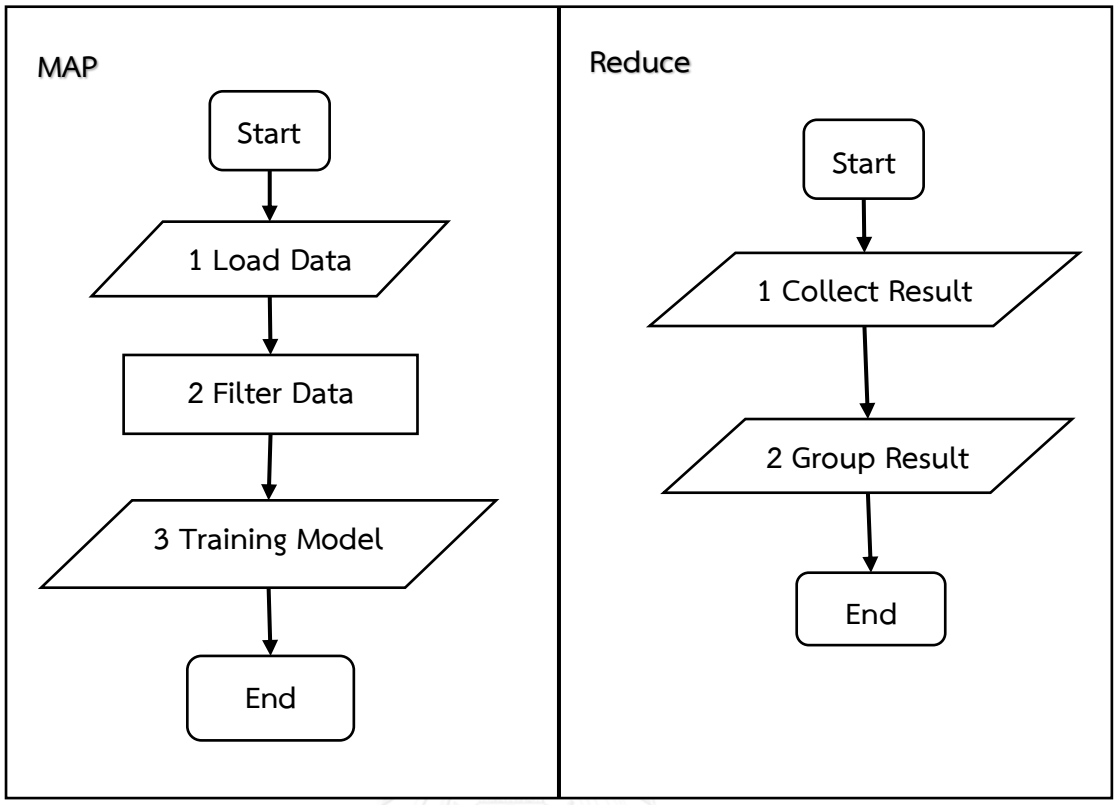
ตัวอย่างลบ และทำการหาผลรวมเพื่อเปรียบเทียบกับค่าขีดแบ่ง ถ้าค่าขีดแบ่งมีค่าสูงกว่าผลรวมจะจำแนกข้อมูลชุดนั้นให้เป็น C1 สามารถดูตัวอย่างเพิ่มเติมได้ในตารางที่ 2

ตารางที่ 2 ตัวอย่างวิธีการจำแนก

ตัวอย่าง	***C1vsC2***	***C1vsC3***	***C1vsC4***	**กลุ่มที่ทำนายได้**
1	**ไม่ใช่ C1**	**ไม่ใช่ C1**	**ไม่ใช่ C1**	**ไม่เป็น C1**
2	**ไม่ใช่ C2**	**ไม่ใช่ C3**	**ไม่ใช่ C1**	**C4**
3	**ไม่ใช่ C2**	**ไม่ใช่ C3**	**ไม่ใช่ C4**	**C1**

4.2 การประยุกต์ใช้งานระบบประมวลผลแบบกระจาย

เนื่องจากข้อมูลในปัจจุบันมีขนาดใหญ่ และมีกลุ่มจำนวนมาก จึงเป็นเหตุให้ไม่สามารถใช้วิธีการจำแนกประเภทหลายฉลากแบบหนึ่งต่อหนึ่งได้ ด้วยการใช้งานคอมพิวเตอร์เครื่องเดียว ตัวอย่าง เช่น ข้อมูลมีกลุ่มอยู่ทั้งหมด 101 กลุ่ม หากใช้วิธีการจำแนกประเภทหลายฉลากแบบหนึ่งต่อหนึ่งแล้ว วิธีการนี้จำเป็นต้องสร้างตัวจำแนกมากถึง 5050 ตัว เป็นเหตุให้ไม่สามารถประมวลผลโดยใช้งานคอมพิวเตอร์เครื่องเดียว เป็นต้น งานวิจัยชิ้นนี้จึงได้นำเสนอวิธีการประยุกต์ใช้ระบบประมวลผลแบบกระจายสปาร์ค เพื่อแก้ไขปัญหาการใช้งานวิธีจำแนกประเภทหลายฉลากแบบหนึ่งต่อหนึ่งกับข้อมูลที่มีกลุ่มจำนวนมาก อย่างไรก็ตามระบบปรมวลผลแบบการจายสปาร์คนั้นมีลักษณะการทำงานเป็นแบบเริ่มทำงานก่อนจะเสร็จก่อน ซึ่งระบบนี้จะทำการจัดการทรัพยากรที่ใช้ในการประมวลผลโดยอัตโนมัติ โดยการใช้งานระบบนี้ร่วมกับวิธีการจำแนกแบบหนึ่งต่อหนึ่งจะทำให้ไม่สามารถใช้งานประสิทธิภาพทรัพยากรของระบบได้อย่างเต็มที่ งานวิจัยชิ้นนี้จึงได้นำเสนอวิธีการแบ่งส่วนการทำงานออกเป็นหลาย ๆ ส่วน เพื่อทำให้ใช้ทรัพยากรในการประมวลผลได้อย่างเต็มที่ และลดระยะเวลาในการประมวลผลลงอีกด้วย ซึ่งขั้นตอนนี้จะแบ่งส่วนการประยุกต์ใช้งานระบบประมวลผลแบบกระจายออกเป็น 2 ส่วน ดังรูปที่ 10 ได้แก่ แมพ และรีดิวซ์



รูปที่ 10 การประยุกต์ใช้งานระบบประมวลผลแบบกระจาย

4.2.1 แมพ

ในส่วนนี้จะประกอบด้วยขั้นตอนการทำงาน 3 ขั้นตอน ดังรูปที่ 10

4.2.1.1 การจัดเก็บข้อมูล (Load Data)

ในขั้นตอนนี้จะเป็นการเปลี่ยนแปลงข้อมูล เพื่อทำการจัดเก็บข้อมูลลงไปยังหน่วยความจำ โดยที่จะทำการเก็บข้อมูลชุดสอน และข้อมูลชุดทดสอบไว้ภายในหน่วยความจำ จากนั้นทำการอ่านข้อมูลระบุที่อยู่ของแต่ละกลุ่มข้อมูลในข้อมูลชุดสอน

4.2.1.2 การคัดกรองข้อมูล (Filter Data)

ในขั้นตอนนี้จะเป็นการคัดกรองข้อมูลชุดสอน โดยใช้ข้อมูลระบุที่อยู่ของแต่ละกลุ่มข้อมูล ซึ่งทำการบันทึกค่าไว้ในขั้นตอนการทำงานที่ 4.2.1.1 ในขั้นตอนนี้จะจับคู่กลุ่มข้อมูลที่เป็นไปได้และทำการรวมข้อมูลระบุที่อยู่ข้อมูลชุดสอนของแต่ละคู่เข้าด้วยกัน จากนั้นทำการคัดกรองข้อมูลชุดสอนที่ถูกจัดเก็บอยู่ในหน่วยความจำ โดยใช้ข้อมูลระบุที่อยู่ที่สร้างขึ้นใหม่ แล้วนำมาใช้คัดกรองข้อมูลชุดสอนที่ไม่เกี่ยวข้องออกไปและนำมาใช้ในการสร้างข้อมูลชุดสอนของแต่ละตัวจำแนก โดยที่ข้อมูล

ชุดสอนของแต่ละตัวจำแนกนั้นจะอยู่ในลักษณะของ คีย์/แวลู โดยที่ คีย์ คือ กลุ่มข้อมูล และแวลู คือ ลักษณะของข้อมูลนั้น ๆ

4.2.1.3 การสร้างโครงสร้างการจำแนก (Training Model)

ในขั้นตอนนี้เป็นการนำข้อมูลชุดสอนของแต่ละตัวจำแนก ซึ่งได้ทำการสร้างไว้ในขั้นตอนที่ 4.2.1.2 มาใช้งานเพื่อสร้างโครงสร้างการจำแนกแบบหนึ่งต่อหนึ่งของแต่ละตัวจำแนก ด้วยการใช้วิธีซัพพอร์ตเวคเตอร์แมชชีน หลังจากการสร้างโครงสร้างการจำแนกแบบหนึ่งต่อหนึ่งเสร็จสิ้น ข้อมูลชุดทดสอบจะถูกนำมาใช้งานเพื่อทำการทดสอบโครงสร้างการจำแนกของแต่ละตัวจำแนก

4.2.2 รีดิวซ์

ส่วนของการรีดิวซ์จะประกอบด้วย 2 ขั้นตอน ดังรูปที่ 10

4.2.2.1 การรวมผลการจำแนกของแต่ละตัวจำแนก (Collect Result)

ในขั้นตอนนี้จะทำการรีดิวซ์เพื่อทำการรวมผลของการทดสอบโครงสร้างการจำแนกแบบหนึ่งต่อหนึ่ง ของตัวจำแนกแต่ละตัวหลังจากใช้งานข้อมูลชุดทดสอบ โดยลักษณะของผลการจำแนกที่ได้นั้น คีย์/แวลู จะได้คีย์ คือ รหัสของเอกสาร แวลู คือ ผลของการจำแนก ซึ่งผลการจำแนกนั้นจะถูกรวบรวมเข้าด้วยกัน โดยแบ่งผลการจำแนกตามกลุ่มข้อมูล ซึ่งผลการจำแนกที่ถูกเก็บไว้นั้นจะต้องเป็นผลของตัวจำแนกมีความเกี่ยวข้องกับกลุ่มข้อมูลใด ๆ ด้วย จากนั้น ขั้นตอนในวิธีการที่ 4.1.3 จะถูกนำมาใช้ในการตัดกลุ่มข้อมูลที่ไม่เกี่ยวข้องออกไปและทำการเลือกตอบ

4.2.2.2 การรวมผลขั้นสุดท้าย (Group Result)

ในขั้นตอนนี้จะทำการรีดิวซ์เพื่อรวมผลการเลือกตอบในแต่ละกลุ่มข้อมูลที่ได้จากรวมผลการจำแนกไว้ในขั้นตอนที่ 4.2.2.1 ซึ่งทำการรวมผลการจำแนกด้วยการใช้งาน คีย์คือ รหัสของเอกสาร เพื่อทำการรวมคำตอบของหลาย ๆ กลุ่มข้อมูลที่มีความเกี่ยวข้องกับเอกสารนั้น ๆ เข้าด้วยกันซึ่งผลการจำแนกที่ได้ขั้นตอนนี้จะออกมาในรูปแบบของคีย์/แวลูโดยที่ คีย์ คือ รหัสของเอกสาร และแวลู คือ ผลการจำแนกกลุ่มข้อมูลแบบหลายฉลาก

บทที่ 5
การทดลองและวิเคราะห์ผล

ในงานวิจัยชิ้นนี้ผู้วิจัยแบ่งการทดลองออกเป็น 4 ส่วน ประกอบด้วย

1) การทดลองเลือกเคอร์เนลที่ดีที่สุดของซัพพอร์ตเวคเตอร์แมชชีนเพื่อนำมาใช้ในการสร้างโครงสร้างการจำแนก โดยการทดลองนี้จะใช้งาน ชุดข้อมูลขนาดเล็ก 1 ชุดข้อมูล และชุดข้อมูลขนาดกลาง 1 ชุดข้อมูล

2) การทดลองเปรียบเทียบประสิทธิภาพและระยะเวลาในการประมวลผลของวิธีโอเวอร์แซมพลิ่งกับวิธีอันเดอร์แซมพลิ่ง เพื่อใช้ในการตัดสินใจเลือกวิธีที่มีความเหมาะสมมากที่สุด โดยการทดลองนี้จะใช้งานชุดข้อมูลขนาดเล็ก 1 ชุดข้อมูล และชุดข้อมูลขนาดกลาง 1 ชุดข้อมูล

3) การทดลองเพื่อเปรียบเทียบวิธีการที่ผู้วิจัยได้นำเสนอกับวิธีพื้นฐานประกอบด้วย วิธีการจำแนกแบบหนึ่งต่อทั้งหมดซึ่งประยุกต์ใช้งานวิธีอันเดอร์แซมพลิ่ง และวิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้วิธีจัดลำดับกลุ่มข้อมูล โดยการทดลองนี้จะใช้ ชุดข้อมูลทั้งหมด 6 ชุดข้อมูล [23]

4) การทดลองเพื่อใช้ในการวัดประสิทธิภาพของระยะเวลาในการประมวลผลของระบบประมวลผลแบบกระจาย โดยทำการเปรียบเทียบผลการทดลองประกอบด้วยงาน 1 2 3 และ 5 แมพ/รีดิวซ์ การทดลองนี้จะใช้ชุดข้อมูลทั้งหมด 6 ชุดข้อมูล โดยรายละเอียดของชุดข้อมูลที่นำมาใช้งานทั้งหมดนั้นจะเป็นดังตารางที่ 3 และอัตราการเกิดความไม่สมดุลของข้อมูลเมื่อทำการสร้างตัวจำแนกแบบหนึ่งต่อหนึ่งจะแสดงในตารางที่ 4 โดยที่ตารางนี้แสดงให้เห็นถึงอัตราความไม่สมดุลที่ประกฏขึ้นในการสร้างตัวจำแนกในแต่ละชุดข้อมูล โดยอัตราการเกิดความไม่สมดุลของข้อมูลนั้นสามารถคำนวณได้โดยใช้สมการที่ 16

$$Imbalance\ ratio = \frac{examples\ in\ majority\ class}{examples\ in\ minority\ calss} \quad (16)$$

ตารางที่ 3 รายละเอียดของแต่ละชุดข้อมูล

ชุดข้อมูล	จำนวนกลุ่มข้อมูล	จำนวนตัวอย่าง	จำนวนกลุ่มข้อมูลต่อเอกสาร
Yeast	14	2,417	4.237
Emotion	6	593	1.869
IMDB	28	120,919	2.000
Tmc2007	22	28,596	2.158
Birds	19	645	1.014
Rcv1v2	101	6,002	3.226

ตารางที่ 4 ความไม่สมดุลของข้อมูลในการสร้างตัวจำแนกแบบหนึ่งต่อหนึ่งของแต่ละชุดข้อมูล

ชุดข้อมูล	จำนวนตัวจำแนก	อัตราความไม่สมดุลมากที่สุด (max)	อัตราความไม่สมดุลน้อยที่สุด (min)	ค่าเฉลี่ยความไม่สมดุลของแต่ละตัวจำแนก (avg)
Yeast	91	72.428	1.009	12.633
Emotion	15	1.804	1.101	1.320
IMDB	378	136.668	1.003	9.539
Tmc2007	231	22.209	1.001	4.560
Birds	171	51.333	1.111	4.676
Rcv1v2	5,050	1,416.000	1.000	20.100

5.1 การทดลองเลือกเคอร์เนลที่ดีที่สุดของซัพพอร์ตเวคเตอร์แมชชีน

ในการทดลองนี้ ผู้วิจัยได้เลือกใช้งานชุดข้อมูลขนาดเล็ก 1 ชุดข้อมูลคือ Emotion และชุดข้อมูลขนาดกลาง 1 ชุดข้อมูล คือ Yeast โดยทำการเลือกเคอร์เนลเพื่อทำการทดลองทั้งหมด 3 เคอร์เนล ประกอบด้วย Linear RBF และ Polynomial โดยทำการเลือกเคอร์เนลที่มีประสิทธิภาพมากที่สุด เพื่อนำไปใช้ในการทดลองที่ 3 ซึ่งจะทำการวัดประสิทธิภาพของแต่ละเคอร์เนลโดยใช้ตัวชี้วัดดังนี้ Accuracy Label-based และ Example-based โดยการทดลองนี้จะทำการปรับค่าพารามิเตอร์ของซัพพอร์ตเวคเอตร์แมตชีนแต่ละเคอร์เนลดังนี้ สำหรับเคอร์เนล Linear จะปรับพารามิเตอร์ C ประกอบด้วย 0.5 1 5 10 และ 100 สำหรับเคอร์เนลPolynomial จะปรับพารามิเตอร์ degree ประกอบไปด้วย 1 3 5 10 และ 15 และสำหรับเคอร์เนล RBF จะทำการปรับค่าพารามิเตอร์ gamma

โดยค่าที่ใช้ในการปรับของแต่ละชุดข้อมูลนั้นจะแตกต่างกัน ซึ่งจะเลือกค่าพื้นฐานเป็นค่าแรก และทำการปรับเพิ่มจำนวนขึ้นเรื่อย ๆ โดยค่าพื้นฐานของ RBF นั้นจะสามารถคำนวณได้จาก 1ต่อจำนวนลักษณะข้อมูล ซึ่งค่าที่จะเพิ่มขึ้นมี 4 ค่า คือ ค่าพื้นฐาน + 0.01 ค่าพื้นฐาน + 0.1 ค่าพื้นฐาน + 0.5 และค่าพื้นฐาน + 1 โดยการทดลองนี้นั้นจะใช้วิธีการทดลองแบบ 3 fold cross-validation ซึ่งผลการทดลองได้ถูกแสดงให้เห็นในตารางที่ 5-7

ตารางที่ 5 ผลการทดลองประสิทธิภาพของแต่ละเคอร์เนลในเชิงความแม่นยำ ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบประสิทธิภาพระหว่างเคอร์เนลที่ดีที่สุดกับเคอร์เนลใด ๆ

Datasets	*Linear*	*Poly*	*RBF*
Yeast	0.742 (+1.1%)	0.697 (+7.1%)	**0.750**
Emotion	0.648 +(6.8%)	0.672 (+3.3%)	**0.695**
Average	0.695 (+3.7%)	0.684 (+5.3%)	**0.722**

ตารางที่ 5 กล่าวถึงการเปรียบเทียบผลการทดลองเชิงความแม่นยำของการทำนายแต่ละเคอร์เนลประกอบด้วย Linear Polynomial และ RBF ซึ่งแสดงให้เห็นว่า เคอร์เนล RBF นั้นมีความสามารถในการทำนายสูงที่สุดจากเคอร์เนลทั้งหมด 3 เคอร์เนล ซึ่งเคอร์เนล RBF มีประสิทธิภาพในการทำนายสูงกว่าเคอร์เนล Linear โดยเฉลี่ย 3.7% และ Polynomial โดยเฉลี่ย 5.3 %

ตารางที่ 6 ผลการทดลองประสิทธิภาพของแต่ละเคอร์เนลในเชิงการเลือกตอบหลายกลุ่ม ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบประสิทธิภาพระหว่างเคอร์เนลที่ดีที่สุดกับเคอร์เนลใด ๆ

Datasets	Label-Based Macro F_1			Label-Based Micro F_1		
	Linear	*Poly*	*RBF*	*Linear*	*Poly*	*RBF*
Yeast	0.618 (+3.1%)	0.393 (+38.3%)	**0.637**	0.607 (+1.8%)	0.408 (+33.9%)	**0.618**
Emotion	0.353 (+33.8%)	0.467 (+12.3%)	**0.533**	0.543 (+4.1%)	0.549 (+3.0%)	**0.566**
Average	0.485 (+17.1%)	0.430 (+26.5)	**0.585**	0.575 (+1.2%)	0.478 (+19.3%)	**0.592**

ตารางที่ 6 อธิบายความสามารถในการจัดกลุ่มข้อมูลแบบหลายประเภท โดยเปรียบเทียบประสิทธิภาพในการจัดกลุ่มของแต่ละเคอร์เนลประกอบด้วย Linear Polynomial และ RBF ซึ่งผลของการทดลองนี้แสดงให้เห็นว่า เคอร์เนล RBF นั้นมีประสิทธิภาพสูงที่สุดในการจำแนกประเภทแบบหลายฉลาก ซึ่งเคอร์เนล RBF มีประสิทธิภาพสูงกว่าเคอร์เนล Linear โดยเฉลี่ย 17.1 % และ Polynomial โดยเฉลี่ย 26.5 % ในตัวชี้วัด Macro-F_1 และมีประสิทธิภาพสูงกว่าเคอร์เนล Linear โดยเฉลี่ย 1.2 % และ Polynomial โดยเฉลี่ย 19.3 % ในตัวชี้วัด Micro-F_1

ตารางที่ 7 ผลการทดลองประสิทธิภาพของแต่ละเคอร์เนลในการเลือกคำตอบที่ถูกต้อง ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบประสิทธิภาพระหว่างเคอร์เนลที่ดีที่สุดกับเคอร์เนลใด ๆ

Datasets	Example-Based Macro F_1			Example-Based Micro F_1		
	Linear	*Poly*	*RBF*	*Linear*	*Poly*	*RBF*
Yeast	0.639 (+3.9%)	0.561 (+15.6%)	**0.665**	0.649 (+2.7%)	0.572 (+14.2%)	**0.667**
Emotion	0.353 (+33.8%)	0.467 (+12.4%)	**0.533**	0.542 (+21.4%)	0.515 (+25.3%)	**0.690**
Average	0.496 (+17.2%)	0.514 (+14.1%)	**0.599**	0.595 (+12.2%)	0.543 (+19.9%)	**0.678**

ตารางที่ 7 อธิบายถึงการเปรียบเทียบประสิทธิภาพของการจำแนกประเภทแบบหลายฉลากในด้านของความสามารถเลือกตอบกลุ่มที่อยู่ในผลเฉลยมากที่สุด โดยจะเป็นการเปรียบเทียบประสิทธิภาพในการทำงานของเคอร์เนล 3 เคอร์เนล ประกอบด้วย Linear Polynomial และ RBF โดยผลการทดลองในตารางนี้แสดงให้เห็นว่าความสามารถในการเลือกตอบของ RBF มีประสิทธิภาพสูงกว่าเคอร์เนลอื่น ๆ ซึ่งมีประสิทธิภาพสูงกว่าเคอร์เนล Linear โดยเฉลี่ย 17.2 % และ Polynomial โดยเฉลี่ย 14.1 % ในตัวชี้วัด Macro-F_1 และมีประสิทธิภาพสูงกว่าเคอร์เนล Linear โดยเฉลี่ย 12.2 % และ Polynomial โดยเฉลี่ย 19.9 % ในตัวชี้วัด Micro-F_1

จากผลการทดลองในตารางที่ 5-7 แสดงให้เห็นว่าเคอร์เนล RBF นั้นมีประสิทธิภาพสูงที่สุดทั้งในด้านของความแม่นยำ ความสามารถในการเลือกตอบหลายกลุ่มข้อมูล และความสามารถในการเลือกตอบตรงตามตัวอย่าง จากผลการทดลองนี้ทำให้ผู้วิจัยเลือกใช้เคอร์เนล RBF ในการทดลองที่ 5.3

5.2 การทดลองเลือกวัดประสิทธิภาพระหว่างโอเวอร์แซมพลิ่งและอันเดอร์แซมพลิ่ง

ในการทดลองนี้ ผู้วิจัยได้เลือกใช้งานชุดข้อมูลขนาดเล็ก 1 ชุดข้อมูล ประกอบด้วย Emotion และชุดข้อมูลขนาดกลาง 1 ชุดข้อมูล ประกอบด้วย Yeast โดยจะทำการใช้วิธีโอเวอร์แซมพลิ่ง เพื่อทำการสุ่มเพิ่มข้อมูลเสียงข้างน้อยและทำการสร้างสมดุลให้ข้อมูล ใช้วิธีอันเดอร์แซมพลิ่งเพื่อสร้างสมดุลระหว่างข้อมูลโดยการสุ่มลบข้อมูลที่มีเสียงข้างมากให้มีจำนวนเท่ากับข้อมูลเสียงข้างน้อย การทดลองนี้จะมีการวัดประสิทธิภาพการจำแนกของวิธีโอเวอร์แซมพลิ่งและวิธีอันเดอร์แซมพลิ่งเพื่อใช้ในการเลือกวิธีที่มีความเหมาะสมที่สุดไปใช้ในการทดลองที่ 3

ตารางที่ 8 ผลการทดลองประสิทธิภาพของวิธีโอเวอร์แซมพลิ่งและอันเดอร์แซมพลิ่งในเชิงความแม่นยำ ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบประสิทธิภาพระหว่างวิธีโอเวอร์แซมพลิ่งและวิธีอันเดอร์แซมพลิ่ง

Datasets	*Oversampling*	*Undersampling*
Yeast	**0.867 (-4.3%)**	0.830
Emotion	**0.656 (-2.9%)**	0.637
Average	**0.761 (-3.7%)**	0.733

ตารางที่ 8 แสดงผลของการเปรียบเทียบผลการทดลองระหว่างวิธีโอเวอร์แซมพลิ่งและอันเดอร์แซมพลิ่งในเชิงความแม่นยำของระบบ ซึ่งชี้ให้เห็นว่าการแก้ปัญหาความไม่สมดุลของข้อมูลด้วยการประยุกต์ใช้วิธีโอเวอร์แซมพลิ่งนั้นมีประสิทธิภาพสูงว่าวิธีอันเดอร์แซมพลิ่ง โดยวิธีโอเวอร์แซมพลิ่งมีประสิทธิภาพในการทำนายสูงกว่าวิธีอันเดอร์แซมพลิ่ง 3.7 %

ตารางที่ 9 ผลการทดลองประสิทธิภาพของวิธีโอเวอร์แซมพลิ่งและอันเดอร์แซมพลิ่งในเชิงการเลือกตอบหลายกลุ่ม ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบประสิทธิภาพระหว่างวิธีโอเวอร์แซมพลิ่งและวิธีอันเดอร์แซมพลิ่ง

Datasets	Label-Based Macro F_1		Label-Based Micro F_1	
	Oversampling	*Undersampling*	*Oversampling*	*Undersampling*
Yeast	**0.409 (-2.9%)**	0.397	**0.599 (-1.7%)**	0.589
Emotion	**0.365 (-3.5%)**	0.352	**0.551 (-1.6%)**	0.542
Average	**0.387 (-3.3%)**	0.374	**0.575 (-1.7%)**	0.565

ตารางที่ 9 อธิบายถึงการเปรียบเทียบความสามารถในการจัดกลุ่มข้อมูลแบบหลายฉลาก โดยจะเปรียบเทียบประสิทธิภาพระหว่างวิธีโอเวอร์แซมพลิ่งและอันเดอร์แซมพลิ่ง ซึ่งผลของการทดลองนี้แสดงให้เห็นว่าวิธีการแก้ปัญหาความไม่สมดุลของข้อมูลด้วยวิธีโอเวอร์แซมพลิ่งนั้นมีประสิทธิภาพสูงกว่าวิธีอันเดอร์แซมพลิ่ง วิธีโอเวอร์แซมพลิ่งมีประสิทธิภาพสูงกว่าวิธีอันเดอร์แซมพลิ่ง 3.3 % ในตัวชี้วัด Macro-F_1 และวิธีโอเวอร์แซมพลิ่งมีประสิทธิภาพสูงกว่าวิธีอันเดอร์แซมพลิ่ง 1.7 % ในตัวชี้วัด Micro-F_1

ตารางที่ 10 ผลการทดลองประสิทธิภาพของวิธีโอเวอร์แซมพลิ่งและอันเดอร์แซมพลิ่งในเชิงการเลือกคำตอบที่ถูกต้อง ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบประสิทธิภาพระหว่างวิธีโอเวอร์แซมพลิ่งและวิธีอันเดอร์แซมพลิ่ง

Datasets	Example-Based Macro F_1		Example-Based Micro F_1	
	Oversampling	*Undersampling*	*Oversampling*	*Undersampling*
Yeast	**0.596 (-3.2%)**	0.577	**0.612 (-2.3%)**	0.598
Emotion	**0.550 (-0.7%)**	0.546	**0.551 (-1.6%)**	0.542
Average	**0.573 (-2.1%)**	0.561	**0.581 (-1.9%)**	0.570

ตารางที่ 10 อธิบายถึงการเปรียบเทียบประสิทธิภาพของการจำแนกประเภทแบบหลายฉลากในด้านของความสามารถเลือกตอบกลุ่มที่อยู่ในผลเฉลยมากที่สุดของวิธีโอเวอร์แซมพลิ่งและอันเดอร์แซมพลิ่งซึ่งผลการทดลองแสดงให้เห็นว่า ความสามารถในการเลือกตอบของวิธีโอเวอร์แซมพลิ่งมีประสิทธิภาพสูงกว่าอันเดอร์แซมพลิ่ง ซึ่งวิธีโอเวอร์แซมพลิ่งมีประสิทธิภาพสูงกว่าวิธีอันเดอร์แซมพลิ่ง 2.1 % ในตัวชี้วัด Macro-F_1 และวิธีโอเวอร์แซมพลิ่งมีประสิทธิภาพสูงกว่าวิธีอันเดอร์แซมพลิ่ง 1.9 % ในตัวชี้วัด Micro-F_1

ตารางที่ 11 ผลการทดลองเปรียบเทียบระยะเวลาประมวลผลของวิธีโอเวอร์แซมพลิ่งและอันเดอร์แซมพลิ่ง ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบระยะเวลาในการประมวลผลระหว่างวิธีโอเวอร์แซมพลิ่งและวิธีอันเดอร์แซมพลิ่ง

Datasets	*Oversampling*	*Undersampling*
Yeast	31.05 (-26.9%)	**22.68**
Emotion	3.73 (-11.1)	**3.32**
Average	18.10 (-26.2%)	**13.35**

ตารางที่ 11 อธิบายถึงการเปรียบเทียบระยะเวลาที่ใช้ในการประมวลผลของวิธีโอเวอร์แซมพลิ่งและวิธีอันเดอร์แซมพลิ่ง ซึ่งผลการทดลองในตารางนี้แสดงให้เห็นว่า วิธีโอเวอร์แซมพลิ่งใช้เวลาในการประมวลผลมากกว่าวิธีอันเดอร์แซมพลิ่งโดยเฉลี่ยถึง 26.2 % ในข้อมูลขนาดกลาง 1 ชุดข้อมูล และข้อมูลขนาดเล็ก 1 ชุดข้อมูล

จากผลการทดลองในตารางที่ 8-11 แสดงให้เห็นว่า วิธีโอเวอร์แซมพลิ่งนั้นมีประสิทธิภาพสูงกว่าวิธีอันเดอร์แซมพลิ่งเล็กน้อย อย่างไรก็ตาม วิธีโอเวอร์แซมพลิ่งนั้นใช้เวลาในการประมวลผลมากกว่าวิธีอันเดอร์แซมพลิ่งเป็นอย่างมาก ทำให้ผู้วิจัยเลือกที่จะใช้งานวิธีอันเดอร์แซมพลิ่ง เพราะวิธีนี้มีประสิทธิภาพใกล้เคียงกับวิธีโอเวอร์แซมพลิ่ง อีกทั้งใช้เวลาในการประมวลผลน้อยกว่าวิธีโอเวอร์แซมพลิ่งเป็นอย่างมาก

5.3 การทดลองเพื่อเปรียบเทียบวิธีการที่ผู้วิจัยได้นำเสนอกับวิธีพื้นฐาน

ในการทดลองนี้ ผู้วิจัยได้เลือกใช้งานชุดข้อมูลทั้งหมด 6 ชุดข้อมูลที่มีขนาดของกลุ่มข้อมูลแตกต่างกัน ประกอบด้วย ชุดข้อมูลที่มีกลุ่มข้อมูลขนาดเล็ก 1 ชุดข้อมูล คือ Emotion ชุดข้อมูลที่มีจำนวนกลุ่มข้อมูลขนาดกลาง 2 ชุดข้อมูล ประกอบด้วย Yeast และ Birds ชุดข้อมูลที่มีกลุ่มข้อมูล

ขนาดใหญ่ 3 ชุดข้อมูล ประกอบด้วย Tmc2007 IMDB และ Rcv1v2 (subset) โดยการทดลองนี้ผู้วิจัยจะทำการเปรียบเทียบประสิทธิภาพระหว่างวิธีหนึ่งต่อทั้งหมดที่มีการประยุกต์ใช้วิธีอันเดอร์แซมพลิ่ง วิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล ซึ่งวิธีนี้จะทำการปรับค่า gamma โดยค่าที่ใช้ในการปรับของแต่ละชุดข้อมูลนั้นจะแตกต่างกัน โดยจะเลือกค่าพื้นฐานเป็นค่าแรก และทำการปรับเพิ่มจำนวนขึ้นเรื่อย ๆ เพื่อหาค่า gamma ที่มีความเหมาะสมมากที่สุด โดยค่าพื้นฐานของ RBF นั้นจะสามารถคำนวณได้จาก 1ต่อจำนวนลักษณะข้อมูล ซึ่งค่าที่จะเพิ่มขึ้นมี 4 ค่า ดังนี้ ค่าพื้นฐาน + 0.01 ค่าพื้นฐาน + 0.1 ค่าพื้นฐาน + 0.5 และค่าพื้นฐาน + 1 โดยสามารถอ้างอิงค่า gamma ที่ใช้ในแต่ละชุดข้อมูลได้ในตารางที่ 15 โดยที่ค่า gamma ซึ่งถูกเลือกใช้ในวิธีนี้นั้นสามารถแสดงได้ในตารางที่ 16 และทำการปรับค่าขีดแบ่งเพื่อใช้ในการเลือกตอบแบบหลายฉลากโดยสามารถอ้างอิงค่าขีดแบ่งที่ใช้ในการเลือกตอบได้ในตารางที่ 17 และวิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้การคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้องที่ผู้วิจัยได้นำเสนอ ซึ่งจะทำการวัดประสิทธิภาพของแต่ละวิธี โดยใช้ตัวชี้วัดดังนี้ Accuracy Label-based และ Example-based ซึ่งผลการทดลองจะแสดงให้เห็นในตารางที่ 12-14 โดยการทดลองนี้ได้มีการใช้งานวิธี 5 fold cross-validation

ตารางที่ 12 ผลการทดลองการจัดกลุ่มแบบหลายประเภทในเชิงความแม่นยำ ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบประสิทธิภาพของวิธีที่ผู้วิจัยได้นำเสนอกับวิธีมาตราฐาน

Datasets	*US-OVA*	*OVO*	*MRUS-OVO*
Yeast	0.518 (+31.9%)	0.703 (+7.6%)	**0.761**
Emotion	0.644 (+7.2%)	0.652 (+6.0%)	**0.694**
IMDB	0.408 (+54.6%)	0.887 (+1.4%)	**0.900**
Tmc2007	0.898 (+6.2%)	0.925 (+3.5%)	**0.957**
Blrds	0.702 (+23.1%)	0.826 (+9.5%)	**0.913**
Rcv1v2	0.653 (+30.7%)	0.751 (+20.3%)	**0.943**
Average	0.637 (+26.0%)	0.790 (+8.2%)	**0.861**

จากตารางที่ 12 กล่าวถึงการเปรียบเทียบผลการทดลองในเชิงความแม่นยำในการจำแนกระหว่างวิธีหนึ่งต่อทั้งหมดที่มีการประยุกต์ใช้งานวิธีอันเดอร์แซมพลิ่ง วิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้จัดลำดับความสำคัญของกลุ่มข้อมูล และวิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้การคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้องซึ่งแสดงให้เห็นว่าการวิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้วิธีการเลือกตอบโดยวิธีการคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้องมีประสิทธิภาพในการจำแนกสูงกว่าวิธีพื้นฐานทั้งสองวิธี แสดงให้เห็นว่าการประยุกต์ใช้งานวิธีการจำแนกแบบหนึ่งต่อหนึ่งที่ผู้วิจัยได้นำเสนอนั้นสามารถเพิ่มประสิทธิภาพของการจำแนกในเชิงความแม่นยำเป็นอย่างมาพลิ่ง ซึ่งวิธีที่ผู้วิจัยได้นำเสนอนั้นมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อทั้งหมดที่มีการประยุกต์ใช้วิธีอันเดอร์แซมพลิ่ง 26.0 % และมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อหนึ่งประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล 8.2 %

ตารางที่ 13 ผลการทดลองวิธีการจัดกลุ่มแบบหลายประเภทในเชิงการเลือกตอบหลายกลุ่ม ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบประสิทธิภาพของวิธีที่ผู้วิจัยได้นำเสนอกับวิธีมาตราฐาน

Datasets	Label-Based Macro F_1			Label-Based Micro F_1		
	US-OVA	*OVO*	*MRUS-OVO*	*US-OVA*	*OVO*	*MRUS-OVO*
Yeast	0.357 (+10.1%)	0.359 (+9.5%)	**0.397**	0.420 (+29.5%)	0.480 (+19.5%)	**0.596**
Emotion	0.207 (+49.3%)	0.352 (+13.9%)	**0.409**	0.519 (+7.7%)	**0.582 (-3.6%)**	0.562
IMDB	0.117 (+22.0%)	**0.158 (-5.3%)**	0.150	0.196 (+60.7%)	0.431 (+13.6%)	**0.499**
Tmc2007	0.207 (+61.5%)	0.462 (+13.9%)	**0.537**	0.211 (+73.5%)	0.723 (+9.4%)	**0.798**
Birds	0.143 (+50.3%)	0.171 (+40.6%)	**0.288**	0.290 (+56.2%)	0.515 (+22.2%)	**0.662**
Rcv1v2	0.129 (+32.8%)	0.188 (+2.1%)	**0.192**	0.105 (+62.2%)	0.228 (+17.9%)	**0.278**
Average	0.193 (+41.2%)	0.281 (+14.3%)	**0.328**	0.290 (+48.5%)	0.563 (+0.2%)	**0.564**

ตารางที่ 13 อธิบายถึงการเปรียบเทียบความสามารถในการจัดกลุ่มข้อมูลแบบหลายประเภทโดยจะเปรียบเทียบกับวิธีหนึ่งต่อทั้งหมดที่มีการประยุกต์ใช้งานวิธีอันเดอร์แซมพลิ่ง เพื่อลดปัญหาความไม่สมดุลของข้อมูล วิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้จัดลำดับความสำคัญของกลุ่มข้อมูล

และวิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้การคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้องซึ่งผลการทดลองแสดงให้เห็นว่าวิธีการจำแนกแบบหนึ่งต่อหนึ่งที่ประยุกต์ใช้วิธีการเลือกตอบโดยวิธีการคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้องมีประสิทธิภาพสูงกว่าวิธีการพื้นฐานในด้านประสิทธิภาพการจำแนกประเภทแบบหลายฉลาก ซึ่งวิธีที่ผู้วิจัยได้นำเสนอนั้นมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อทั้งหมดที่มีการประยุกต์ใช้วิธีอันเดอร์แซมพลิ่ง 41.2 % และมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อหนึ่งประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล 14.3 % ในตัวชี้วัด Macro-F_1 และมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อทั้งหมดที่มีการประยุกต์ใช้วิธีอันเดอร์แซมพลิ่ง 48.5 % และมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อหนึ่งประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล 0.2 % ในตัวชี้วัด Micro-F_1

ตารางที่ 14 ผลการทดลองการจัดกลุ่มแบบหลายประเภทในเชิงการเลือกคำตอบที่ถูกต้อง ซึ่งค่าในวงเล็บคืออัตราการเปรียบเทียบประสิทธิภาพของวิธีที่ผู้วิจัยได้นำเสนอกับวิธีมาตราฐาน

Datasets	Example-Based Macro F_1			Example-Based Micro F_1		
	US-OVA	*OVO*	*MRUS-OVO*	*US-OVA*	*OVO*	*MRUS-OVO*
Yeast	0.438 (+25.9%)	0.483 (+18.3%)	**0.591**	0.446 (+26.2%)	0.480 (+20.6%)	**0.605**
Emotion	0.525 (+14.6%)	0.586 (+4.7%)	**0.615**	0.519 (+25.1%)	0.582 (+15.8%)	**0.692**
IMDB	0.255 (+49.2%)	0.439 (+12.5%)	**0.502**	0.247 (+50.5%)	0.431 (+13.6%)	**0.499**
Tmc2007	0.305 (+62.1%)	0.717 (+10.8%)	**0.804**	0.309 (+61.3%)	0.723 (+9.4%)	**0.798**
Birds	0.290 (+57.4%)	0.522 (+23.2%)	**0.680**	0.294 (+55.5%)	0.515 (+22.2%)	**0.662**
Rcv1v2	0.104 (+75.1%)	0.227 (+45.7%)	**0.418**	0.105 (+75.0%)	0.228 (+46.1%)	**0.423**
Average	0.319 (+46.9)	0.495 (+17.6%)	**0.601**	0.320 (+47.7%)	0.493 (+19.6%)	**0.613**

ตาราง 14 อธิบายถึงการเปรียบเทียบประสิทธิภาพของการจำแนกประเภทแบบหลายฉลากในด้านของความสามารถเลือกตอบกลุ่มที่อยู่ในผลเฉลยมากที่สุด โดยวิธีการที่ใช้ในการเปรียบเทียบประกอบด้วย วิธีหนึ่งต่อทั้งหมดที่มีการประยุกต์ใช้งานวิธีอันเดอร์แซมพลิ่งเพื่อลดปัญหาความไม่

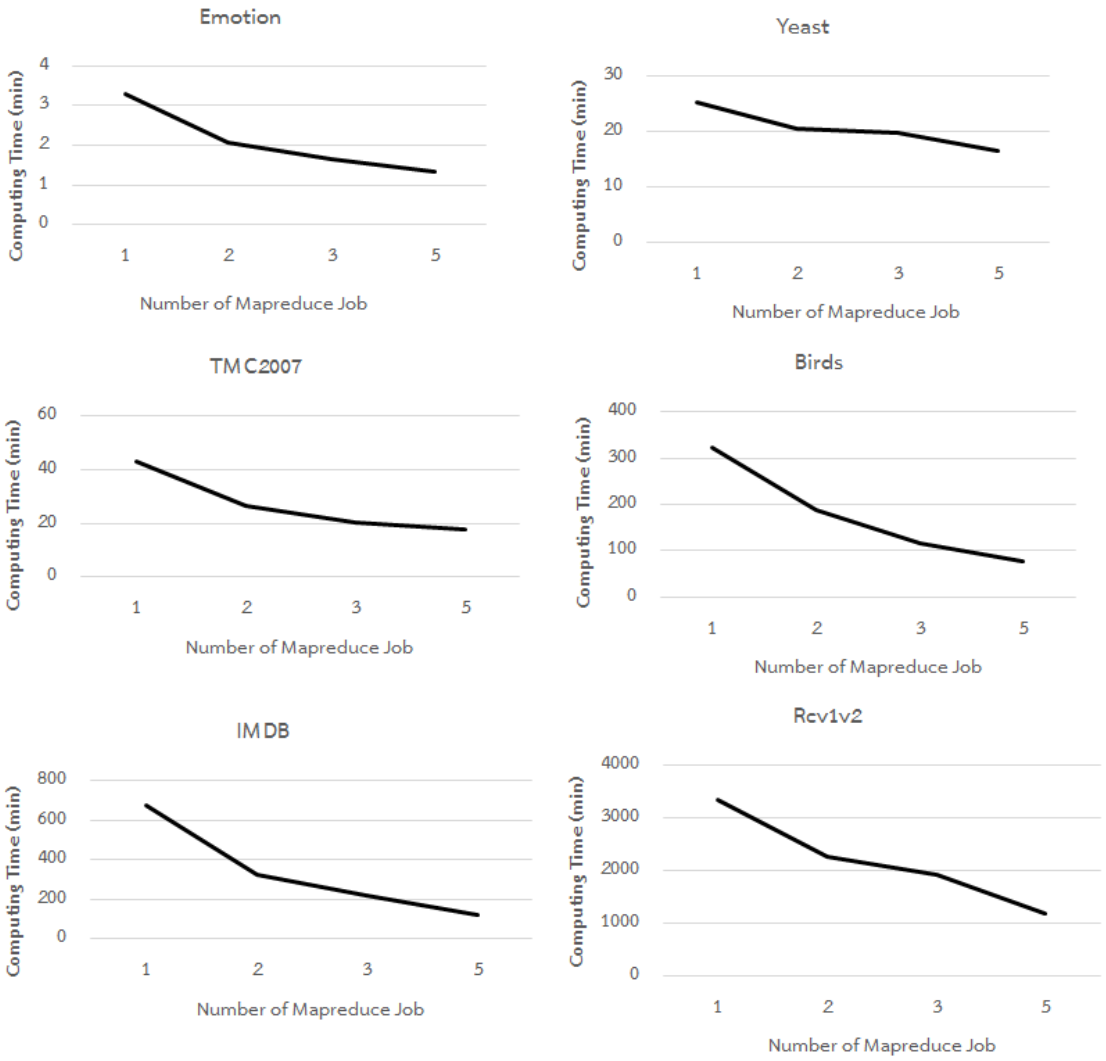
สมดุลของข้อมูล วิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้จัดลำดับความสำคัญของกลุ่มข้อมูล และวิธีหนึ่งต่อหนึ่งที่ประยุกต์ใช้การคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้อง โดยผลการทดลองแสดงให้เห็นถึงความสามารถในการเลือกตอบของวิธีจำแนกหนึ่งต่อหนึ่งที่ประยุกต์ใช้วิธีการเลือกตอบโดยวิธีการคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้องซึ่งมีประสิทธิภาพสูงกว่าวิธีพื้นฐานทั้งหมด อีกทั้งหากนำวิธีการแก้ไขปัญหาความไม่สมดุลของข้อมูลมาประยุกต์ใช้งานจะแสดงให้เห็นถึงประสิทธิภาพของระบบที่เพิ่มขึ้นอย่างเห็นได้ชัด ซึ่งวิธีที่ผู้วิจัยได้นำเสนอนั้นมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อทั้งหมดที่มีการประยุกต์ใช้วิธีอันเดอร์แซมพลิ่ง 46.9 % และมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อหนึ่งประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล 17.6 % ในตัวชี้วัด Macro-F_1 และมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อทั้งหมดที่มีการประยุกต์ใช้วิธีอันเดอร์แซมพลิ่ง 47.7 % และมีประสิทธิภาพสูงกว่าวิธีหนึ่งต่อหนึ่งประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล 19.6 % ในตัวชี้วัด Micro-F_1

จากผลการทดลองทั้งหมดจึงสามารถสรุปได้ว่า วิธีการจำแนกแบบหนึ่งต่อหนึ่งที่ประยุกต์ใช้การคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้องนั้นมีความเหมาะสมกับการจำแนกประเภทข้อมูลแบบหลายฉลากเป็นอย่างมาก อีกทั้งจากผลการทดลองจะแสดงให้เห็นว่าวิธีที่ผู้วิจัยที่ได้นำเสนอนั้นมีประสิทธิภาพสูงกว่าวิธีพื้นฐานอย่างเห็นได้ชัด

5.4 การทดลองเปรียบเทียบเวลาในการประมวลผล

ในการทดลอง จะมีการวัดประสิทธิภาพระยะเวลาในการประมวลผลของระบบประมวลผลแบบกระจายสปาร์ค โดยจะเปรียบเทียบผลการทดลองประกอบด้วยงาน 1 2 3 และ 5 แมพ/รีดิวซ์ โดยการทดลองนี้จะใช้ชุดข้อมูลทั้งหมด 6 ชุดข้อมูล ประกอบด้วย Emotion Yeast Tmc2007 Birds IMDB และ Rcv1v2 (subset) ซึ่งการทดลองนี้ได้ทำการทดลองโดยใช้งานเครื่องในการประมวลผลทั้งหมด 6 เครื่อง ประกอบด้วย เครื่องแม่ข่าย 1 เครื่อง และเครื่องลูกข่าย 5 เครื่อง โดยประสิทธิภาพของแต่ละเครื่อง มีดังนี้ CPU 8 Core 2.50 GHz Memory 12 GB โดยระยะเวลาในการประมวลผลของแต่ละชุดข้อมูลนั้นสามารถแสดงได้ดังรูปที่ 11

จากผลการทดลองในรูปที่ 11 แสดงให้เห็นถึงความต่างของระยะเวลาที่ใช้ในการประมวลผลโดยใช้งานระบบประมวลผลแบบกระจายสปาร์ค ซึ่งสามารถแสดงให้เห็นว่าการนำระบบประมวลผลแบบกระจายมาประยุกใช้งานกับวิธีจำแนกประเภทแบบหลายฉลากแบบหนึ่งต่อหนึ่งด้วยการแบ่งงานออกเป็นหลาย ๆ ส่วนนั้นสามารถลดระยะเวลาประมวลผลของแต่ละชุดข้อมูลได้เป็นอย่างมาก และยังสามารถรักษาประสิทธิภาพของการจำแนกประเภทให้คงเดิมได้อีกด้วย



รูปที่ 11 การเปรียบเทียบระยะเวลาที่ใช้ในการประมวลผลของแต่ละงานแมพ/รีดิวซ์

บทที่ 6
สรุปผลและข้อเสนอแนะ

6.1 สรุปผลการทดลอง

งานวิจัยนี้ผู้วิจัยมีวัตถุประสงค์ เพื่อเพิ่มประสิทธิภาพของการจำแนกแบบหลายฉลากโดยใช้วิธีการจำแนกแบบหนึ่งต่อหนึ่ง โดยประยุกต์ใช้งานซัพพอร์ตเวคเตอร์แมชชีนและประยุกต์ใช้งานระบบประมวลผลแบบกระจายสปาร์ค เพื่อทำการลดระยะเวลาใช้การประมวลของระบบทั้งหมด ซึ่งระหว่างทำการวิจัย ผู้วิจัยได้พบปัญหาในการใช้งานวิธีการจำแนกแบบหนึ่งต่อหนึ่งร่วมกับวิธีซัพพอร์ตเวคเตอร์แมชชีน โดยที่ปัญหานี้นั้นจะเกิดขึ้นเมื่อทำการทดสอบโครงสร้างการจำแนกแบบหนึ่งต่อหนึ่งด้วยชุดข้อมูลทดสอบ ซึ่งจะพบว่ามีมีกลุ่มข้อมูลที่ไม่เกี่ยวข้องปรากฏขึ้นทำให้เกิดการจำแนกผิดพลาดได้ ผู้วิจัยได้พัฒนาวิธีการจำแนกแบบหนึ่งต่อหนึ่งที่ประยุกต์ใช้การคัดกรองกลุ่มข้อมูลที่ไม่เกี่ยวข้องเพื่อแก้ไขปัญหาที่พบข้อมูลที่ไม่เกี่ยวข้อง และได้ทำการประยุกต์ใช้วิธีอันเดอร์แซมพลิ่งเพื่อใช้ในการแก้ไขปัญหาความไม่สมดุลของข้อมูลที่เกิดขึ้นในบางตัวจำแนก อย่างไรก็ตาม การประยุกต์ใช้งานระบบประมวลผลแบบกระจายสปาร์คมีข้อจำกัดในด้านของลักษณะการทำงานซึ่งเป็นการทำงานแบบเริ่มทำงานก่อนจะเสร็จก่อน จึงทำให้ไม่สามารถใช้งานประสิทธิภาพของระบบได้เต็มที่ ผู้วิจัยจึงได้เสนอวิธีการประยุกต์ใช้งานระบบประมวลผลแบบกระจาย โดยการแบ่งงานแมพ/รีดิวซ์ออกเป็นหลาย ๆ ส่วนย่อย เพื่อให้ระบบประมวลผลแบบกระจายสปาร์คนั้นสามารถแสดงประสิทธิภาพและใช้งานทรัพยากรในการประมวลผลได้อย่างเต็มที่ โดยงานวิจัยนี้ผู้วิจัยได้แบ่งการทดลองออกเป็น 4 ส่วน ประกอบด้วย

1) การทดลองเลือกเคอร์เนลที่ดีที่สุดของซัพพอร์ตเวคเตอร์แมชชีนการทดลองนี้มี เพื่อใช้ในการทดสอบหาเคอร์เนลที่มีประสิทธิภาพในการจำแนกสูงที่สุด โดยที่ผู้วิจัยได้เลือกใช้ชุดข้อมูลขนาดเล็ก 1 ชุดข้อมูล และข้อมูลขนาดกลาง 1 ชุดข้อมูล เนื่องจากหากทำการทดสอบการเลือกเคอร์เนลที่ดีที่สุดโดยใช้ชุดข้อมูลทั้งหมดที่ผู้วิจัยเลือกใช้จำเป็นต้องใช้เวลาในการประมวลผลอย่างมหาศาล ผู้วิจัยจึงได้เลือกใช้ข้อมูลขนาดเล็ก 1 ชุดข้อมูล และข้อมูลขนาดกลาง 1 ชุดข้อมูล เพื่อลดระยะเวลาที่ใช้ในการทำการทดลอง ซึ่งผลการทดลองแสดงให้เห็นว่าเคอร์เนล RBF นั้นมีประสิทธิภาพสูงที่สุดผู้วิจัยจึงเลือกใช้เคอร์เนล RBF

2) การทดลองเพื่อเปรียบเทียบประสิทธิภาพในการจำแนก และระยะเวลาในการประมวลผลระหว่างวิธีโอเวอร์ และวิธีอันเดอร์แซมพลิ่ง โดยที่ผู้วิจัยได้เลือกใช้ชุดข้อมูลขนาดเล็ก 1 ชุด

ข้อมูล และข้อมูลขนาดกลาง 1 ชุดข้อมูล เนื่องจากหากทำการทดสอบการเลือกวิธีการแก้ปัญหาความไม่สมดุลที่มีความเหมาะสมมากที่สุด โดยใช้ชุดข้อมูลทั้งหมดที่ผู้วิจัยเลือกใช้จำเป็นต้องใช้เวลาในการประมวลผลอย่างมหาศาล โดยเฉพาะอย่างยิ่งเมื่อทำการทดลองด้วยวิธีโอเวอร์แซมพลิ่ง ผู้วิจัยจึงได้เลือกใช้ข้อมูลขนาดเล็ก 1 ชุดข้อมูล และข้อมูลขนาดกลาง 1 ชุดข้อมูล เพื่อลดระยะเวลาทำการทดลองเลือกวิธีการแก้ปัญหาความไม่สมดุลที่มีความเหมาะสมกับงานวิจัยชิ้นนี้ ซึ่งผลการทดลองในส่วนนี้แสดงให้เห็นว่าวิธีโอเวอร์แซมพลิ่งและวิธีอันเดอร์แซมพลิ่งนั้นมีประสิทธิภาพใกล้เคียงกัน อย่างไรก็ตาม วิธีโอเวอร์แซมพลิ่งนั้นมีระยะเวลาในการประมวลผลมากกว่าอันเดอร์แซมพลิ่งเป็นอย่างมาก ผู้วิจัยจึงเลือกใช้วิธีอันเดอร์แซมพลิ่ง เนื่องจากวิธีนี้ใช้ระยะเวลาในการประมวลผลน้อยกว่าเป็นอย่างมากอีกทั้งมีประสิทธิภาพใกล้เคียงกับโอเวอร์แซมพลิ่ง

3) การทดลองเปรียบเทียบประสิทธิภาพของวิธีการที่ผู้วิจัยได้นำเสนอกับวิธีการพื้นฐาน การทดลองนี้ผู้วิจัยทำการทดลองโดยใช้ ชุดข้อมูลทั้งหมด 6 ชุด โดยทำการเปรียบเทียบกับวิธีการจำแนกแบบหนึ่งต่อทั้งหมดซึ่งได้ทำการประยุกต์ใช้วิธีอันเดอร์แซมพลิ่งเพื่อแก้ปัญหาความไม่สมดุลของข้อมูล และวิธีจำแนกแบบหนึ่งต่อหนึ่งซึ่งประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล ซึ่งในการทดลองนี้แสดงให้เห็นว่าวิธีการจำแนกแบบหนึ่งต่อหนึ่งนั้นมีความเหมาะสมในการจำแนกข้อมุลแบบหลายฉลากเป็นอย่างมาก ซึ่งจากผลการทดลองจะแสดงให้เห็นว่า วิธีที่ผู้วิจัยได้นำเสนอนั้นมีประสิทธิภาพสูงกว่าวิธีพื้นฐานอย่างเห็นได้ชัด

4) การทดลองเปรียบเทียบระยะเวลาในการประมวลผลโดยใช้ระบบประมวลผลแบบกระจาย การทดลองนี้ผู้วิจัยได้เลือกใช้ชุดข้อมูลในการทดลองทั้งหมด 6 ชุดข้อมูล โดยทำการเปรียบเทียบระยะเวลาที่ใช้ในการประมวลผลของระบบประมวลผลแบบกระจายโดยไม่มีการแบ่งส่วนการทำงานกับวิธีการประยุกต์ใช้ระบบประมวลผลแบบกระจายที่ผู้วิจัยนำเสนอ ซึ่งเป็นการแบ่งการทำงานออกเป็นส่วนย่อย ๆ หลายส่วน โดยผลการทดลองแสดงให้เห็นว่าวิธีการแบ่งงานออกเป็นหลายส่วนที่ผู้วิจัยเสนอนั้นสามารถลดระยะเวลาในการประมวลผลลงอย่างเห็นได้ชัดอีกทั้งยังสามารถคงประสิทธิภาพในการจำแนกให้คงเดิมได้อีกด้วย

จากผลการทดลองทั้งหมดแสดงให้เห็นว่างานวิจัยชิ้นนี้สามารถเพิ่มประสิทธิภาพในการจำแนกข้อมูลแบบหลายฉลากได้และสามารถลดระยะเวลาในการประมวลผลด้วยการประยุกต์ใช้งานระบบประมวลผลแบบกระจายสปาร์คได้อีกด้วย

6.2 ข้อจำกัดของงานวิจัย

ข้อจำกัดของงานวิจัยนี้ประกอบด้วย 2 ส่วน คือ

1) ทรัพยากรในการประมวลผลที่มีอยู่อย่างจำกัด งานวิจัยนี้ผู้วิจัยมีทรัพยากรที่ใช้ในการประมวลอยู่จำกัด ทำให้ไม่สามารถนำวิธีการที่ได้นำเสนอมาทดสอบกับข้อมูลที่มีขนาดใหญ่ได้

2) จำนวนตัวจำแนก โดยวิธีการจำแนกแบบหนึ่งต่อหนึ่งทำการจำคู่กลุ่มข้อมูลที่เป็นไปได้เพื่อใช้ในการสร้างโครงสร้างตัวจำแนก ซึ่งจำเป็นต้องสร้างตัวจำแนกจำนวนมาก ซึ่งถ้าหากสามารถหาความสัมพันธ์ของกลุ่มข้อมูลแต่ละกลุ่มได้ก็จะสามารถลดจำนวนของตัวจำแนกที่ไม่เกี่ยวข้องกันได้เป็นจำนวนมาก

6.3 งานวิจัยในอนาคต

งานวิจัยในอนาคตนั้นสามารถแบ่งได้เป็น 2 ส่วน คือ

1) พัฒนาวิธีการจำแนกแบบหนึ่งต่อหนึ่งที่คำนึงถึงความสัมพันธ์ของโครงสร้างลำดับชั้นของข้อมูล ซึ่งสามารถใช้ในการลดจำนวนของตัวจำแนกได้

2) ประยุกต์ใช้วิธีการอื่น ๆ เพื่อหาความสัมพันธ์ของกลุ่มข้อมูล เทำการลดจำนวนตัวจำแนก ซึ่งสามารถลดระยะเวลาในการประมวลผลได้อีกด้วย

6.4 ผลงานที่ได้รับการตีพิมพ์และรออนุมัติตีพิมพ์

1) หัวเรื่องงานวิจัยที่ได้รับการตีพิมพ์ชื่อ “Enhancing Accuracy of Multi-Label Classification by Applying One-vs-One Support Vector Machine” ได้รับการตีพิมพ์ในงานประชุมวิชาการระดับนานาชาติ The 13th International Joint Conference on Computer Science and Software Engineering จัดขึ้น ณ จังหวัด ขอนแก่น ประเทศไทย วันที่ 13 กรกฎาคม 2559 ถึง วันที่ 15 กรกฎาคม 2559

2) หัวเรื่องงานวิจัยที่รออนุมัติการตีพิมพ์ชื่อ “Applying One-Versus-One SVMs to Classify Multi-Label Data with Large Labels Using Spark” รออนุมัติการตีพิมพ์ในงานประชุมวิชาการระดับนานาชาติ The 9th International Conference on Knowledge and Smart Technology จัดขึ้น ณ จังหวัด ชลบุรี ประเทศไทย วันที่ 1 กุมพาพันธ์ 2560 ถึง วันที่ 4 กุมภาพันธ์ 2560

รายการอ้างอิง

1. G. Tsoumakas, and I. Katakis, *Multi-label classification: An overview.* Int. J. Data Warhouse, 2007. 3: p. 1-13.
2. Sorower, M.S., *A literature survey on algorithms for multi-label learning.* 2010.
3. J. Xu, *An extended one-versus-rest support vector machine for multi-label classification.* Neurocomputing, 2011. 74: p. 3114-3124.
4. P. Vateekul, S. Dendamrongvit and M. Kubat, *Improving SVM performance in multi-label domains: Threshold adjustment.* Int. J. Artif. Intell. T., 2013. 22.
5. T. Ananpiriyakul, P. Poomsirivilai, and P. Vateeku, *Label correction strategy on hierarchical multi-label classification.* Lect. Notes. Comput. Sc., 2014. 8556: p. 213-227.
6. G. Anthony, H. Gregg and M. Tshilidzi, *Image classification using SVMs: One-against-one vs one-against-all.* The 28th Asian Conference on Remote Sensing, 2007.
7. R. K. Eichelberger, and V. S. Sheng, *Does one-against-all or one-against-one improve the performance of multiclass classifications.* The 27th AAAI Conference on Artificial Intelligence, 2013.
8. S. Daengduang, P. Vateekul,"Enhancing Accuracy of MultiLabel Classification by Applying One-vs-One Support Vector Machine," 13th International Joint Conference on Computer Science and Software Engineering, 2016.
9. S. Merghani, A. Elrahman, and A. Abraham, *A review of class imbalance problem.* JNIC, 2013. 1: p. 332-340.
10. "Apache Spark," http://spark.apache.org/.
11. "Apache Hadoop," http://hadoop.apache.org/.
12. C. Y. Lin, C. H. Tsai, , C. P. Lee, C. J. Lin,"Large-scale logistic regression and linear support vector machines using Spark," 2014 IEEE International Conference on Big Data., pp. 519-528, 2014.

13. J. Maillo, I. Triguero, F. Herrera,“A Map-reduce-Based kNearest Neighbor Approach for Big Data Classification,” Trustcom/BigDataSE/ISPA, 2015 IEEE, vol. 2, pp.167-172, 2015.
14. N. K. Alham, M. Li, Y. Liu, S. Hammoud, “A Map-reducebased distributed SVM algorithm for automatic image annotation,” Comput. Math. Appl., vol.62, pp.2801-2811, 2011.
15. N. Phachongkitphiphat, and P. Vateekul, *An improvement of flat approach on hierarchical text classification using top-level pruning classifiers.* The 11th International Joint Conference on Computer Science and Software Engineering, 2014. 86-90.
16. X. Y. Liu, J. Wu, and Z. H. Zhou, *Exploratory undersampling for class-imbalance learning.* IEEE Trans. Syst. Man. Cybern. B. Cybern., 2009. 39: p. 539-550.
17. T. Choeikiwong, and P. Vateekul, *Software defect prediction in imbalanced Datasetss using unbiased support vector machine.* Lecture Notes in Electrical Engineering, 2015. 339: p. 923-931.
18. B. Zhang, X. Xu, and J. Su, *An ensemble method for multi-class and multi-label text categorization.* The international conference on intelligent systems and knowledge engineering, 2007.
19. S. Dendamrongvit and M. Kubat, *Undersampling Approach for Imbalanced Training Sets and Induction from Multi-label Text-Categorization Domains.* New Frontiers in Applied Data Mining. Springer Berlin Heidelberg, 2009.
20. “Mastering Apache Spark 2.0” https://jaceklaskowski.gitbooks.io
21. F. Ö. Çatak, M. E. Balaban, “A Map-reduce-based distributed SVM algorithm for binary classification,” Turk. J. Elec. Eng. & Comp. Sci., vol. 24, pp.863—873, 2016.

22. Fürnkranz, J., Hüllermeier, E., Mencía, E.L. and Brinker, K., "Multilabel classification via calibrated label ranking," *Machine learning*, vol. *73*(2), pp.133-153, 2008.

23. "Mulan," http://mulan.sourceforge.net.

24. "Spark-sklearn," http://pythonhosted.org/spark-sklearn/.

ภาคผนวก

ภาคผนวก ก.

บทความทางวิชาการณ์เรื่อง “Enhancing Accuracy of Multi-Label Classification by Applying One-vs-One Support Vector Machine” โดย สุทธิพงษ์ แดงด้วง และ พีรพล เวทีกูล ในงานประชุมวิชาการ The 13th International Joint Conference on Computer Science and Software Engineering จัดขึ้น ณ จังหวัด ขอนแก่น ประเทศไทย วันที่ 13 กรกฎาคม 2559 ถึง วันที่ 15 กรกฎาคม 2559

2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)

Enhancing Accuracy of Multi-Label Classification by Applying One-vs-One Support Vector Machine

Suthipong Daengduang, Peerapon Vateekul

Department of Computer Engineering
Faculty of Engineering Chulalongkorn University
Bangkok, Thailand
Suthipong.D@student.chula.ac.th, Peerapon.V@Chula.ac.th

Abstract— **Multi-label classification is a supervised learning, where one example can belong to several classes. In the case of Support Vector Machine (SVM), One-versus-All (OVA) is the most common approach to tackle this problem. However, the accuracy is very limited due to extremely imbalanced training set. It is interesting that there have only very few works that applied One-versus-One (OVO) in the multi-label domain even though it has been shown to provide better accuracy than OVA in the multiclass domain. In this paper, we propose a multi-label classification framework that employs OVO incorporating with the undersampling technique to alleviate the imbalanced issue. In the experiment, there are five standard benchmarks. The results show that our proposed algorithm outperforms OVA and traditional OVO in all data sets in terms of accuracy and F_1.**

Keywords- Multi-Label; One-vs-One; Classification; Support Vector Machine;

I. INTRODUCTION

Multi-label classification is a supervised learning task where a single example can belong to many classes [1-4]. Recently, it has increasingly been required in wide range of application, e.g. text categorization, semantic image labeling, bioinformatics, music categorization, and etc. Existing methods for the multi-label classification are categorized into two main methods: algorithm adaptation and problem transformation [5]. The algorithm adaption aims to invent a new approach specifically for multi-label classification. The problem transformation focuses on transforming a multi-label classification task to a set of binary classification tasks. This approach is very popular since it can employ any classification techniques that are suitable for any domains resulting in high prediction accuracy.

In this problem transformation approach, there are two main strategies: One-versus-One (OVO) and One-versus-All (OVA). First, OVA induces a set of binary classifiers equal to the number of classes. For *i*-th class, all examples in this class are labeled as positives, while the remaining examples are labeled as negatives [6]. For example, assume there are ten classes, *C1-C10*, and each class contains equal number of examples. To construct classifier *C1*, OVA selects *C1's* examples to be positives and assigns the remaining examples (*C2-C10*) to be negatives. Although, OVA is popular due to it is easy to implement; however, this approach usually encounters "an imbalanced issue," where the majority class outnumbers the minority class. In this example, the number of positive examples is only 10%, while the number of negative examples is 90%. This circumstance always leads to low prediction accuracy. Second, OVO induces a set of all class-pair binary classifiers [7]. For example, assume there are three classes, *C1-C3*. OVO aims to create all pairs of classifiers resulting in 3 classifiers: *C1vsC2, C1vsC3*, and *C2vsC3*. The prediction result is the class with the highest vote by all classifiers. Although, OVO is very popular and has been proved to outperform OVA in the multiclass domain [6, 7], it has been rarely used in the multi-label domain since OVO can only predict one output class.

For the imbalanced issue, there are two common solutions: undersampling and oversampling [8]. Assume the majority class is positive, and the minority class is negative. First, the undersampling strategy balances data by randomly removing examples from minority class. For example, there are 50 positives and 200 negatives. This approach selects 50 positives and randomly selects 50 negatives from all negatives. Second, the oversampling strategy balances data by copying existing examples or adding more examples of the minority class. For example, there are 50 positives and 200 negatives. This approach selects 50 positives and randomly adds 150 positives. The undersampling is a preferred solution to handle with imbalance issue [9].

In this paper, we propose a novel multi-label framework that applies OVO instead of OVA to improve prediction accuracy, while adapts the undersampling technique to alleviate the imbalance bias. Our framework is called "US-OVO". Support Vector Machine (SVM) is our baseline classifier due to its success reported in prior works [3]. To be more specific, the strategy called "negative voting" is presented in order to employ OVO in the multi-label domain, where the traditional voting in OVO cannot be used in this domain and proved in Section 6. The experiment is conducted on five standard data sets. The results show that our approach outperforms the undersampling OVA on all data sets in terms of F_1.

The rest of this paper is organized as follows. Section 2 reviews previous works that relate to our method. Section 3 describes our performance evaluation. Section 4 describes our proposed method. Section 5 shows experimental results. And, Section 6 is conclusion of this paper with possible future work.

978-1-5090-2033-1/16/$31.00 ©2016 IEEE

2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)

II. Related Works

A. Support Vector Machine

SVM which is widely used to be classifier for multi-label classification, is based on the concept of hyperplane to define decision boundaries. The hyperplane is one that separates between a set of objects having different class memberships.

The purpose of SVM is to induce a hyperplane function in (1), where $\vec{w}$ is a weight vector and b is a bias [4, 5, 10].

$$h(\vec{w}, b) = \vec{w} \times (x + b) \quad (1)$$

In (2) shows the optimization function to construct SVM hyperplane, where is a penalty parameter of misclassifications.

$$Minimize_{w,b,\xi} \ \frac{1}{2} w^T w + C \sum_{i=1}^{|D|} \xi_i$$

$$subject\ to\ y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \quad , \xi_i \geq 0 \quad (2)$$

In a non-linear separable problem, SVM handles this by using a kernel function non-linear to map the data into a higher space, where a linear hyperplane cannot be used to do the separation. A kernel function is shown in (3).

$$K(x_i, x_j) \equiv \phi(x_i)\phi(x_j) \quad (3)$$

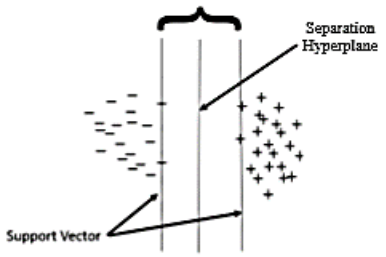


Figure 1. A SVM hyperplane on binary classes: positive and negative.

B. Existing Strategies on Multi-Label Classification

SVM is a one of the most famous classification technique. SVM has been proved to outperform other classification techniques [3, 10]. Therefore, SVM has been applied for multi-label classification. R-SVM applies SVM with threshold adjustment to minimize a bias of the majority class. The R-SVM collects best threshold from inducted SVM model and adjusts the threshold to apply with training subset that is generated from data set. Then, best threshold of each training subsets are collected and averaged to final threshold. Finally, the final threshold is applied on inducted model. Therefore, the R-SVM can improve accuracy of multi-label classification [4]. In [13], the OVA strategy is applied along with feature selection and undersampling technique to alleviate an imbalanced issue resulting in higher accuracy.

Recently, OVO strategy has been applied for multi-label classification. Twin SVM strategy trains two difference binary SVM classifiers to classify both of classes. A class label is then selected from vote score. Twin SVM has been applied with naïve bayes by using ensemble method [11]. This approach trains naïve bayes classifier for all possible classes and use twin SVM to train pair of all classifiers. Naïve Bayes (NB) classifier selects classes that have probability more than threshold. And, the class that has low probability is refined by twin SVM classifier. However, this approach takes double computing time on testing process. And, NB has been applied to reduce the computing time [11]. But, in testing process, unrelated data has been used in classifier, resulting to misclassification problem.

However, R-SVM and twin SVM with NB are not used to be our baseline method. Because, R-SVM is threshold adjustment method. Twin SVM with NB is the application of twin-svm with ensemble method. We are interested in undersampling approach. Therefore, OVA with undersampling is used to be our baseline.

C. Solution for Imbalance Issue

There are four solutions to solve imbalance problem, including sampling method, cost sensitive learning, recognition base, and ensemble-base [8].

Sampling method is a preprocessing approach for handling imbalance issue, including undersampling and oversampling. The undersampling randomly removes negative example to balance positive. However, undersampling may remove significant patterns and cause a loss of useful information. Oversampling may lead to the overfitting problem.

Cost sensitive learning assigns the highest cost of misclassification to positive class and inducts model with the lowest cost. This strategy may lead to overfitting problem.

Recognition based technique is one-class learning which learns on target class only. Many classifiers can't be inducted by using the recognition base.

Ensemble based approach is a combination of multiple classifier to improve prediction performance. The ensemble base takes more computing time and leads to overfitting problem.

III. Peformance Evaluation

For classification task, there are four base values to compare prediction result [12]. The accuracy can be calculated as the following equation (4) by using prediction criteria in Table I.

TABLE I. PRECDICTION CRITERIA

		Actual Class	
		A	***Not A***
Predicted Class	A	True Positive (TP)	False Positive (FP)
	Not A	False Negative (FN)	True Negative (TN)

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

For single label classification task, there are three base measurements, including perdition, recall, F_1. Precision is the part of positive prediction instances that are relevant (5). Recall is the part of positive instances that are correctly predict as positive (6). F_1 is a weight of precision and recall (7).

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$F_1 = \frac{2*Precision*Recall}{Precision+Recall} \quad (7)$$

TABLE II. MACRO-AVEARGING AND MICRO-AVERAGE ON LABEL-BASED (LB) MEASURES

LB-measures	Macro-averaging	Micro-averaging
Precision	$\frac{\sum Precision}{Class\ No.}$	$\frac{\sum TP}{\sum TP+\sum FP}$
Recall	$\frac{\sum Precision}{Class\ No.}$	$\frac{\sum TP}{\sum TP+\sum FN}$
F_1	$\frac{\sum F_1}{Class\ No.}$	$\frac{2 * Precision * Recall}{Precision + Recall}$

In multi-label classification task, there are three base measurements, including accuracy, label-base, and example-base. The label-base measurement, that has two measurements, including macro-average and micro-average. In macro-average, precision, recall and F_1 are computed for each class and averaged. So, the macro-F_1 is an equal weight for each class. The micro-average is computed by using the precision and recall that are summed all local value (*TP*, *FP*, *FN*, and *TN*). Then, the micro-F_1 is computed. The label-base measurement can be calculated as the following equation in TABLE II.

TABLE III. EXAMPLE-BASED (EB) MEASURES

Precision	Recall	F_1
$\frac{P_i \cap T_i}{P_i}$	$\frac{P_i \cap T_i}{T_i}$	$\frac{2 * Precision * Recall}{Precision + Recall}$

The example-base measurement is used to evaluate the performance for each example class. There are two example-base measurements, including macro-average and micro-average. Precision, recall and F_1 can be computed by follow equation in TABLE III. T_i is true class, when P_i is prediction class.

IV. A PROPOSED METHOD

OVO strategy is the well-known for multiclass classification. Using OVO with multi-label classification, the problem has been found in testing process. The problem is a having unrelated data in classifier. It can cause of misclassification Fig. 2. For example, there are four classes, *C1-C4*. Testing data that include class *C3* and *C4*, are added to classifier *C1vsC2*. Actually, classifier *C1vsC2* can classify either *C1* or *C2*. Therefore, if classifier *C1vsC2* predicts the testing data to be positive, the data that has class *C3* is assigned to be class *C1*. The classifier predicts testing data to be negative, the data that has class *C4* will be assigned to be class *C2*. These can lead to misclassification problem.

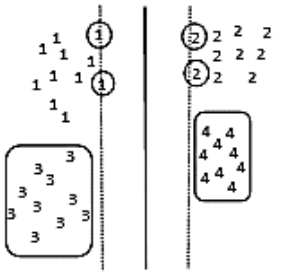


Figure 2. Classifier *C1-versus-C2*

Our approach collects negative result from each classifier and compared with threshold for class selection. The approach can be used to reduce the unrelated data in classifier. However, the imbalance issue has been found in some classifiers. Hence, the undersampling strategy has been used to solve the imbalance problem. A performance of our approach is enhanced.

The undersampling OVA strategy is our baseline method. Because, it is the most common approach of OVA. In the proposed framework, there are two main processes, including unbiased model induction and classification.

A. Unbiased Model Induction

The undersampling is applied for inducting model. This approach can reduce imbalance issue in process of classifier construction. Positive examples are selected from label. Negative examples are then randomly selected from pair of label. The number of negative examples must be equal to positive examples. Then, binary classifiers are created from each pair of classes. For example, there are four *classes C1-C4*. This approach create six binary classifiers as *C1vsC2, C1vsC3, C1vsC4, C2vsC3, C2vsC4*, and *C3vsC4*. Testing data are classified by the conducted model TABLE IV.

TABLE IV. A PESUDO COCE FOR THE PROCESS OF "UNBIASED MODEL INDUCTION"

Algorithm 1 Unbiased Model Induction

```
i=0
for each label do
    for each pair of label - i do
        select positive from label
        random select negative from pair label
        train classifier with SVM
    end for
    i+=1
end for
Return classification model
```

Negative results are collected from classifiers that associate with the label. Then, negative results are counted and compared with threshold. If number of negative result is higher than threshold, the class label isn't selected. On the other hand, the class label is selected, if the number of negative result is lower than or equal threshold (θ) (11). Finally, results of each classes at the same instance, are marge to be multi-label. For example, class *C1* that has classifier *C1vsC2, C1vsC3* and *C1vsC4*. In testing process, our approach collects negative voting from classifier *C1vsC2, C1vsC3*, and *C1vsC4*. Example 1, all classifier are predicted to be negative. The negative voting score is more than threshold. Class *C1* is not assigned to instance. Example 2, only classifier *C1vsC4* is predicted to be negative. Class *C1* is assigned to the instance. Because, negative voting score is lower than or equal to threshold TABLE V.

$$\#Negative \begin{cases} \leq \theta, assign\ positive \\ \geq \theta, do\ not\ assign\ \text{positive} \end{cases} \quad (11)$$

TABLE V. AN EXAMPLE OF APPLYING OUR NEGATIVE VOTING STRATEGY TO CLASSIFY C1, WHERE θ (MAX. NEGATIVE)=1

Testing Obs	*C1vsC2*	*C1vsC3*	*C1vsC4*	Predicted class
1	Not C1	Not C1	Not C1	Not C1
2	Not C2	Not C3	Not C1	C1

However, θ of each classes have a difference value. Therefore, the validation data set has been used to adjust the best threshold value of each classes. The threshold is then used in testing process TABLE VI.

TABLE VI. A PESUDO COCE FOR THE PROCESS OF "CLASSIFICATION"

Algorithm 2 Classification

```
threshold = θ
classify testing data with Unbiased Model
for each label do
    for each associated classifier do
        collect negative result from associated classifier
        count negative from associated classifier
        if count negative ≤ θ
        assign predicted label
    end for
end for
```

V. EXPERIMENTS

A. Data sets

In the experiment, there are five data sets including emotions, scene, tmc2007, yeast, and MIDB as shown details in TABLE VII.

TABLE VII. STATISTIC OF EXPERIMENTAL DATA SETS

Data set	*#Labels*	*#Examples*	*#Labels per example*
Yeast	14	2,417	4.237
Emotion	6	593	1.869
Scene	6	2,407	1.074
MIDB	28	120,919	2.000
Tmc2007	22	28,596	2.158

B. Experimental Results

There are three comparison methods, including undersampling One-vs-All (US-OVA), One-vs-One (OVO), and undersampling One-vs-One (US-OVO). They were evaluated by using five performance measures: accuracy, label-based macro F_1, label-based micro F_1, example-based macro F_1, and example-based micro F_1.

TABLE X shows accuracy of each strategy on five data sets. In comparison between baseline (US-OVA) and our methods, the results show that both of our OVO strategies are superior to the baseline. On average, an accuracy of OVO is higher than that of the baseline for 13.76% and US-OVO is even higher for 20.04%. Moreover, the undersampling strategy is really effective since US-OVO outperforms OVO on all data sets.

2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)

TABLE VIII. A COMPARISON BETWEEN BASELINE (US-OVA) AND OUR METHODS (OVO, US-OVO) IN TERMS OF LB ALONG WITH IMPROVEMENT PERCENTAGE FROM THE BASELINE F_1

Data set	Label-Based Macro F_1			Label-Based Micro F_1		
	US-OVA	*OVO*	*US-OVO*	*US-OVA*	*OVO*	*US-OVO*
Yeast	0.2981	0.3731 (+7.5%)	0.3849 (+8.68%)	0.3374	0.4586 (+12.12%)	0.4675 (+13.01%)
Emotion	0.1073	0.3044 (+19.71%)	0.3722 (+26.49%)	0.4537	0.3932 (-6.05%)	0.4012 (-0.525%)
Scene	0.3271	0.4652 (+13.81%)	0.6995 (+37.24)	0.3444	0.5729 (+22.85%)	0.6876 (+34.32%)
MIDB	0.1179	0.1340 (+1.61%)	0.1376 (+1.35)	0.1966	0.3520 (+15.54%)	0.3657 (+16.91%)
Tmc2007	0.2075	0.3143 (+10.68%)	0.4321 (+22.46%)	0.2117	0.3374 (+12.57%)	0.4639 (+25.22%)
Average	0.2115	0.3182 (+10.67%)	0.4052 (+19.37%)	0.3086	0.4228 (+11.42%)	0.4771 (+16.85%)

TABLE IX. A COMPARISON BETWEEN BASELINE (US-OVA) AND OUR METHODS (OVO, US-OVO) IN TERMS OF EB ALONG WITH IMPROVEMENT PERCENTAGE FROM THE BASELINE F_1

Data set	Example-Based Macro F_1			Example-Based Micro F_1		
	US-OVA	*OVO*	*US-OVO*	*US-OVA*	*OVO*	*US-OVO*
Yeast	0.4381	0.5533 (+11.42%)	0.5912 (+15.31%)	0.4463	0.5619 (+11.56%)	0.6051 (+15.88%)
Emotion	0.6003	0.6131 (+1.31%)	0.6191 (+1.91%)	0.6103	0.6142 (+0.39%)	0.6279 (+1.76%)
Scene	0.5604	0.7408 (+18.04%)	0.8418 (+28.04%)	0.5606	0.7473 (+19.67%)	0.8372 (+27.66%)
MIDB	0.2559	0.4229 (+16.7%)	0.5223 (+26.64%)	0.2475	0.4281 (+18.06%)	0.5126 (+26.51%)
Tmc2007	0.3059	0.4499 (+14.40%)	0.5879 (+28.20%)	0.3097	0.4563 (+14.66%)	0.5958 (+28.61%)
Average	0.4321	0.5560 (+12.39%)	0.6324 (+20.03%)	0.4348	0.5615 (+12.67%)	0.6357 (+20.09%)

TABLE X. A COMPARISON BETWEEN BASELINE (US-OVA) AND OUR METHODS (OVO, US-OVO) IN TERMS OF ACCURACY ALONG WITH IMPROVEMENT PERCENTAGE FROM THE BASELINE

Data set	*US-OVA*	*OVO*	*US-OVO*
Yeast	0.5186	0.6107 (+9.21%)	0.7096 (+19.1%)
Emotion	0.6446	0.6527 (+0.81%)	0.6625 (+1.79%)
Scene	0.5560	0.7937 (+23.77%)	0.8867 (+33.07%)
MIDB	0.4080	0.7746 (+36.66%)	0.8406 (+43.26%)
Tmc2007	0.8989	0.9097 (+1.08%)	0.9290 (+3.01%)
Average	0.6052	0.7428 (+13.76%)	0.8056 (+20.04%)

On performance evaluation of our method and baseline using label-based macro and micro F_1, the results are shown in TABLE VIII. The label-based macro F_1 of our method is higher than the baseline (10.67%). Moreover, the undersampling affects to performance of classification (19.37%). The label-based micro F_1 of our method is higher than the baseline (11.42%). Performance of label-based micro F_1 is increased, when applied the undersampling strategy (16.85%). Therefore, our method is prove to outperform baseline in label-based F_1 term.

TABLE IX represents performance evaluation of our method and baseline in terms of example-based. There two evaluators, including macro F_1 and micro F_1. In term of macro F_1, the result show that our method is better than baseline (12.39%). Applying undersampling, macro F_1 of our method is higher than baseline (20.03%). In term of micro F_1 the result show that our approach is better than baseline (12.67%). Using undersampling, micro F_1 of our approach is higher than baseline (20.09%). Therefore, our method can predict result closet the true class than baseline.

VI. CONCLUSION

In this paper, we propose a framework that applied OVO strategy with SVM for enhancing accuracy of multi-label classification. In term of accuracy, the result in TABEL X shows that our proposed method outperforms OVA all data sets. For label-based F_1, the result in TABLE VIII indicates that our method can improve performance of multi-label classification task better than OVA. In term of example-based F_1, the result in TABEL IX represents that our method has more efficiency to predict the result that close to true class than OVA. Therefore, our approach outperformed OVA. Although, OVA had been applied with undersampling to solve the imbalance issue. Our approach is significantly better than OVA.

REFERENCES

[1] M. S. Sorower, "A literature survey on algorithms for multi-label learning," Technical report, Oregon State University, Corvallis, OR, USA, 2010.

[2] G. Tsoumakas, and I. Katakis, "Multi-label classification: An overview," Int. J. Data Warhouse., vol. 3, pp. 1-13, 2007.

[3] J. Xu, "An extended one-versus-rest support vector machine for multi-label classification," Neurocomputing, vol. 74, pp. 3114-3124, 2011.

[4] P. Vateekul, S. Dendamrongvit and M. Kubat, "Improving SVM performance in multi-label domains: Threshold adjustment," Int. J. Artif. Intell. T., vol. 22, 2013.

[5] T. Anunpiriyakul, P. Poomsirivilai, and P. Vateekul, "Label correction strategy on hierarchical multi-label classification," Lect. Notes. Comput. Sc., vol. 8556, pp. 213-227, 2014.

[6] G. Anthony, H. Gregg and M. Tshilidzi, "Image classification using SVMs: One-against-one vs one-against-all," The 28th Asian Conference on Remote Sensing, 2007.

[7] R. K. Eichelberger, and V. S. Sheng, "Does one-against-all or one-against-one improve the performance of multiclass classifications," The 27th AAAI Conference on Artificial Intelligence, 2013.

[8] S. Merghani, A. Ebrahman, and A. Abraham, "A review of class imbalance problem," JNIC., vol. 1, pp. 332-340, 2013.

[9] X. Y. Liu, J. Wu, and Z. H. Zhou, "Exploratory undersampling for class-imbalance learning," IEEE Trans. Syst. Man. Cybern. B. Cybern., vol. 39, pp. 539-550, 2009.

[10] T. Choeikiwong, and P. Vateekul, Software defect prediction in imbalanced data sets using unbiased support vector machine, Lecture Notes in Electrical Engineering, vol. 339, 2015, pp. 923-931.

[11] B. Zhang, X. Xu, and J. Su, "An ensemble method for multi-class and multi-label text categorization," The international conference on intelligent systems and knowledge engineering, 2007.

[12] N. Phachongkitpkiphat, and P. Vateekul, "An improvement of flat approach on hierarchical text classification using top-level pruning classifiers," The 11th International Joint Conference on Computer Science and Software Engineering, pp. 86-90, 2014.

[13] S. Dendamrongvit and M. Kubat, "Undersampling Approach for Imbalanced Training Sets and Induction from Multi-label Text-Categorization Domains," New Frontiers in Applied Data Mining. Springer Berlin Heidelberg, pp. 40-52, 2009.

ภาคผนวก ข.

บทความทางวิชาการณ์เรื่อง “Applying One-Versus-One SVMs to Classify Multi-Label Data with Large Labels Using Spark” โดย สุทธิพงษ์ แดงด้วง และ พีรพล เวทีกูล ในงานประชุมวิชาการ 9th International Conference on Knowledge and Smart Technology จัดขึ้น ณ จังหวัด ชลบุรี ประเทศไทย วันที่ 1 กุมพาพันธ์ 2560 ถึง วันที่ 4 กุมพาพันธ์ 2560

Applying One-Versus-One SVMs to Classify Multi-Label Data with Large Labels Using Spark

Abstract—**Multi-Label classification aims to classify an example that can belong to many classes. Although One-versus-All (OVA) is the most common approach, our prior work has shown that the proposed One-versus-One (OVO) always gives higher prediction accuracy than OVA. However, OVO requires an extremely high computational cost when there are a large number of labels. In this paper, we apply our OVO SVMs on the proposed Spark framework along with a mechanism to split a job into a set of small jobs and then process them in parallel. The framework can induce OVO SVMs very fast, while maintaining the prediction accuracy even though there are a large number of classes. The experiment was conducted on five standard benchmarks. The result shows that our framework can really reduce computing time on Spark environment, while significantly outperforms OVA in terms of F1 on all data.**

Keywords- one-versus-one; support vector machine; spark; map-reduce

I. INTRODUCTION

Multi-Label Classification (ML) is a supervised learning task where a single example can be associated with many classes [1-4]. Recently, it has increasingly been require in wide range of an application such as text categorization, sematic image labeling, bioinformatics and music categorization instruction. ML is categorized into two main methods including Algorithm Adaption (AA) and Problem Transform (PT) [5]. AA is concerned with developing a new algorithm where is specific to ML. PT is a technique that transforms ML to a set of binary classification task. This technique is very popular applied on many classification techniques. It is suitable for any domains resulting in high performance accuracy. PT is classified into two strategies: One-versus-All (OVA) and One-versus-One (OVO).

OVA strategy creates a set of binary classifiers equal to the number of classes. For *i-th* class, all examples in this class are labeled as positives, while the other examples are labeled as negatives. [6]. OVO strategy induces a set of all class-pair binary classifiers. The prediction result is the class with the highest vote by all classifiers. Although, OVO is popular and has been proved to outperform OVA in the multiclass domain [6, 7]. However, this strategy is unsuitable for large class example. Because, it creates many classifier. Fig. 1 indicates that the number of classifier is depend on number of class.

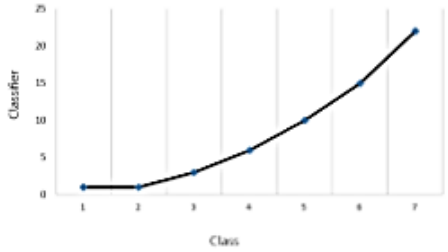


Figure 1. Growth rate of classifier per classes

Since, the real world data is increase. Computing time is subsequently increased due to data size. High performance gadget is used to process data. However, distributed computing system is developed to solve the limitation [8-13]. This system use map-reduce technique to distribute data to each computing node for processing the data. Map-reduce technique consists of map and reduce process. Map distributes and transforms data to key, value (k, v) format. Reduce is used to reduce k, v from each compute nodes to key, list of value (k, list (v)). The distributed computing system has two famous systems: Hadoop and Spark. Hadoop is based on disk while spark is based on memory [12, 13]. However, spark was proved to be high performance than Hadoop [12].

In this paper, we propose a multi-label framework that applies OVO and map-reduce technique to improve prediction accuracy and decrease computing time. Our framework is called "MRUS-OVO". Support Vector Machine (SVM) is our baseline

classifier due to its success reported in prior works [3]. The experiment is conducted on five standard data sets. The results show that our approach outperforms OVA on all data sets in terms of F1 and reduce computing time by applying map-reduce framework

II. Related Works

A. Support Vector Machine

SVM algorithm is widely used to be classifier for multi-label classification. It is based on the concept of hyperplane to define decision boundaries. Amount set of objects that have different class memberships are separated by the hyperplane. The purpose of SVM is to induce a hyperplane function in equation (1), where $\vec{w}$ is a weight vector and b is a bias [4, 5, 14]. In equation (2), it shows the optimization function to construct SVM hyperplane, where is a penalty parameter of misclassifications. In a non-linear separable problem, SVM handles this by using a kernel function non-linear to map the data into a higher space, where a linear hyperplane cannot be used to separate the different memberships. A kernel function is shown in equation (3).

$$h(\vec{w}, b) = \vec{w} \times (x + b) \quad (1)$$

$$Minimize_{w,b,\xi} \ \frac{1}{2} w^T w + C \sum_{i=1}^{|D|} \xi_i$$

$$subject\ to\ y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \quad , \xi_i \geq 0 \quad (2)$$

$$K(x_i, x_j) \equiv \phi(x_i)\phi(x_j) \quad (3)$$

B. Existing Strategies on Multi-Label Classification

SVM which is a one of the most famous classification technique has been proved to outperform other classification techniques [3, 15]. Therefore, SVM has been applied for multi-label classification. R-SVM applies SVM with threshold adjustment to minimize a bias of the majority class. The R-SVM collects best threshold from inducted SVM model and adjusts the threshold to apply with training subset that is generated from data set. Then, best threshold of each training subsets are collected and averaged to final threshold. Finally, the final threshold is applied on inducted model. Therefore, the R-SVM can improve accuracy of multi-label classification [4]. In [13], the OVA strategy is applied along with feature selection and undersampling technique to alleviate an imbalanced issue resulting in higher accuracy.

Nowadays, OVO strategy has been applied for multi-label classification. Twin SVM strategy trains two difference binary SVM classifiers to classify both of classes. A class label is then selected from vote score. Twin SVM has been applied with naïve bayes by using ensemble method [16]. This approach trains naïve bayes classifier for all possible classes and use twin SVM to train pair of all classifiers. Naïve Bayes (NB) classifier selects classes that have probability more than threshold. And, the class that has low probability is refined by twin SVM classifier. However, this approach takes double computing time on testing process. And, NB has been applied to reduce the computing time [16]. But, in testing process, unrelated data has been used in classifier, resulting to misclassification problem. Therefore, in previous work we propose method call "negative voting" that can solve this problem [17]. This method is focus on negative result that use to remove unrelated data in testing process.

However, R-SVM and twin SVM with NB are not used to be our baseline method. Because, R-SVM is threshold adjustment method. Twin-SVM with NB is the application of twin-SVM with ensemble method. We are interested in undersampling approach. Therefore, OVA with undersampling is used to be our baseline.

C. Existing Map-reduce Works

Map-reduce is a framework that include map and reduce process [8-13]. The map-reduce framework is widely used to process the data that have a large size and more complexity. Map-reduce uses the k,v pair data type in map and reduce process. Map process distributes the data to each computing node and transforms data to k, v format. And reduce process is used to merge result that is computed from each compute nodes to k, list (v). An overview of the map-reduce system is shown in Fig. 2.

Figure 2. Map-reduce frame work

Reference [8] applies SVM with map-reduce framework. In map process it will split training data to a subset data that has deference size and use local SVM to train model for find local weight. After that, reduce process sum up the partial weight vectors to global weight if SVM is linear. In the other hand, if SVM is nonlinear in reduce process joins partial alpha arrays to produce global alpha array. Reference [11] applies SVM with map-reduce framework. In map process, it will split training data to a subset data and use local SVM to employ model for find local support vector. Then, local support vectors merge to find global support vector in reduce process.

III. Propose Method

Recently, the growth of data is increasing in complexity, size and relevance. The single machine cannot be use to process data. Subsequently, the development of distributed computing system was attended. Here, this work aims to apply map-reduce on multi-label classification using OVO SVMs. Using OVO strategy with SVM, unrelated data is found in testing process. It can lead to misclassification problem Fig. 3. Therefore, this work use negative voting method to remove the unrelated data in testing process and adapt threshold adjusting to select multi-label. A framework consists of two main process:

construction of OVO classifier and applying of map-reduce.

A. *Negative voting*

Negative voting is used to label the example by comparing the results that collect from classifiers that associate with the label to remove unrelated data in classifier. Because, the unrelated data is found in testing process when applying OVO strategy with SVM. For example, class C1 that has classifier C1vsC2, C1vsC3 and C1vsC4. In testing process, testing data was appear nearly class C3. However by using classifier C1vsC2 or C1vsC4 these classifier can predict only class C1, C2 or C4. It can lead to misclassification problem. This method was focus on negative to remove unrelated class to avoid misclassification. For example, C1vsC2 was predicted to class C1 then C2 was removed from this example. C1vsC3 was predicted to class C3 then C1 was remove from this example. C1vsC4 was predicted to class C1 then C4 was removed from this example. This example was labeled to class C3 (Fig. 4). Then, negative score are counted and apply with threshold adjusting to label the example more than one class. If number of negative score is higher than threshold, the class label isn't selected. On the other hand, the class label is selected, if the number of negative score is lower than or equal threshold (θ).

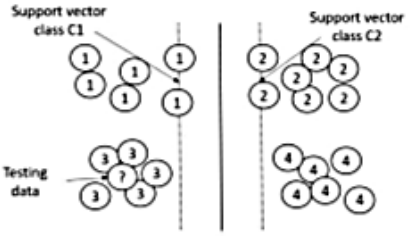


Figure 3. Classifier C1-versus-C2.

B. *Applying of map-reduce,*

In apply distributed computing system, this work use spark system. Because, the system was proved to be highest performance than hadoop. Spark store data in memory. This storage is called "Resilient Distributed Dataset" (RDD). In this work, we use six computing nodes including one master node and five clusters. Performance of each computing node is 8 core 2.50 GHz Cpu and 12 GB memory.

This process was inducted a set of small job to employ set of classifier by using SVM. Each job have a same number of classifier. In map process, training and testing data are stored into memory. Then, pointer of each classes is read to list for filter the training data to create training set of each classifier. Then induct the classification model by using SVM. After that, the model is tested by testing data. In reduce process, the result of each classifier that associates *i-th* class is collected. And, negative voting method is used to label the examples. Finally, the reduce function is used to merge the results that have the same instance to multi-label result.

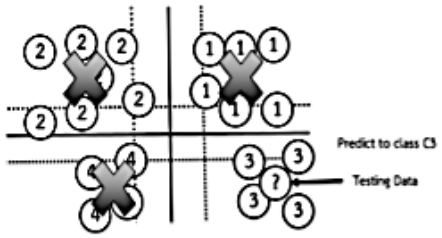


Figure 4. Classifier that associated class C1 including C1vsC2, C1vsC3 and C1vs C4.

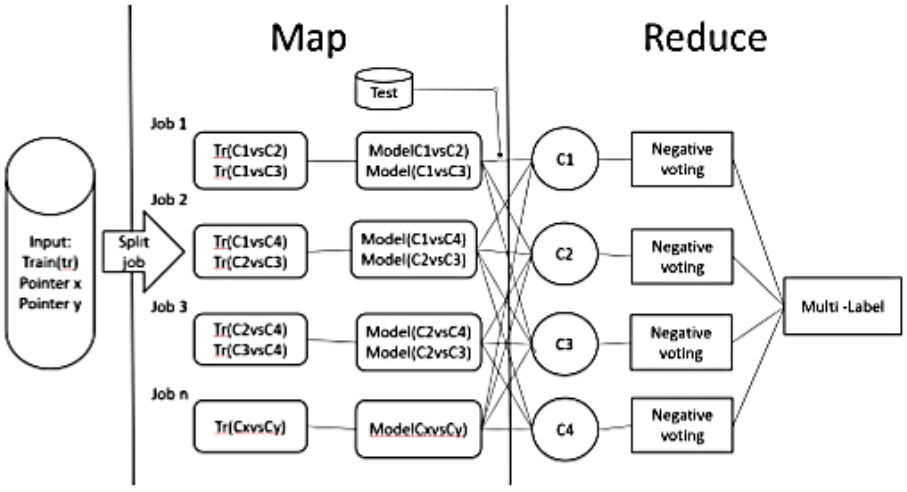


Figure 5. Our map-reduce frame work

IV. Experiment

A. Datasets

In the experiment, there are five data sets including yeast, scene, tmc2007, rcv1v2 (subset) and birds as shown details in Table I.

TABLE I. Datasets description

Data sets	#Labels	#Examples	#Labels per example	#Total classifier
Yeast	14	2,417	4.23	91
Scene	6	2,407	1.07	15
Tmc 2007	22	28,596	2.158	231
Rcv1v2 (subsef)	101	6000	3.226	5,050
Birds	19	645	1.01	171

B. Experimented Result

For computing time evaluation, there are four comparison map-reduce 1, 2, 3, 5 jobs. They are evaluated by using 3 performance measures: training time, testing time and speed up rate.

For prediction performance evaluation, there are two comparison methods, including undersampling One-vs-All (US-OVA), and map-reduce undersampling One-vs-One (MRUS-OVO). They are evaluated by using five performance measures: accuracy, label-based macro F1, label-based micro F1, example-based macro F1, and example-based micro F1.

Table IV shows accuracy of our strategy on five datasets. In comparison between baseline (US-OVA) and our methods, the results show that our MRUS-OVO strategies are superior to the baseline. On average, MRUS-OVO is even higher for 16.39%.

On performance evaluation, our method and baseline using label-based macro and micro F1, the results are shown in Table II. The label-based macro F1 of our method is higher than the baseline (14.90%). The label-based micro F1 of our method is higher than the baseline (26.80%). Therefore, our method is proved to outperform baseline in label-based F1 term.

Table III represents performance evaluation of our method and baseline in terms of example-based. There are two evaluators, including macro F1 and micro F1. In term of macro F1, the result shows that our method is better than baseline (26.64%). In term of micro, F1 the result shows that our approach is better than baseline (26.09%). Therefore, our method can predict result closet the true class than baseline.

TABLE II. A comparison between baseline (US-OVA) and our methods (MRUS-OVO) in terms of accuracy along with improvement percentage from the baseline

Datasets	US-OVA	MRUS-OVO
Yeast	0.5186	0.7618 (+24.82%)
Scene	0.8770	0.8935 (+1.65%)
Tmc2007	0.8983	0.9576 (+5.93%)
Rcv1v2 (subset)	0.6534	0.9430 (+28.96)
Birds	0.7023	0.9133 (+21.10)
Average	0.7299	0.8938 (+16.39%)

Table V shows training time, testing time, total time and speed up. These indicate that our method can reduce the computing time. This work evaluates the proposed method by using five standard datasets that have difference size. The result indicates that computing time is growth depend on number of class when use OVO technique. However, our method can reduce computing time of all datasets and can remain the high accuracy prediction.

TABLE III. A comparison between baseline (US-OVA) and our methods (MRUS-OVO) in terms of LB along with improvement percentage from the baseline F_1

Datasets	Label-Based Macro F1		Label-Based Micro F1	
	US-OVA	MRUS-OVO	US-OVA	MRUS-OVO
Yeast	0.3501	0.3977 (+4.76%)	0.4207	0.5931 (+17.24%)
Scene	0.6571	0.7454 (+8.83%)	0.7901	0.8304 (+4.06%)
Tmc2007	0.2076	0.5372 (+32.96%)	0.2117	0.7980 (+58.63%)
Rcv1v2 (subset)	0.0289	0.0920 (+6.31%)	0.1051	0.2780 (+17.29%)
Birds	0.0727	0.2889 (+21.62%)	0.2943	0.6627 (+36.84%)
Average	0.2632	0.4122 (+14.90%)	0.3643	0.6324 (+26.80%)

TABLE IV. A COMPARISON BETWEEN BASELINE (US-OVA) AND OUR METHODS (MRUS-OVO) IN TERMS OF EB ALONG WITH IMPROVEMENT PERCENTAGE FROM THE BASELINE F_1

Data set	Example-Based Macro F1		Example-Based Micro F1	
	US-OVA	MRUS-OVO	US-OVA	MRUS-OVO
Yeast	0.4912	0.5771 (+8.59%)	0.5020	0.5901 (+8.81%)
Scene	0.7903	0.8412 (+5.09%)	0.7921	0.8342 (+4.21%)
Tmc2007	0.3060	0.8048 (+49.88%)	0.3098	0.7980 (+48.82%)
Rcv1v2 (subset)	0.1047	0.4188 (+31.41%)	0.1051	0.4234 (+31.83%)
Birds	0.2907	0.6803 (+38.96%)	0.2943	0.6627 (+36.84%)
Average	0.3965	0.6630 (+26.64%)	0.4006	0.6616 (+26.09%)

TABLE V. COMPUTING TIME (MINUTE) FOR MRUS-OVO WITH THREE STANDARD DATASETS

Datasets	Training Time (min)				Testing time (min)				Speed up
	1	2	3	5	1	2	3	5	
Yeast	25.24	20.39	19.79	16.41	0.14	0.11	0.10	0.06	1.54
Scene	4.01	2.52	2.01	1.75	0.025	0.014	0.009	0.007	2.29
Tmc2007	42.73	26.23	20.24	17.50	0.76	0.35	0.23	0.17	2.46
Rcv1v2 (subset)	3,361.02	2,259.04	1,926.43	1,184.26	440.45	230.52	156.12	86.80	2.99
Birds	322.67	189.18	117.00	76.41	0.62	0.35	0.28	0.21	3.03
Average	752.93	499.47	417.09	259.26	88.41	46.26	31.34	17.45	3.04

V. CONCLUSION

In this paper, we aim to increase an accuracy of multi-label classification by applying our One-versus-One (OVO) strategy on SVMs; however, the computational cost can be increased when the number of labels is large. To overcome this limitation, we propose a process to split and employ a set of classifiers in parallel on Spark. There are five experimental datasets. The result showed that F1 of our OVO SVMs is superior to that of the baseline, One-versus-All (OVA). Also, the computational cost can really be reduced by half on average in all data sets when there are five sub-jobs running on Spark.

REFERENCES

[1] M. S. Sorower, "A literature survey on algorithms for multi-label learning," Technical report, Oregon State University, Corvallis, OR, USA, 2010.

[2] G. Tsoumakas, and I. Katakis, "Multi-label classification: An overview," Int. J. Data Warhouse., vol. 3, pp. 1-13, 2007.

[3] J. Xu, "An extended one-versus-rest support vector machine for multi-label classification," Neurocomputing, vol. 74, pp. 3114-3124, 2011.

[4] P. Vateekul, S. Dendamrongvit and M. Kubat, "Improving SVM performance in multi-label domains: Threshold adjustment," Int. J. Artif. Intell. T., vol. 22, 2013.

[5] T. Ananpiriyakul, P. Poomsirivilai, and P. Vateekul, "Label correction strategy on hierarchical multi-label classification," Lect. Notes. Comput. Sc., vol. 8556, pp. 213-227, 2014.

[6] G. Anthony, H. Gregg and M. Tshilidzi, "Image classification using SVMs: One-against-one vs one-against-all," The 28th Asian Conference on Remote Sensing, 2007.

[7] R. K. Eichelberger, and V. S. Sheng, "Does one-against-all or one-against-one improve the performance of multiclass classifications," The 27th AAAI Conference on Artificial Intelligence, 2013.

[8] F. Ö. Çatak, M. E. Balaban, "A Map-reduce-based distributed SVM algorithm for binary classification," Turk. J. Elec. Eng. & Comp. Sci., vol. 24, pp.863—873, 2016.

[9] C. Y. Lin, C. H. Tsai, , C. P. Lee, C. J. Lin,"Large-scale logistic regression and linear support vector machines using Spark," 2014 IEEE International Conference on Big Data., pp. 519-528, 2014.

[10] J. Maillo, I. Triguero, F. Herrera,"A Map-reduce-Based k-Nearest Neighbor Approach for Big Data Classification," Trustcom/BigDataSE/ISPA, 2015 IEEE, vol. 2, pp.167-172, 2015.

[11] N. K. Alham, M. Li, Y. Liu, S. Hammoud,"A Map-reduce-based distributed SVM algorithm for automatic image annotation," Comput. Math. Appl., vol.62, pp.2801-2811,2011.

[12] "Apache Spark", http://spark.apache.org/

[13] "Apache Hadoop", http://hadoop.apache.org/

[14] T. Choeikiwong, P. Vateekul,"Software defect prediction in imbalanced data sets using unbiased support vector machine," Lecture Notes in Electrical Engineering, vol.339, pp.923-931, 2015.

[15] S. Dendamrongvit and M. Kubat, "Undersampling Approach for Imbalanced Training Sets and Induction from Multi-label Text-Categorization Domains," New Frontiers in Applied Data Mining. Springer Berlin Heidelberg, pp. 40-52, 2009.

[16] B. Zhang, X. Xu, J. Su,"An ensemble method for multi-class and multi-label text categorization," The international conference on intelligent systems and knowledge engineering, 2007.

[17] S. Daengduang, P.Vateekul,"Enhancing Accuracy of Multi-Label Classification by Applying One-vs-One Support Vector Machine," 13th International Joint Conference on Computer Science and Software Engineering, 2016.

ภาคผนวก ค.

ในการปรับค่าพารามิเตอร์ของวิธีซัพพอร์ตเวคเตอร์แมชชีนโดยใช้เคอร์เนล RBF นั้น จำเป็นต้องปรับค่าพารามิเตอร์ที่มีความเหมาะสมกับแต่ละชุดข้อมูล ซึ่งค่า gamma ของแต่ละชุดข้อมูลนั้นมีค่าต่างกัน โดยจะเลือกค่าพื้นฐานเป็นค่าแรกแล้วทำการปรับเพิ่มจำนวนขึ้นเรื่อย ๆ ค่าพื้นฐานของเคอร์เนล RBF นั้นจะสามารถคำนวณได้จาก 1ต่อจำนวนลักษณะข้อมูล ซึ่งค่าที่จะเพิ่มขึ้นมี 4 ค่า ดังนี้ ค่าพื้นฐาน + 0.01 ค่าพื้นฐาน + 0.1 ค่าพื้นฐาน + 0.5 และค่าพื้นฐาน + 1 โดยสามารถอ้างอิงค่า gamma ที่ใช้ในแต่ละชุดข้อมูลได้ในตารางที่ 15 โดยการทดลองด้วยวิธี 5 fold cross-validation นั้นในแต่ละรอบจะทำการเลือกค่า gamma ที่มีความเหมาะสมที่สุดในรอบนั้น ๆ โดยค่าที่ใช้ในแต่ละรอบของวิธีจำแนกแบบหนึ่งต่อหนึ่งที่มีการประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูลจะเป็นดังตารางที่ 16 และค่าที่ใช้ในแต่ละรอบของวิธีการจำแนกแบบหนึ่งต่อทั้งหมดซึ่งประยุกต์ใช้งานวิธีอันเดอร์แซมพลิ่งจะเป็นดังตารางที่ 17

ตารางที่ 15 ค่าพารามิเตอร์ gamma ของแต่ละชุดข้อมูล

ชุดข้อมูล	ค่าพารามิเตอร์ gamma				
	1	2	3	4	5
Yeast	0.009	0.019	0.109	0.509	1.009
Emotion	0.013	0.023	0.113	0.513	1.013
IMDB	0.001	0.011	0.101	0.501	1.001
Tmc2007	0.002	0.012	0.102	0.502	1.002
Birds	0.003	0.013	0.103	0.503	1.003
Rcv1v2	0.002	0.012	0.102	0.502	1.002

ตารางที่ 16 ค่า gamma ที่มีความเหมาะสมที่สุดในการทดลองแต่ละรอบของวิธีจำแนกแบบหนึ่งต่อหนึ่งที่มีการประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล

ชุดข้อมูล	ค่า gamma ที่มีความเหมาะสมในแต่ละรอบ				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Yeast	1.009	0.509	0.109	1.009	0.019
Emotion	0.113	0.023	0.013	0.023	0.013
IMDB	0.101	0.101	1.001	0.011	0.501
TMC2007	0.102	0.012	0.102	0.012	0.502
Birds	0.013	0.103	0.103	0.503	0.013
Rcv1V2	0.012	0.102	0.102	0.502	1.002

ตารางที่ 17 ค่า gamma ที่มีความเหมาะสมที่สุดในการทดลองแต่ละรอบของวิธีจำแนกแบบหนึ่งต่อหนึ่งที่มีการประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล

ชุดข้อมูล	ค่า gamma ที่มีความเหมาะสมในแต่ละรอบ				
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Yeast	0.109	0.009	0.509	0.109	0.019
Emotion	0.013	0.023	1.013	0.013	0.113
IMDB	0.011	0.101	0.501	0.011	0.101
TMC2007	0.012	0.102	1.002	0.102	0.012
Birds	0.103	0.003	0.013	0.103	0.103
Rcv1V2	0.102	0.102	0.012	0.502	1.002

ค่าขีดแบ่งถูกใช้ในการเลือกตอบข้อมูลแบบหลายฉลากของวิธีการจำแนกแบบหนึ่งต่อหนึ่งที่มีการประยุกต์ใช้วิธีจัดลำดับความสำคัญของกลุ่มข้อมูล โดยค่าขีดแบ่งที่ถูกใช้ในการเลือกตอบข้อมูลแบบหลายฉลากในการทดลองที่ 5.3 นั้นได้ถูกแสดงในตารางที่ 16

ตารางที่ 18 ค่าขีดแบ่งที่ใช้ในการเลือกตอบของวิธีจัดลำดับความสำคัญของกลุ่มข้อมูล

ชุดข้อมูล	ค่าขีดแบ่งที่กำหนด					จำนวนกลุ่มข้อมูล
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Yeast	11	8	12	13	10	14
Emotion	4	4	3	4	3	6
IMDB	25	27	26	27	24	28
Tmc2007	17	15	16	16	17	22
Birds	15	14	13	16	15	19
Rcv1v2	72	77	78	94	97	101

ภาคผนวก ง.

การทดลองด้วยวิธี 5 fold cross-validation ผู้วิจัยได้ใช้งานเครื่องมือ spark-sklearn [24] โดยที่เครื่องมือชิ้นนี้จะทำการแบ่งข้อมูลให้มีจำนวนข้อมูลเท่า ๆ กันด้วยวิธี StratifiedKFold ซึ่งจะทำการแบ่งข้อมูลออกเป็น 5 ส่วน ได้โดยการกำหนดค่าพารามิเตอร์ cv ให้มีค่าเท่ากับ 5 เพื่อใช้ในการแบ่งข้อมูล ดังรูปที่ 12

cv : *integer or cross-validation generator, default=3*
A cross-validation generator to use. If int, determines the number of folds in StratifiedKFold

รูปที่ 12 คำอธิบายวิธีการแบ่งข้อมูลในการทำ cross-validation โดยใช้เครื่องมือ spark-sklearn

ประวัติผู้เขียนวิทยานิพนธ์

นาย สุทธิพงษ์ แดงด้วง เกิดวันที่ 19 กรกฎาคม พศ.2534 สำเร็จการศึกษาในระดับมัธยมศึกษา ณ โรงเรียนวัดสุทธิวราราม ต่อมาได้เข้าศึกษาต่อในระดับปริญญาตรี ณ มหาวิทยาลัยกรุงเทพ คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ ในปี พ.ศ. 2553 และในปี พ.ศ. 2557 จึงได้สำเร็จการศึกษาปริญญา วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2557