การจำแนกข้อความขนาดใหญ่แบบหลายฉลากมีลำดับชั้น
โดยใช้วิธีการแบบแฟลตด้วยยุทธศาสตร์ตัดเล็มแบบเอสวีเอ็ม

นายณัฐชนน ผจงกิจพิพัฒน์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

LARGE-SCALE HIERARCHICAL MULTI-LABEL TEXT CLASSIFICATION USING FLAT APPROACH WITH SVM PRUNING STRATEGY

Mr. Natchanon Phachongkitphiphat

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

หัวข้อวิทยานิพนธ์	การจำแนกข้อความขนาดใหญ่แบบหลายฉลากมีลำดับชั้นโดยใช้วิธีการแบบแฟลตด้วยยุทธศาสตร์ตัดเล็มแบบเอสวีเอ็ม
โดย	นายณัฐชนน ผจงกิจพิพัฒน์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ดร.พีรพล เวทีกูล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ดร.พีรพล เวทีกูล)

.....กรรมการภายนอกมหาวิทยาลัย

(รองศาสตราจารย์ ดร.กฤษณะ ไวยมัย)

ณัฐชนน ผจงกิจพิพัฒน์ : การจำแนกข้อความขนาดใหญ่แบบหลายฉลากมีลำดับชั้นโดยใช้วิธีการแบบแฟลตด้วยยุทธศาสตร์ตัดเล็มแบบเอสวีเอ็ม (LARGE-SCALE HIERARCHICAL MULTI-LABEL TEXT CLASSIFICATION USING FLAT APPROACH WITH SVM PRUNING STRATEGY) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ดร.พีรพล เวทีกูล, 68 หน้า.

การจำแนกประเภทแบบหลายฉลากมีลำดับชั้น เป็นการจำแนกประเภทที่รวมลักษณะเฉพาะของปัญหาสองรูปแบบคือ ข้อมูลแต่ละตัวอาจจัดอยู่ได้ในหลายคลาส และคลาสเหล่านี้มีความสัมพันธ์เป็นโครงสร้างลำดับชั้น ซึ่งข้อมูลในชีวิตจริงมักจะมีลักษณะซับซ้อนเช่นนี้ การจำแนกประเภทข้อความแบบหลายฉลากมีลำดับชั้น เป็นหัวข้อการวิจัยที่ได้รับความสนใจอย่างมากในปัจจุบัน เพราะโครงสร้างลำดับชั้นใช้อธิบายความสัมพันธ์ของข้อมูลประเภทข้อความได้ดี ข้อมูลประเภทข้อความที่เราพบอยู่ทุกวันก็คือ ข้อมูลบนเว็บไซต์นั่นเอง เว็บไซต์ที่เพิ่มจำนวนขึ้นอย่างรวดเร็วทำให้เว็บอย่างเว็บไดเรกทอรีและวิกิพีเดียจำเป็นต้องมีระบบการจำแนกประเภทอย่างอัตโนมัติเมื่อมีหน้าเว็บใหม่เข้ามาในฐานข้อมูล ด้วยข้อมูลมหาศาลเช่นนี้ ปัญหานี้จึงถือเป็นการจำแนกประเภทขนาดใหญ่แบบหลายฉลากมีลำดับชั้น งานวิจัยหลายงานนำเสนอวิธีแก้ปัญหาการจำแนกประเภทแบบหลายฉลากมีลำดับชั้น แต่วิธีเหล่านั้นประมวลผลข้อมูลขนาดใหญ่ไม่ได้ เนื่องจากการประมวลผลอาจต้องใช้พื้นที่เก็บข้อมูลขนาดใหญ่มาก อาจใช้เวลาประมวลผลนานเกินไป หรือทำนายคลาสได้ไม่แม่นยำ บางวิธีการที่พอจะรองรับข้อมูลขนาดใหญ่ได้ก็ไม่ได้นำโครงสร้างลำดับชั้นมาใช้ให้เกิดประโยชน์

งานวิจัยนี้จึงได้นำเสนอการจำแนกข้อความขนาดใหญ่แบบหลายฉลากมีลำดับชั้นที่ปรับปรุงวิธีการ k-NN ซึ่งเป็นวิธีการแบบแฟลต และนำโครงสร้างลำดับชั้นมาใช้ด้วยการฝึกตัวจำแนกประเภท SVM ที่โหนดชั้นบนของโครงสร้างลำดับชั้น เพื่อช่วยกรองคำตอบให้มีความถูกต้องแม่นยำมากขึ้น นอกจากนี้ยังมีการตัดฟีเจอร์ที่ปรากฏน้อยครั้งออกไปเพื่อช่วยลดจำนวนฟีเจอร์ และการนำฟีเจอร์สำคัญของข้อมูลทดสอบมาช่วยเลือกข้อมูลเรียนรู้เพื่อลดข้อมูลที่จะต้องพิจารณาอีกด้วย ผลการประเมินประสิทธิภาพแสดงให้เห็นว่าวิธีที่นำเสนออยู่อันดับที่ 4 มีค่า LBMaF เท่ากับ 25.70% เมื่อทดสอบบนข้อมูลวิกิพีเดียขนาดกลาง และอยู่อันดับที่ 2 มีค่า LBMaF เท่ากับ 23.48% เมื่อทดสอบบนข้อมูลวิกิพีเดียขนาดใหญ่

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2559

5670192221 : MAJOR COMPUTER ENGINEERING
KEYWORDS: HIERARCHICAL MULTI-LABEL CLASSIFICATION / LARGE-SCALE DATA / TEXT CLASSIFICATION / K-NEAREST NEIGHBOR

NATCHANON PHACHONGKITPHIPHAT: LARGE-SCALE HIERARCHICAL MULTI-LABEL TEXT CLASSIFICATION USING FLAT APPROACH WITH SVM PRUNING STRATEGY. ADVISOR: PEERAPON VATEEKUL, Ph.D., 68 pp.

Hierarchical multi-label classification is a type of classification which combines two aspects of problems; an instance may belong to more than one class, and these classes are organized into a hierarchical structure. Real world data are often complex like this. Hierarchical multi-label text classification is becoming ever more popular nowadays, because hierarchical structure can be applied to describe the relationship of textual data. Textual data which we have seen every day are web pages. As the size of web pages has been becoming extremely large, website such as Web directory and Wikipedia need the automated system to classify new web pages in their databases. This kind of problem is, therefore, a large-scale hierarchical multi-label classification. Many researches proposed various methods to deal with the problem, but these methods cannot process large-scale data. The methods may require a large storage space, may take too long to process or may have low accuracy. Meanwhile, some methods that can process large-scale data do not utilize the hierarchical structure at all.

This thesis proposed large-scale hierarchical multi-label text classification method that improved k-nearest neighbor method and utilized the hierarchical structure by trained SVM at the top level of hierarchy in order to increase the precision. Furthermore, we removed features that rarely appeared in training dataset to reduce large number of features, and used important features of test data to select training data in order to reduce large number of data. The evaluation showed that our proposed method ranked fourth on Wiki-Medium dataset with 25.70% LBMaF and ranked second on Wiki-Large dataset with 23.48% LBMaF.

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

กิตติกรรมประกาศ

งานวิจัยและวิทยานิพนธ์ฉบับนี้ไม่อาจเสร็จสมบูรณ์ได้ด้วยการทำงานของผู้วิจัยเพียงคนเดียว ความตั้งใจและความพยายามของผู้วิจัยยังมิอาจเทียบกับกำลังใจและการสนับสนุนจากบุคคลรอบข้างได้

ขอบพระคุณ อาจารย์ ดร.พีรพล เวทีกูล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ให้คำปรึกษาทั้งเรื่องงานวิจัย การทำงานและการดำเนินชีวิต ทั้งยังเชื่อมั่นและให้กำลังใจลูกศิษย์คนนี้อยู่เสมอ

ขอบพระคุณ ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล ผู้ให้เกียรติเป็นประธานกรรมการสอบวิทยานิพนธ์ และสละเวลาเพื่อให้คำปรึกษางานวิจัยทุกสัปดาห์

ขอบพระคุณ รองศาสตราจารย์ ดร.กฤษณะ ไวยมัย ผู้ให้เกียรติเป็นกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำเกี่ยวกับงานวิจัยและการเขียนเล่มวิทยานิพนธ์

ขอบพระคุณ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย สถานที่ที่เปรียบเสมือนบ้านหลังที่สอง ที่มอบทุนอุดหนุนการศึกษา "ทุนอัจฉริยะคืนรัง" ให้แก่ผู้วิจัย และขอบพระคุณ อาจารย์ประจำภาควิชาฯ ทุกท่านที่คอยเอาใจใส่ ไถ่ถามความคืบหน้าของงานวิจัย

ขอบพระคุณ ครูมกุฎ อรฤดี และอาจารย์วิกรัย จาระนัย ที่สอนให้คิดเพื่อผู้อื่น และสอนให้ทำงานทุกชิ้นอย่างประณีต

ขอบคุณสมาชิกทุกคนใน Data Mining Group และห้องปฏิบัติการอัจฉริยภาพเครื่องจักรและการค้นพบความรู้ (MIND Lab) โดยเฉพาะธีรวิทย์ พนิดาและเอกภพ

ขอบคุณทุกคนที่ช่วยเหลือและให้คำปรึกษาเรื่องต่างๆ แก่ผู้วิจัย

สุดท้ายนี้ ขอขอบพระคุณบิดามารดา และครอบครัวที่ให้กำลังใจ สนับสนุนให้ผู้วิจัยศึกษาต่อระดับปริญญามหาบัณฑิต และสนับสนุนผู้วิจัยทุกๆ ด้าน ขอบคุณครับ

สารบัญ

หน้า

หน้า

หน้า

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1. ที่มาและความสำคัญของปัญหา

การจำแนกประเภท (Categorization / Classification) [1, 2] เป็นกระบวนการการรู้จำ แยกความแตกต่าง และทำความเข้าใจสิ่งของ แนวความคิด หรือเหตุการณ์ต่างๆ เพื่อแบ่งสิ่งเหล่านั้นเป็นกลุ่ม การจำแนกประเภทเป็นพื้นฐานสำคัญที่ปรากฏในศาสตร์และศิลป์มากมาย เช่น คณิตศาสตร์ วิทยาศาสตร์ ภาษาศาสตร์ และสังคมศาสตร์ และแน่นอนว่ามีบทบาทสำคัญในวงการคอมพิวเตอร์เช่นกัน การจำแนกประเภททำได้โดยคนที่มีความรู้หรือผู้เชี่ยวชาญในศาสตร์นั้นๆ เริ่มแรกจึงนำคอมพิวเตอร์มาช่วยทำงานที่ต้องทำซ้ำๆ เท่านั้น แต่เมื่อปริมาณข้อมูลเริ่มมากและเพิ่มขึ้นเรื่อยๆ จนคนไม่อาจประมวลผลข้อมูลทั้งหมดได้ หรืออาจไม่เห็นรูปแบบที่ซ่อนอยู่ซึ่งนำมาช่วยจำแนกประเภทข้อมูลได้ คอมพิวเตอร์จึงยิ่งมีบทบาทสำคัญต่อการทำงานประเภทนี้

การจำแนกประเภทข้อมูลที่ง่ายที่สุดเริ่มจากการจำแนกประเภทระหว่างคลาส[1]สองคลาส จากนั้นจึงพัฒนาเป็นการจำแนกระหว่างหลายคลาส โดยยังคงตอบได้มากที่สุดเพียงคลาสเดียว ต่อมาจึงพัฒนาเป็นปัญหาที่ตอบได้หลายคลาสพร้อมกัน และซับซ้อนมากขึ้น เมื่อคลาสมีความสัมพันธ์กันเป็นโครงสร้างลำดับชั้น ซึ่งข้อมูลที่พบในชีวิตประจำวันนั้นจะอยู่ในรูปแบบเช่นนี้

การจำแนกประเภทขนาดใหญ่แบบหลายฉลากมีลำดับชั้น (Hierarchical Multi-Label Classification - HMC) เป็นการจำแนกประเภทที่มีลักษณะเฉพาะคือ ข้อมูลเรียนรู้ (Training Data) และข้อมูลทดสอบ (Test Data) แต่ละตัว อาจจัดอยู่ได้ในหลายคลาส และคลาสเหล่านี้มีความสัมพันธ์เป็นโครงสร้างลำดับชั้น (Class Hierarchy) คือ คลาสลำดับชั้นล่างมีความหมายเฉพาะเจาะจงกว่าคลาสลำดับชั้นบน ตัวอย่างการจำแนกประเภทแบบหลายฉลากมีลำดับชั้น เช่น

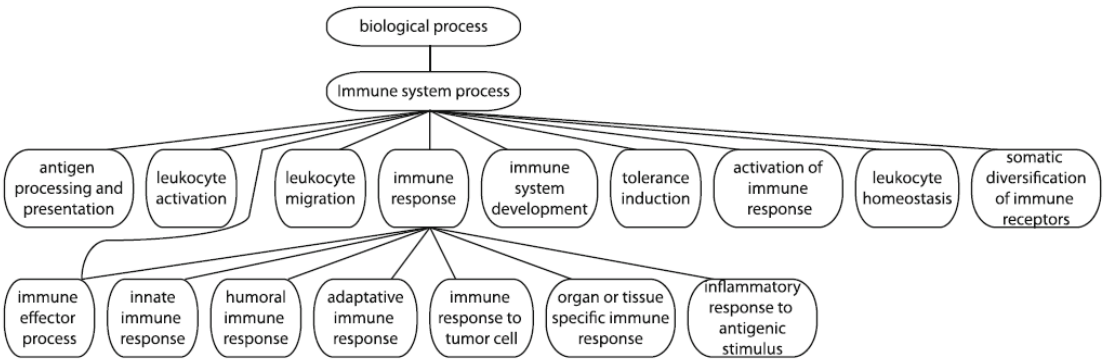
การจำแนกประเภทข้อความ (Text Categorization) ใน [3, 4] นำเสนอตัวอย่างที่แสดงว่าการจัดประเภทไฟล์เป็นโครงสร้างแบบลำดับชั้น ช่วยเพิ่มประสิทธิภาพของระบบค้นคืนสารสนเทศ (Information Retrieval Systems) ได้ โดยผู้เขียนยกตัวอย่างว่า ถ้าต้องการค้นคำว่า jaguar (เสือชนิดหนึ่ง) บนเว็บไซต์ ผู้ใช้อาจพบข้อมูลที่ต้องการได้ยาก เนื่องจากผลลัพธ์ส่วนใหญ่ที่ได้จากการค้นหา เป็นข้อมูลเกี่ยวกับรถยนต์ยี่ห้อ jaguar กรณีที่เป็นการค้นหาโดยใช้คำกำกวมหรือมีหลาย

---

[1] ผู้วิจัยจะใช้คำว่า คลาส แทนคำว่า ฉลาก และ ประเภท (label / class) เพื่อให้เข้าใจง่ายขึ้นเมื่อใช้คำว่า ประเภทร่วมกับคำว่า ประเภทที่เป็นส่วนประกอบของคำ เช่น การทำนายประเภทด้วยตัวจำแนกประเภท

ความหมายเช่นนี้ ถ้าผู้ใช้จำกัดขอบเขตการค้นหาให้อยู่ในโครงสร้างลำดับชั้นที่เป็นหมวดหมู่ที่ต้องการ ผู้ใช้จะพบคำตอบที่ต้องการได้ง่ายขึ้น เช่น โครงสร้างลำดับชั้นที่มีสัตว์ (Animal) เป็นโหนดราก[2,3] (Root Node)

การทำนายฟังก์ชันการทำงานของโปรตีน (Protein Function Prediction) ฟังก์ชันการทำงานของโปรตีน (Protein Function) มีรูปแบบตามธรรมชาติเป็นโครงสร้างลำดับชั้น เช่น Enzyme Commission [4, 5] และ Gene Ontology [4, 6] ใน Enzyme Commission แต่ละโหนดแทนเอนไซม์ (Enzyme) ชนิดต่างๆ ซึ่งต่างจากใน Gene Ontology ที่แต่ละโหนดแทนโปรตีนชนิดใดก็ได้ การทำนายฟังก์ชันการทำงานของโปรตีนนั้นสำคัญมาก เนื่องจากโรคหลายชนิดเกิดจากหรือเกี่ยวข้องกับโปรตีนที่ทำงานผิดปกติ ข้อมูลฟังก์ชันการทำงานของโปรตีนที่ถูกต้องจะช่วยให้นักวิจัยและแพทย์ วินิจฉัย สร้างตัวยา รวมถึงคิดค้นวิธีการรักษาโรคเหล่านั้นได้ ส่วนหนึ่งของโครงสร้าง Gene Ontology มีลักษณะดังรูปที่ 1

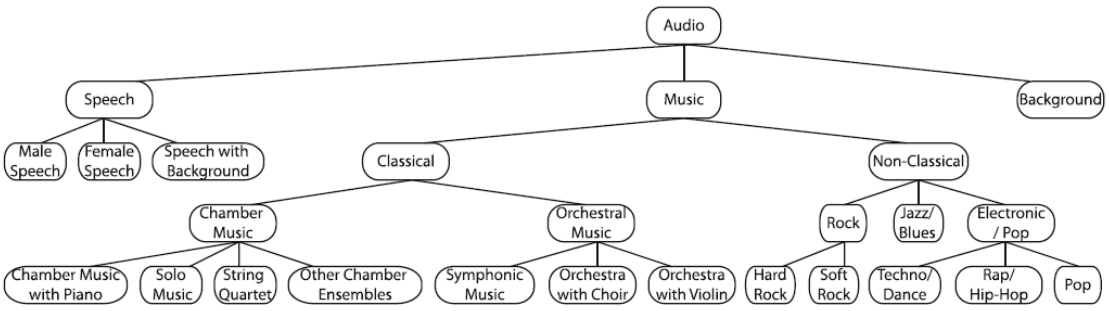


รูปที่ 1 โครงสร้างระดับบนของกระบวนการของระบบภูมิคุ้มกันโรคใน Gene Ontology
(อ้างอิงจาก Fig. 9 ใน [4])

การจำแนกแนวเพลง (Music Genre Classification) การใช้แนวเพลง (Genre) ในการค้นหาเพลงจากระบบฐานข้อมูลเพลงเป็นวิธีหนึ่งที่นิยมใช้กันมากที่สุด ทำให้แนวเพลงมีบทบาทสำคัญต่อการจัดการและค้นคืนข้อมูลเพลง ตัวอย่างการจำแนกประเภทเสียงเป็นลำดับชั้นแสดงได้ดังรูปที่ 2

[2] โหนด คือ คลาสที่อยู่ในโครงสร้างลำดับชั้น

[3] โหนดราก คือ โหนดที่อยู่ชั้นบนสุดในโครงสร้างลำดับชั้น



รูปที่ 2 การจำแนกประเภทเสียงเป็นลำดับชั้น

(อ้างอิงจาก Fig. 10 ใน [4])

การจำแนกประเภทข้อความแบบหลายฉลากมีลำดับชั้น เป็นหัวข้อที่นักวิจัยสนใจกันมากในปัจจุบัน เพราะโครงสร้างลำดับชั้นใช้อธิบายความสัมพันธ์ของข้อมูลประเภทข้อความได้ดี และมีการใช้งานมากขึ้นเรื่อยๆ โดยเฉพาะเว็บไซต์ เช่น เว็บไดเรกทอรี (Web directory) และวิกิพีเดีย (Wikipedia) [7] เว็บไซต์เหล่านี้มีการใช้งานอย่างแพร่หลายและเกือบจะตลอดเวลา จึงจำเป็นต้องมีระบบการจำแนกประเภทอย่างอัตโนมัติเมื่อมีเว็บใหม่เพิ่มเข้ามาในฐานข้อมูล งานวิจัยหลายงานนำเสนอวิธีแก้ปัญหาดังกล่าว เช่น [8, 9] แต่วิธีเหล่านั้นใช้กับการประมวลผลข้อมูลที่มีขนาดใหญ่ไม่ได้ เนื่องจากการประมวลผลอาจต้องใช้พื้นที่เก็บข้อมูลขนาดใหญ่มาก อาจใช้เวลาประมวลผลนานเกินไป จนนำผลลัพธ์มาใช้ประโยชน์ไม่ได้ หรือจำแนกประเภทได้ไม่แม่นยำ

ข้อมูลขนาดใหญ่ที่กล่าวถึงนี้มีขนาดใหญ่ทั้งในแง่ของจำนวนข้อมูล จำนวนคลาส และจำนวนฟีเจอร์ (Feature) ตัวอย่างข้อมูลขนาดใหญ่ เช่น ข้อมูลวิกิพีเดียจาก The Fourth Pascal Challenge on Large Scale Hierarchical Text Classification Challenge (LSHTC4) [10, 11] ที่มีจำนวนข้อมูลเรียนรู้มากกว่า 2,000,000 ชุด คลาสมากกว่า 300,000 คลาส และฟีเจอร์มากกว่า 1,600,000 ฟีเจอร์

งานวิจัยนี้จึงมุ่งเน้นที่จะพัฒนาการจำแนกข้อความขนาดใหญ่แบบหลายฉลากมีลำดับชั้นโดยปรับปรุงวิธีการแบบแฟลตและนำโครงสร้างลำดับชั้นมาช่วยกรองคำตอบให้ถูกต้องมากยิ่งขึ้น คลาสที่โหนดใบ[4] (Leaf Node) ที่มีโอกาสเป็นคำตอบหาได้จากการทำนายคลาสด้วยตัวจำแนกประเภท (Classifier) ที่สร้างโดยใช้ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine – SVM) ที่โหนดลำดับชั้นบน โดยตัดฟีเจอร์ที่ปรากฏน้อยครั้งออกจากชุดข้อมูลเรียนรู้ก่อนนำไปสร้างตัวจำแนกฯ เพื่อช่วยลดขนาดข้อมูลเรียนรู้ ทำให้โมเดลที่สร้างมีประสิทธิภาพดีขึ้น เมื่อใช้ทำนายคลาส จะช่วยลด

[4] โหนดใบ คือ โหนดในโครงสร้างลำดับชั้นที่ไม่มีโหนดลูก

จำนวนโหนดใบที่มีโอกาสเป็นคำตอบได้ จากนั้นทำนายคลาสคำตอบที่โหนดใบต่อด้วยการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด K อันดับ (k-Nearest Neighbors - k-NN) ที่ปรับปรุงแล้ว โดยเพิ่มการวัดความคล้ายคลึงเชิงมุมโคไซน์ (Cosine Similarity) ของข้อมูลทดสอบกับเซนทรอยด์เวกเตอร์ (Centroid Vector) ของแต่ละคลาสที่มีโอกาสเป็นคำตอบ สุดท้ายจะนำผลการทำนายคลาสมาพิจารณาตามข้อกำหนดเชิงลำดับชั้น (Hierarchical Constraint) จะได้เซตของคลาสที่เป็นคำตอบที่สมบูรณ์

1.2. วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาวิธีการจำแนกประเภทขนาดใหญ่แบบหลายฉลากมีลำดับชั้น ให้มีประสิทธิภาพดีกว่าวิธีการแบบแฟลต วิธี k-NN ที่ใช้เป็นเกณฑ์มาตรฐานของ LSHTC4 และอัลกอริทึมของผู้เข้าร่วม LSHTC4 ทั้งในด้านความแม่นยำของการทำนายคลาสและเวลาที่ใช้สร้างโมเดลทำนาย

1.3. ขอบเขตของการวิจัย

ข้อมูลที่นำมาใช้ทดสอบการจำแนกประเภทในการวิจัยนี้ คือ ข้อมูลวิกิพีเดียขนาดใหญ่ จาก LSHTC4 ประกอบด้วย ชุดข้อมูลเรียนรู้ (Training data set) ชุดข้อมูลทดสอบ (Test data set) และโครงสร้างลำดับชั้น ข้อมูลเรียนรู้และข้อมูลทดสอบแต่ละตัว อาจจัดอยู่ได้ในหลายคลาส คลาสเหล่านี้มีความสัมพันธ์ตามโครงสร้างลำดับชั้นที่กำหนดมาให้ โดยโครงสร้างฯ เป็นกราฟที่มีวัฏจักร (Cycle) ก่อนทำนายคลาสจึงต้องกำจัดวัฏจักรก่อน

เมื่อนิยามปัญหาตามเกณฑ์ในบทที่ 2 หัวข้อที่ 2.1.2 จะได้ว่า ข้อมูลวิกิพีเดียมีรูปแบบดังนี้

(1) มีโครงสร้างลำดับชั้นแบบกราฟ

(2) ข้อมูลแต่ละตัวจัดอยู่ได้ในหลายคลาส (Multiple paths of labels)

(3) ข้อมูลทดสอบทุกตัวต้องมีคลาสที่เป็นคำตอบอยู่ในทุกลำดับชั้นของโครงสร้างฯ (Full depth labeling)

และนิยามวิธีการที่นำเสนอตามเกณฑ์ในบทที่ 2 หัวข้อที่ 2.1.3 ได้ดังนี้

(1) อัลกอริทึมทำนายประเภทได้มากกว่าหนึ่งประเภท (Multiple path prediction)

(2) อัลกอริทึมจะทำนายประเภทที่โหนดใบ (Mandatory leaf-node prediction)

(3) อัลกอริทึมแก้ปัญหาที่มีโครงสร้างลำดับชั้นแบบกราฟได้

(4) เป็นการนำวิธีการแบบแฟลตมาประยุกต์ใช้

1.4. ประโยชน์ที่ได้รับ

ได้วิธีการจำแนกประเภทขนาดใหญ่แบบหลายฉลากมีลำดับชั้นที่มีประสิทธิภาพดีกว่าวิธีการแบบแฟลต วิธี K-NN ที่ใช้เป็นเกณฑ์มาตรฐานของ LSHTC4 และอัลกอริทึมของผู้เข้าร่วม LSHTC4 ทั้งในด้านความแม่นยำของการทำนายคลาสและเวลาที่ใช้สร้างโมเดลทำนาย

1.5. วิธีดำเนินการวิจัย

1. ศึกษาวรรณกรรมและงานวิจัยที่เกี่ยวข้องกับหัวข้อที่จะวิจัย
2. ศึกษาชุดข้อมูลจาก LSHTC ที่จะใช้ทดสอบ โดยพิจารณาว่ามีรูปแบบอย่างไร ประกอบด้วยข้อมูลประเภทใดบ้าง จะนำข้อมูลมาใช้งานได้อย่างไร เพื่อให้คิดแนวทางการวิจัยขั้นต่อไปได้
3. จำลองการทำงานของงานวิจัยที่เกี่ยวข้องเพื่อใช้เป็นมาตรฐานเปรียบเทียบในการประเมินประสิทธิภาพการทำงาน
4. ทดลองแนวทางวิจัยเบื้องต้น ปรับปรุงและประยุกต์การจำแนกประเภทด้วยวิธีที่ศึกษาจากงานวิจัยที่เกี่ยวข้องเพื่อให้เกิดวิธีใหม่
5. วิเคราะห์ผลการทดลองว่าการจำแนกประเภทแต่ละวิธีมีประสิทธิภาพต่างกันอย่างไร เพราะเหตุใด และแต่ละวิธีมีข้อดีข้อเสียหรือไม่ อย่างไร จากนั้นจึงสรุปผลการทดลอง
6. นำสรุปผลการทดลองมาปรับปรุงวิธีการจำแนกประเภทและทำการทดลองซ้ำเพื่อดูผลลัพธ์ที่เปลี่ยนแปลงไป
7. วิเคราะห์และสรุปผลการทดลองขั้นสุดท้าย
8. จัดทำวิทยานิพนธ์

1.6. โครงสร้างเนื้อหาในวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 5 บท คือ บทที่ 1 เป็นบทนำ บทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง บทที่ 3 นำเสนอแนวคิดและวิธีการจำแนกประเภทขนาดใหญ่แบบหลายฉลากมีลำดับชั้นที่ผู้วิจัยพัฒนา อธิบายรายละเอียดตั้งแต่ขั้นตอนการเตรียมข้อมูลจนถึงการรวมผลการทำนายคลาสเป็นคำตอบที่สมบูรณ์ บทที่ 4 กล่าวถึงชุดข้อมูลที่ใช้ในงานวิจัย รวมถึงการทดลองและวิเคราะห์ผล และบทที่ 5 บทสุดท้าย กล่าวถึงบทสรุปและข้อจำกัดของงานวิจัย รวมทั้งแนวทางวิจัยในอนาคตและผลงานตีพิมพ์จากวิทยานิพนธ์

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1. ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องกับงานวิจัยชิ้นนี้ ได้แก่ การจำแนกประเภท การแบ่งประเภทของปัญหาการจำแนกประเภทแบบมีลำดับชั้น การแบ่งประเภทของวิธีแก้ปัญหาการจำแนกประเภทแบบมีลำดับชั้น ประเภทของอัลกอริทึมที่ใช้แก้ปัญหาการจำแนกประเภทแบบมีลำดับชั้น การกำหนดชุดข้อมูลเรียนรู้ การวัดประสิทธิภาพการทำงาน วิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด k อันดับ และวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด 5 อันดับที่ใช้เป็นเกณฑ์มาตรฐานใน LSHTC

2.1.1. การจำแนกประเภท (Classification)

การจำแนกประเภทแบ่งออกเป็น 5 กลุ่ม ได้แก่ การจำแนกสองประเภท การจำแนกหลายประเภท การจำแนกประเภทแบบหลายฉลาก การจำแนกประเภทแบบมีลำดับชั้น และการจำแนกประเภทแบบหลายฉลากมีลำดับชั้น ซึ่งแต่ละกลุ่มมีรายละเอียดดังนี้

2.1.1.1. *การจำแนกสองประเภท (Binary Classification)*

การจำแนกสองประเภทเป็นรูปแบบที่ง่ายที่สุด คลาสทั้งหมดที่เป็นคำตอบได้มีเพียงสองคลาส และข้อมูลจะถูกจำแนกให้อยู่ในคลาสใดคลาสหนึ่งเท่านั้น เช่น การจำแนกเพศคนคนหนึ่งเป็นเพศชายหรือเพศหญิง เพศใดเพศหนึ่งเท่านั้น การจำแนกอีเมลว่าเป็นอีเมลก่อกวนหรือไม่ เป็นต้น

2.1.1.2. *การจำแนกหลายประเภท (Multiclass Classification)*

การจำแนกรูปแบบนี้ คลาสทั้งหมดที่เป็นคำตอบได้มีมากกว่าสองคลาส แต่ข้อมูลจะถูกจำแนกให้อยู่ในคลาสใดคลาสหนึ่งเท่านั้น เช่น การจำแนกกรุ๊ปเลือด คนคนหนึ่งมีกรุ๊ปเลือด A B O หรือ AB กรุ๊ปใดกรุ๊ปหนึ่งเท่านั้น การให้เกรดนักเรียนตามคะแนนที่ได้โดยมีเกณฑ์ที่แน่นอน นักเรียนจะได้เกรดใดเกรดหนึ่งเท่านั้น เป็นต้น

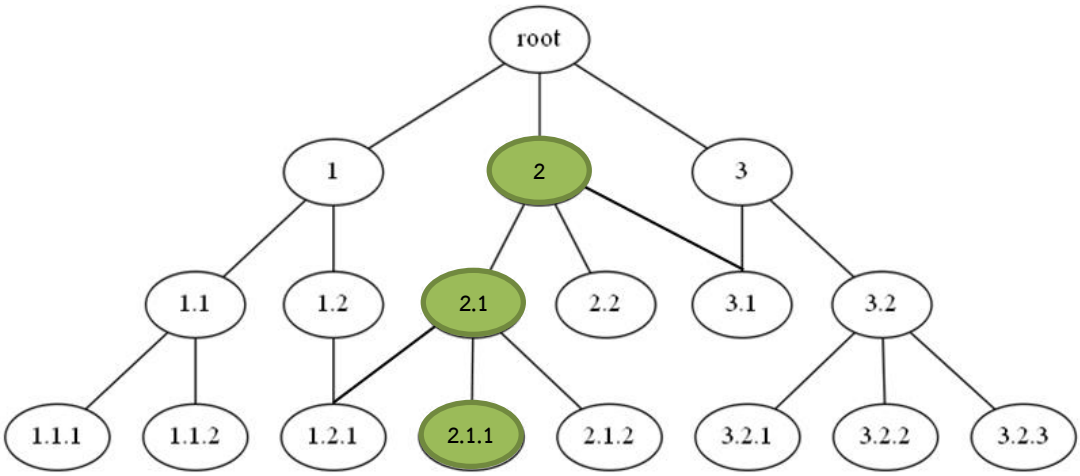
2.1.1.3. *การจำแนกประเภทแบบหลายฉลาก (Multi-label Classification)*

การจำแนกรูปแบบนี้ คลาสทั้งหมดที่เป็นคำตอบได้มีมากกว่าสองคลาส และข้อมูลอาจถูกจำแนกให้อยู่ได้มากกว่าหนึ่งคลาส เช่น การจำแนกประเภทภาพยนตร์ ภาพยนตร์

เรื่องหนึ่งอาจจัดได้ในหลายประเภท ทั้งผจญภัย แอ็คชั่น ตลก การจำแนกประเภทรูปภาพ รูปภาพรูปหนึ่งอาจเป็นได้ทั้งรูปขาวดำ รูปทิวทัศน์ รูปวาด เป็นต้น

2.1.1.4. *การจำแนกประเภทแบบมีลำดับชั้น (Hierarchical Classification - HC)*

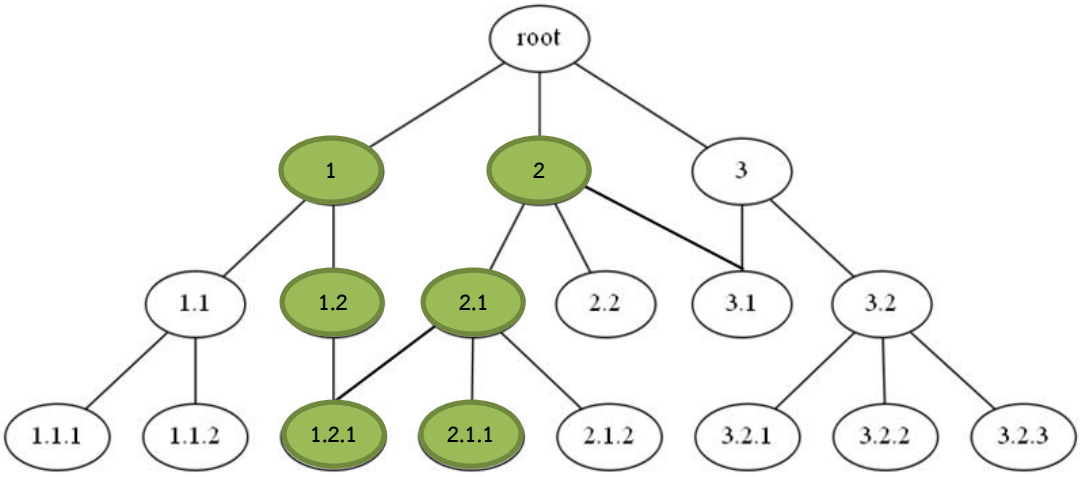
การจำแนกรูปแบบนี้ คลาสจะมีความสัมพันธ์เป็นลำดับชั้น คือ คลาสลำดับชั้นล่างมีความหมายเฉพาะเจาะจงกว่าคลาสลำดับชั้นบน โดยมีข้อกำหนดเชิงลำดับชั้น (Hierarchical Constraint) ซึ่งกำหนดว่า ข้อมูลที่จัดอยู่ในโหนดใดๆ ในโครงสร้างลำดับชั้น จะจัดอยู่ในโหนดบรรพบุรุษ (Ancestor Node) ทั้งหมดของโหนดนั้นด้วยโดยอัตโนมัติ เช่น รูปที่ 3 ถ้าคลาสคำตอบของข้อมูลทดสอบคือ คลาส 2.1.1 แล้วข้อมูลนั้นจะจัดอยู่ในคลาส 2.1 และคลาส 2 ด้วยโดยอัตโนมัติ



รูปที่ 3 ตัวอย่างของข้อกำหนดเชิงลำดับชั้น

2.1.1.5. *การจำแนกประเภทแบบหลายฉลากมีลำดับชั้น (Hierarchical Multi-Label Classification - HMC)*

การจำแนกรูปแบบนี้มีความซับซ้อนมากขึ้นจากการรวมปัญหาการจำแนกสองรูปแบบเข้าด้วยกัน นอกจากจะมีข้อกำหนดเชิงลำดับชั้นแล้ว ข้อมูลอาจถูกจำแนกให้อยู่ได้มากกว่าหนึ่งคลาสด้วย ทำให้มีข้อกำหนดเพิ่มเติมคือ จะจัดว่าอยู่ในหลายคลาส ก็ต่อเมื่อพิจารณาที่ลำดับชั้นหนึ่งๆ ของโครงสร้างลำดับชั้นแล้ว มีคลาสที่เป็นคำตอบจากการทำนายมากกว่าหนึ่งคลาส เช่น ที่ความลึกชั้นที่ 4 ของโครงสร้างฯ คลาสคำตอบของข้อมูลทดสอบมี 2 คลาส ได้แก่ คลาส 1.2.1 และคลาส 2.1.1 ดังรูปที่ 4



รูปที่ 4 ตัวอย่างข้อมูลที่จัดอยู่ได้หลายคลาสในการจำแนกประเภทแบบหลายฉลากมีลำดับชั้น

2.1.2. การแบ่งประเภทของปัญหาการจำแนกประเภทแบบมีลำดับชั้น

ประเภทของปัญหา HC [4] แบ่งได้โดยพิจารณา 3 หัวข้อต่อไปนี้

1. ประเภทของโครงสร้างลำดับชั้น

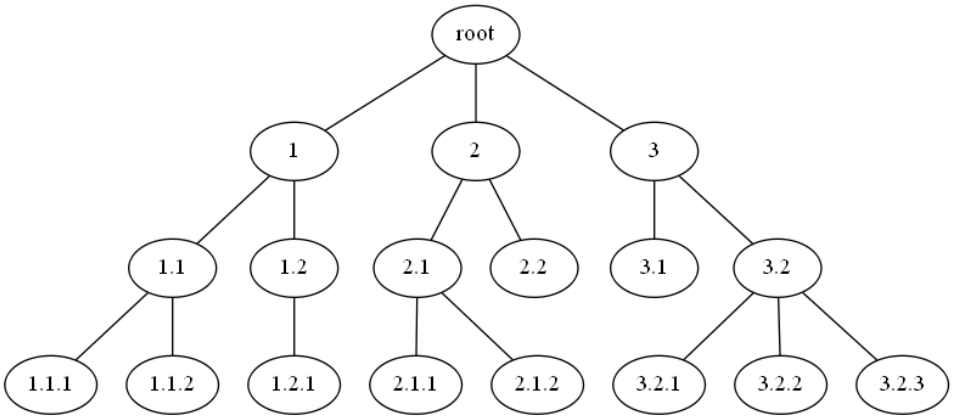
   1.1. Tree คลาสมีความสัมพันธ์เป็นโครงสร้างแบบต้นไม้ (Tree) คือ โหนดลูก (Child Node หรือ Child Class) มีโหนดแม่ (Parent Node หรือ Parent Class) ได้เพียงโหนดเดียว ดังรูปที่ 5

   1.2. Directed Acyclic Graph (DAG) คลาสมีความสัมพันธ์เป็นโครงสร้างแบบกราฟอวัฏจักรระบุทิศทาง (DAG) หรือโครงสร้างแบบกราฟ ซึ่งโหนดลูกมีโหนดแม่ได้มากกว่าหนึ่งโหนด ดังรูปที่ 6

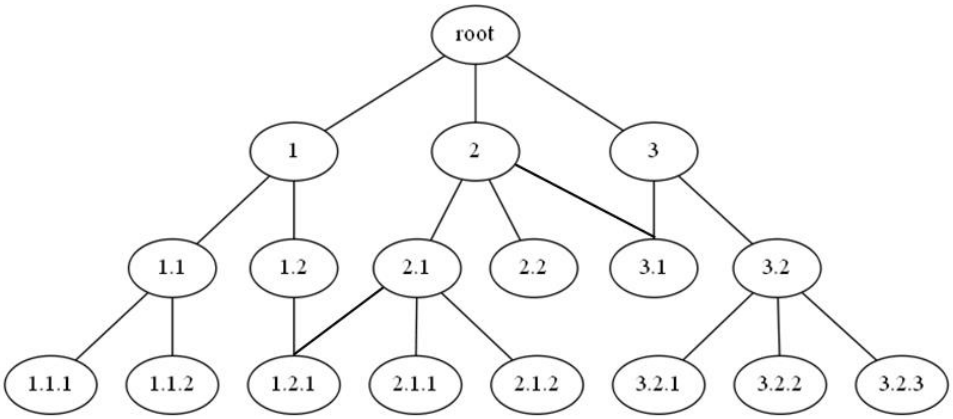
2. จำนวนคลาสคำตอบของข้อมูลตัวอย่าง

   2.1. Single path of labels (SPL) ข้อมูลจัดอยู่ในคลาสเดียวเท่านั้น

   2.2. Multiple paths of labels (MPL) ข้อมูลจัดอยู่ได้ในหลายคลาส ทำให้ปัญหาการจำแนกนี้เป็นแบบหลายฉลากมีลำดับชั้น



รูปที่ 5 ตัวอย่างโครงสร้างลำดับชั้นแบบต้นไม้



รูปที่ 6 ตัวอย่างโครงสร้างลำดับชั้นแบบกราฟ

3. ระดับความลึกของคลาสในโครงสร้างลำดับชั้นที่ข้อมูลตัวอย่างจัดอยู่ได้

   3.1. Full depth labeling (FD) ข้อมูลทุกตัวต้องตอบคลาสที่โหนดใบได้ นั่นคือ คลาสที่เป็นคำตอบอยู่ในทุกลำดับชั้น ตั้งแต่ชั้นบนสุดจนถึงชั้นของโหนดใบ

   3.2. Partial depth labeling (PD) ข้อมูลอย่างน้อยหนึ่งตัวตอบคลาสที่โหนดใบไม่ได้ ตอบได้เพียงโหนดภายใน (Internal Node)

2.1.3. การแบ่งประเภทของวิธีแก้ปัญหาการจำแนกประเภทแบบมีลำดับชั้น

ประเภทของอัลกอริทึมที่ใช้แก้ปัญหา HC [4] แบ่งได้โดยพิจารณา 4 หัวข้อต่อไปนี้

1. จำนวนคลาสที่อัลกอริทึมทำนายได้
   1.1. Single path prediction (SPP) อัลกอริทึมทำนายได้มากที่สุดเพียงหนึ่งคลาส
   1.2. Multiple path prediction (MPP) อัลกอริทึมทำนายได้มากกว่าหนึ่งคลาส
2. ระดับความลึกของคลาสในโครงสร้างลำดับชั้นที่อัลกอริทึมทำนายได้
   2.1. Mandatory leaf-node prediction (MLNP) อัลกอริทึมทำนายคลาสที่โหนดใบของโครงสร้างฯ เสมอ
   2.2. Non-mandatory leaf-node prediction (NMLNP) อัลกอริทึมทำนายคลาสที่ชั้นใดของโครงสร้างฯ ก็ได้
3. ประเภทของโครงสร้างแบบลำดับชั้นที่ใช้อัลกอริทึมแก้ปัญหาได้
   3.1. Tree คลาสที่จะทำนายมีความสัมพันธ์เป็นโครงสร้างแบบต้นไม้
   3.2. DAG คลาสที่จะทำนายมีความสัมพันธ์เป็นโครงสร้างแบบกราฟ
4. ประเภทของอัลกอริทึมแบ่งตามหมวดหมู่ที่กำหนดใน [4]
   4.1. วิธีการแบบบิ๊กแบง (Big-bang approach)
   4.2. วิธีการแบบบนลงล่าง (Top-down approach)
   4.3. วิธีการแบบแฟลต (Flat approach)

2.1.4. ประเภทของอัลกอริทึมที่ใช้แก้ปัญหาการจำแนกประเภทแบบมีลำดับชั้น

การจำแนกประเภทแบบมีลำดับชั้น แบ่งวิธีการแก้ปัญหาออกเป็น 3 กลุ่ม ได้แก่ วิธีการแบบบิ๊กแบง วิธีการแบบบนลงล่าง และวิธีการแบบแฟลต โดยทั้งสามวิธีนำมาประยุกต์ใช้กับการจำแนกประเภทขนาดใหญ่แบบหลายฉลากมีลำดับชั้นได้

1. วิธีการแบบบิ๊กแบง (Big-bang Approach / Global Approach)

วิธีการแบบบิ๊กแบง [4] ฝึกตัวจำแนกประเภทด้วยคลาสทั้งโครงสร้างลำดับชั้นในคราวเดียว ข้อดีคือ มีตัวจำแนกประเภทแค่ตัวเดียว และโมเดลทำนายมีขนาดเล็กกว่าขนาดโดยรวมของโมเดลที่ได้จากวิธีการอื่น นอกจากนี้ยังมั่นใจได้ว่าความสัมพันธ์แบบลำดับชั้นของคลาสจะไม่ถูกละเลย การสร้างโมเดลทำนายด้วยวิธีการแบบบิ๊กแบงอาจใช้ตัวจำแนกประเภทที่ต่างกันไป เช่น SVM, K-NN และ Naïve Bayes (NB)

เมื่อนำวิธีการนี้มาใช้เพื่อจำแนกประเภทข้อมูลขนาดใหญ่ จำเป็นต้องคำนึงถึงเวลาทั้งหมดที่ใช้เพื่อฝึกตัวจำแนกประเภทด้วยโครงสร้างลำดับชั้นทั้งโครงสร้าง ซึ่งโดยปกติ ข้อมูลที่มีขนาดใหญ่จะมีโครงสร้างฯ ขนาดใหญ่ และใช้เวลาฝึกตัวจำแนกฯ นานตามไปด้วย ทำให้ต้องจัดการกับขนาดของโครงสร้างฯ ก่อนนำวิธีนี้ไปใช้ ข้อเสียที่เห็นได้ชัดของวิธีนี้คือ ถ้ามีคลาสใหม่ในโครงสร้างลำดับชั้นแม้เพียงคลาสเดียว จะต้องฝึกตัวจำแนกฯ ใหม่ด้วยโครงสร้างใหม่ทั้งโครงสร้างเท่านั้น

2. วิธีการแบบบนลงล่าง (Top-down Approach / Local Approach)

วิธีการแบบบนลงล่าง [4] ฝึกตัวจำแนกประเภทแยกตามแต่ละคลาสในโครงสร้างลำดับชั้น โดยแบ่งได้ 3 กรณีตามรูปแบบการฝึกตัวจำแนกประเภท คือ (1) การฝึกที่ทุกโหนดในโครงสร้างฯ (2) การฝึกที่ทุกโหนดแม่ และ (3) การฝึกที่ทุกลำดับชั้นความลึก จากนั้นในขั้นตอนการทำนายคลาส จะทำนายด้วยโมเดลที่ได้จากการฝึก ตั้งแต่ลำดับชั้นบนลงไปจนถึงโหนดใบตามโครงสร้างฯ

ข้อดีของวิธีการแบบบนลงล่าง คือ ไม่ต้องฝึกตัวจำแนกประเภทด้วยโครงสร้างฯ ทั้งโครงสร้าง และนำผลลัพธ์ของแต่ละโมเดลมาใช้จำกัดขอบเขตคลาสที่มีโอกาสเป็นคำตอบให้น้อยลงได้ โดยถ้าข้อมูลทดสอบถูกทำนายว่าไม่อยู่ในโหนดแม่แล้ว ก็ไม่จำเป็นต้องทำนายที่โหนดลูกต่อ แต่สิ่งนี้อาจส่งผลเสีย ถ้าโมเดลที่โหนดแม่ทำนายผิด ผลลัพธ์เหล่านั้นจะถูกส่งต่อลงมาเรื่อยๆ เรียกว่า ความผิดพลาดที่ถูกส่งต่อจากโหนดแม่สู่โหนดลูก (Error Propagation) นอกจากนี้อีกปัญหาหนึ่งที่ต้องพิจารณาเมื่อจะนำวิธีนี้ไปใช้ คือ ชุดข้อมูลเรียนรู้ไม่สมดุล (Imbalanced Training Sets) เนื่องจากชุดตัวอย่างลบที่จะใช้ฝึกตัวจำแนกประเภทมีมากกว่าชุดตัวอย่างบวกอยู่มาก (หรือในทางกลับกัน) ทำให้ความแม่นยำในการจำแนกประเภทลดลง เพราะโมเดลมีโอกาสตอบเอนเอียงไปด้านที่มีชุดตัวอย่างมากกว่า การสร้างโมเดลทำนายด้วยวิธีการแบบบนลงล่างอาจใช้ตัวจำแนกประเภทที่ต่างกันไป โดยที่นิยมใช้กันคือ SVM [12]

3. วิธีการแบบแฟลต (Flat Approach)

นอกจากสองวิธีข้างต้นแล้ว วิธีการแบบแฟลตก็เป็นอีกวิธีหนึ่งที่ง่ายและนิยมใช้เพื่อแก้ปัญหาการจำแนกประเภทแบบมีลำดับชั้น วิธีการนี้จะฝึกตัวจำแนกประเภทและทำนายคลาสของข้อมูลที่โหนดใบเท่านั้น เมื่อไม่ได้นำความสัมพันธ์แบบลำดับชั้นมาใช้ประสิทธิภาพการทำนายคลาสจึงไม่ดีนัก แต่ก็มีการประยุกต์ใช้วิธีการนี้ที่น่าสนใจใน [8] ซึ่งใช้การเลือกคลาส (Class Selection) ให้เหลือเฉพาะคลาสที่มีโอกาสเป็นคำตอบ แล้วนำคลาสเหล่านั้นไปอ้างอิงกับความสัมพันธ์แบบลำดับชั้น จากนั้นตัดโหนดลูกของโหนดเหล่านั้นออกไปทั้งหมด จะทำให้โหนดเหล่านั้นกลายเป็นโหนดใบแทน การฝึกตัวจำแนกประเภทจะใช้ชุดข้อมูลเรียนรู้จากแต่ละโหนดใบที่มีโอกาสเป็นคำตอบ รวมกับชุดข้อมูลเรียนรู้จากโหนดบรรพบุรุษของโหนดเหล่านั้น ตราบใดที่โหนดบรรพบุรุษไม่ได้เป็นบรรพบุรุษร่วมกันของโหนดใบที่พิจารณา จากนั้นก็ใช้วิธีการแบบแฟลตซึ่งจำแนกประเภทของข้อมูลที่โหนดใบเท่านั้น วิธีการนี้เป็นวิธีที่ผู้วิจัยสนใจและจะนำไปประยุกต์ใช้ต่อไป เนื่องจากวิธีนี้นำความสัมพันธ์แบบลำดับชั้นมาใช้ให้เกิดประโยชน์ร่วมกับวิธีการแบบแฟลต ทำให้ผลการทำนายคลาสแม่นยำมากขึ้น

2.1.5. การกำหนดชุดข้อมูลเรียนรู้ (Training Policy)

ชุดข้อมูลเรียนรู้สำหรับฝึกตัวจำแนกประเภท ประกอบด้วย ชุดตัวอย่างบวกและชุดตัวอย่างลบ การกำหนดชุดข้อมูลเรียนรู้ [4, 13, 14] ทำได้หลายวิธีดังนี้

กำหนดให้ C แทน เซตของคลาสทั้งหมดในโครงสร้างลำดับชั้น

c_i แทน คลาสใดๆ ใน C

T แทน เซตของชุดข้อมูลเรียนรู้ทั้งหมด

$T^+(c_i)$ แทน เซตของชุดตัวอย่างบวกของคลาส c_i

$T^-(c_i)$ แทน เซตของชุดตัวอย่างลบของคลาส c_i

$Parent(c_i)$ แทน เซตของโหนดแม่ของคลาส c_i

$Child(c_i)$ แทน เซตของโหนดลูกของคลาส c_i

$A(c_i)$ แทน เซตของโหนดบรรพบุรุษของคลาส c_i

$D(c_i)$ แทน เซตของโหนดลูกหลาน (Descendant Node) ของคลาส c_i

$S(c_i)$ แทน เซตของโหนดพี่น้อง (Sibling Node) ของคลาส c_i

$*(c_i)$ แทน ข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุด คือ คลาส c_i

1. Exclusive policy

$$T^+(c_i) = *(c_i)$$
$$T^-(c_i) = T *(c_i) \quad (1)$$

วิธีนี้ชุดตัวอย่างบวกของคลาส c_i จะมีเพียงข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่คลาส c_i เท่านั้น และชุดตัวอย่างลบคือ ข้อมูลเรียนรู้ที่เหลือทั้งหมด การกำหนดชุดข้อมูลเรียนรู้วิธีนี้ทำให้เกิดปัญหาสามข้อ คือ

(1) ไม่ได้นำโครงสร้างลำดับชั้นมาพิจารณาในการกำหนดชุดข้อมูลเรียนรู้

(2) วิธีนี้ใช้งานได้กับปัญหาที่ตอบ Partial depth labeling ได้เท่านั้น

(3) ชุดตัวอย่างลบจะรวมเซตของโหนดลูกหลานของคลาส c_i ด้วย ซึ่งจะขัดข้อกำหนดเชิงลำดับชั้น

2. Less exclusive policy

$$T^+(c_i) = *(c_i)$$
$$T^-(c_i) = T \setminus *(c_i) \cup D(c_i) \quad (2)$$

วิธีนี้ชุดตัวอย่างบวกของคลาส c_i จะมีเพียงข้อมูลเรียนรู้ที่มีประเภทที่เฉพาะเจาะจงที่สุดที่คลาส c_i เท่านั้น และชุดตัวอย่างลบคือ ข้อมูลเรียนรู้ที่เหลือยกเว้นข้อมูลเรียนรู้ที่มีประเภทที่เฉพาะเจาะจงที่สุดที่คลาส c_i และโหนดลูกหลานของ c_i วิธีนี้จะไม่เกิดปัญหาข้อ (1) และ (3) เหมือน Exclusive policy แต่ยังคงใช้กับปัญหาที่ตอบ Partial depth labeling ได้เท่านั้น

3. Less inclusive policy หรือ ALL policy [13] หรือ Exclusive All Training policy (EAT) [14]

$$T^+(c_i) = *(c_i) \cup D(c_i)$$
$$T^-(c_i) = T \setminus *(c_i) \cup D(c_i) \quad (3)$$

วิธีนี้ชุดตัวอย่างบวกของคลาส c_i ประกอบด้วยข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่คลาส c_i และโหนดลูกหลานของ c_i และชุดตัวอย่างลบคือ ข้อมูลเรียนรู้ที่เหลือทั้งหมดที่ไม่ใช่ชุดตัวอย่างบวก

4. Inclusive policy

$$T^+(c_i) = *(c_i) \cup D(c_i)$$
$$T^-(c_i) = T \setminus *(c_i) \cup D(c_i) \cup A(c_i) \quad (4)$$

วิธีนี้ชุดตัวอย่างบวกของคลาส c_i ประกอบด้วยข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่คลาส c_i และโหนดลูกหลานของ c_i เหมือนกับ Less inclusive policy แต่ต่างกันที่ชุดตัวอย่างลบประกอบด้วยข้อมูลเรียนรู้ที่เหลือทั้งหมดที่ไม่ใช่ชุดตัวอย่างบวกและไม่ใช่ข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่โหนดบรรพบุรุษของคลาส c_i

5. Siblings policy [13] หรือ Exclusive Parent Training policy (EPT) [14]

$$T^+(c_i) = *(c_i) \cup D(c_i)$$
$$T^-(c_i) = S(c_i) \cup D(S(c_i)) \quad (5)$$

วิธีนี้ชุดตัวอย่างบวกของคลาส c_i ประกอบด้วยข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่คลาส c_i และโหนดลูกหลานของ c_i และชุดตัวอย่างลบคือข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่โหนดพี่น้องของคลาส c_i และโหนดลูกหลานของคลาสเหล่านั้น

6. Exclusive siblings policy

$$T^+(c_i) = *(c_i)$$
$$T^-(c_i) = S(c_i) \quad (6)$$

วิธีนี้ชุดตัวอย่างบวกของคลาส c_i จะมีเพียงข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่คลาส c_i เท่านั้น และชุดตัวอย่างลบคือข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่โหนดพี่น้องของคลาส c_i

2.1.6. การวัดประสิทธิภาพการทำงาน (Performance Evaluation)

ในการวัดประสิทธิภาพการทำงานจะใช้ตัววัดประสิทธิภาพการทำงาน เพื่อบอกถึงความแม่นยำของการทำนายคลาส สำหรับการจำแนกประเภทแบบต่างๆ ได้มีการนิยาม True Positives (TP), True Negatives (TN), False Positives (FP) และ False Negatives (FN) ซึ่งเป็นการเปรียบเทียบผลลัพธ์จากการทำนายกับผลลัพธ์ที่ถูกต้อง ดังตารางที่ 1

ตารางที่ 1 นิยามของ True Positive, True Negative, False Positive และ False Negative (อ้างอิงจาก [15])

	Actual class (Observation)	
Predicted class (Expectation)	TP (True Positive) Correct result	FP (False Positive) Unexpected result
	FN (False Negative) Missing Result	TN (True Negative) Correct absence of result

Positive และ Negative ในนิยาม คือ ผลลัพธ์การจำแนกคลาสว่าอยู่ในคลาส Positive หรือ Negative ส่วน True และ False ในนิยามบ่งบอกว่า ผลลัพธ์การจำแนกคลาสตรงกับผลลัพธ์ที่ถูกต้องหรือไม่

1. ตัววัดประสิทธิภาพการจำแนกข้อมูลสองประเภท (Binary Classification Performance Measures)

จากนิยามของ TP, TN, FP และ FN ข้างต้น นำมาคำนวณค่า $Precision\ (Pr), Recall\ (Re), F_\beta measure$ และ $Accuracy\ (Acc)$ ได้ดังนี้

$$\mathrm{Pr} = \frac{TP}{TP\ +\ FP} \qquad (7)$$

$$Re = \frac{TP}{TP + FN} \qquad (8)$$

$$F_\beta = \frac{(\beta^2 + 1) \times Pr \times Re}{\beta^2 \times Pr + Re} \quad (9)$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

โดยทั่วไป นิยมกำหนด ค่า β เท่ากับ 1 จะได้

$$F_1 = \frac{2 \times Pr \times Re}{Pr + Re} \quad (11)$$

2. ตัววัดประสิทธิภาพการจำแนกประเภทแบบหลายฉลาก (Multi-label Classification Performance Measures)

ตัววัดประสิทธิภาพการจำแนกประเภทแบบหลายฉลาก เกิดจากการใช้ตัววัดประสิทธิภาพ (7), (8) และ (11) ประกอบกันเพื่อวัดประสิทธิภาพโดยเฉลี่ยของการจำแนกประเภทแบบหลายฉลาก โดยหาค่าเฉลี่ยได้สองวิธี คือ $Macro\text{-}average$ และ $Micro\text{-}average$ ตัววัดประสิทธิภาพชนิดนี้เรียกอีกชื่อหนึ่งว่า $Label\text{-}based\ measures\ (Lb)$ [16] เนื่องจากเป็นการประเมินประสิทธิภาพของแต่ละคลาส

2.1. วิธี $Macro\text{-}average\ (Ma)$

คำนวณค่า $Precision\ (Pr), Recall\ (Re)$ และ F_1 ของแต่ละคลาสก่อนแล้วนำมาหาค่าเฉลี่ย จะได้ค่า $Label\text{-}based\ Macro\ Precision\ (LbMaPr)$ $Label\text{-}based\ Macro\ Recall\ (LbMaRe)$ และ $Label\text{-}based\ Macro\ F_1\ (LbF_1)$ ดังสมการที่ (12) (13) และ (14)

$$LBMaPr_i = \frac{1}{|C|}\sum_{i=1}^{|C|} Pr_i \quad (12)$$

$$LBMaRe_i = \frac{1}{|C|}\sum_{i=1}^{|C|} Re_i \quad (13)$$

$$LBMaF_{1,i} = \frac{1}{|C|}\sum_{i=1}^{|C|} F_{1,i} \quad (14)$$

2.2. วิธี $Micro\text{-}average\ (Mi)$

คำนวณค่า Pr และ Re โดยรวมค่า TP, FP และ FN ของทุกคลาสก่อน จากนั้นจึงคำนวณค่า $Micro\text{-}F_1$ ดังสมการที่ (15) (16) และ (17)

$$LBMiPr_i = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} \qquad (15)$$

$$LBMiRe_i = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)} \qquad (16)$$

$$LBMiF_{1,i} = \frac{2 \times LBMiPr \times LBMiRe}{LBMiPr + LBMiRe} \qquad (17)$$

3. ตัววัดประสิทธิภาพการจำแนกประเภทแบบมีลำดับชั้น (Hierarchical Classification Performance Measures)

3.1. Example-based measures (Eb) [17]

ตัววัดประสิทธิภาพการจำแนกประเภทแบบมีลำดับชั้น เป็นที่รู้จักในอีกชื่อหนึ่งว่า Example-based measures เนื่องจากเป็นการประเมินประสิทธิภาพการทำนายคลาสโดยเฉลี่ยของข้อมูลทดสอบทั้งหมด การจำแนกประเภทแบบมีลำดับชั้น มีคำตอบของข้อมูลทดสอบแต่ละตัวคือ คลาสใดๆ ที่ทำนายได้และโหนดบรรพบุรุษทั้งหมดของคลาสนั้น จึงมีการพัฒนา $Precision\ (Pr), Recall\ (Re)$ และ F_1 เพื่อให้วัดประสิทธิภาพการจำแนกประเภทรูปแบบนี้ได้

กำหนดให้ P_i แทน เซตของคลาสที่ทำนายได้ของข้อมูลทดสอบตัวที่ i T_i แทน เซตของคลาสที่ถูกต้องของข้อมูลทดสอบตัวที่ i $\hat{P}_i$ แทนเซตของ P_i รวมกับบรรพบุรุษทั้งหมดของคลาสในเซต P_i และ $\hat{T}_i$ แทนเซตของ T_i รวมกับบรรพบุรุษทั้งหมดของคลาสในเซต T_i จะคำนวณ $Example\text{-}based\ Precision\ (EbPr)$ $Example\text{-}based\ Recall\ (EbRe)$ และ $Example\text{-}based\ F_1\ (EbF_1)$ ได้ดังนี้

$$EbPr_i = \frac{|\hat{P}_i \cap \hat{T}_i|}{|\hat{P}_i|} \qquad (18)$$

$$EbRe_i = \frac{\left|\hat{P}_i \cap \hat{T}_i\right|}{\left|\hat{T}_i\right|} \quad (19)$$

$$EbF_{1,i} = \frac{2 \times EbPr_i \times EbRe_i}{EbPr_i + EbRe_i} \quad (20)$$

3.2. Example-label based evaluation (ELb) [14]

เมื่อต้องการวัดประสิทธิภาพการจำแนกประเภทแบบหลายฉลากมีลำดับชั้น อาจเลือกใช้ $Label\text{-}based\ measures\ (Lb)$ หรือ $Example\text{-}based\ measures\ (Eb)$ แต่ผลการวัดจะบอกได้เพียงประสิทธิภาพในด้านใดด้านหนึ่ง ตัวจำแนกประเภทที่ดีต้องมีประสิทธิภาพดีทั้งสองด้าน ใน [14] จึงได้เสนอ $Example\text{-}Label\ based\ measures\ (ELb)$ ตัววัดประสิทธิภาพที่รวมหลักการของ $Label\text{-}based$ และ $Example\text{-}based$ เข้าด้วยกันดังสมการที่ (21)

$$ELbFunc(Eb, Lb) = \frac{2 \times Eb \times Lb}{Eb + Lb} \quad (21)$$

เมื่อต้องการหาค่า $ELb\ Precision\ (ELbPr)$ $ELb\ Recall\ (ELbRe)$ หรือ $ELb\ F_1\ (ELbF_1)$ ให้แทนค่า Eb และ Lb ของค่านั้นเข้าไป เช่น ต้องการหาค่า $ELbPr$ แทนค่า $Example\text{-}based\ Precision\ (EbPr)$ และ $Label\text{-}based\ Precision\ (LbPr)$ ในสมการจะได้ผลลัพธ์ดังสมการที่ (22)

$$ELbPr(EbPr, LbPr) = \frac{2 \times EbPr \times LbPr}{EbPr + LbPr} \quad (22)$$

3.3. Multi-label Graph Induced Accuracy (MGIA) [18]

ตัววัดประสิทธิภาพนี้ มีการกำหนดค่าความผิดพลาดระหว่างคลาสที่ทำนายได้กับคลาสที่เป็นคำตอบที่ถูกต้องแต่ละคู่ ซึ่งส่วนใหญ่ค่าความผิดพลาดนี้จะใช้ค่าระยะทางระหว่างคลาสแต่ละคู่นั้นในโครงสร้างลำดับชั้น เมื่อรวมค่าความผิดพลาดทั้งหมดที่เกิดขึ้น จะสรุปได้ว่าการทำนายมีความผิดพลาดมากน้อยเพียงใด $MGIA$ จะนำค่าความผิดพลาดที่หาได้มาคำนวณตามสมการที่ (23) เพื่อหาความแม่นยำของการทำนาย $MGIA$ มีค่าอยู่ระหว่าง 0 ถึง 1 ยิ่งค่าเข้าใกล้ 1 มาก ความแม่นยำของการทำนายยิ่งมาก

$$MGIA = 1 - \frac{fnerror}{|P \cup T \backslash P \cap T| * D_{max}} \quad (23)$$

เมื่อ $fnerror$ คือ ผลรวมค่าความผิดพลาดของคลาสที่ทำนายกับคลาสที่เป็นคำตอบที่ถูกต้องแต่ละตัว

P คือ เซตของประเภทที่ทำนายได้

T คือ เซตของประเภทที่เป็นคำตอบที่ถูกต้อง

D_{max} คือ ค่าความผิดพลาดมากสุดที่ถูกกำหนดไว้ใช้เป็นค่าระยะทางระหว่างคลาสที่ทำนายได้กับคลาสที่เป็นคำตอบที่ถูกต้องโดยปริยาย (Default True Class) หรือระยะทางระหว่างคลาสที่เป็นคำตอบที่ถูกต้องกับคลาสที่ทำนายได้โดยปริยาย (Default Predicted Class)

3.4. Lowest Common Ancestor evaluation (LCA) [18]

ตัววัดประสิทธิภาพนี้มีพื้นฐานมาจาก $Example\text{-}based\ measures$ และมีการเพิ่มบรรพบุรุษของคลาสที่ทำนายได้ทั้งหมดกับคลาสที่เป็นคำตอบที่ถูกต้องเข้าไปด้วย แต่การเพิ่มบรรพบุรุษทั้งหมดเข้าไปทำให้ผลลัพธ์ของการวัดประสิทธิภาพไม่ดีนัก สำหรับคลาสที่มีบรรพบุรุษจำนวนมาก [18] จึงใช้แนวคิด LCA ตามทฤษฎีกราฟ [19] เพื่อเลือกบรรพบุรุษที่เหมาะสม แล้วนำเซตที่ได้มาคำนวณค่า $Lowest\ Common\ Ancestor\ Precision\ (Pr_{LCA})$ $Recall\ (Re_{LCA})$ และ $F_1\ (F_{1,LCA})$ โดย

กำหนดให้ Y แทน เซตของคลาสที่ถูกต้องของข้อมูลที่จะจำแนกประเภททุกตัว

$\hat{Y}$ แทน เซตของคลาสที่ทำนายได้ของข้อมูลที่จะจำแนกประเภททุกตัว

Y_{aug} แทน เซตของ Y รวมกับบรรพบุรุษที่เลือกตาม LCA

$\hat{Y}_{aug}$ แทน เซตของ $\hat{Y}$ รวมกับบรรพบุรุษที่เลือกตาม LCA

จะได้ว่า

$$Pr_{LCA} = \frac{|\hat{Y}_{aug} \cap Y_{aug}|}{|\hat{Y}_{aug}|} \quad (24)$$

$$Re_{LCA} = \frac{|\hat{Y}_{aug} \cap Y_{aug}|}{|Y_{aug}|} \quad (25)$$

$$F_{1,LCA} = \frac{2 \times Pr_{LCA} \times Re_{LCA}}{Pr_{LCA} + Re_{LCA}} \qquad (26)$$

2.1.7. วิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด k อันดับ (k-NN)

ขั้นตอนการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด k อันดับ มีดังนี้

1. กำหนดจำนวนข้อมูลเรียนรู้ k ตัวที่จะนำคลาสมาพิจารณา
2. คำนวณระยะทางระหว่างข้อมูลทดสอบแต่ละตัวกับข้อมูลเรียนรู้ทั้งหมด

   กำหนด X และ Y แทน ข้อมูลที่จะวัดระยะห่างระหว่างกัน

   x_i และ y_i แทน ฟีเจอร์ตัวที่ i ของ X และ Y ตามลำดับ

   การคำนวณระยะทางสำหรับข้อมูลที่ฟีเจอร์มีค่าต่อเนื่องมักใช้ Euclidean distance ดังสมการที่ (27) ส่วนการคำนวณระยะทางสำหรับข้อมูลที่ฟีเจอร์มีค่าไม่ต่อเนื่อง เช่น การจำแนกประเภทข้อความ มักใช้ Hamming distance ซึ่งคำนวณได้ดังสมการที่ (28)

$$distance(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (27)$$

$$distance(X,Y) = \sum_{i=1}^{n}|x_i - y_i| \qquad (28)$$

3. พิจารณาว่าข้อมูลเรียนรู้ที่มีระยะทางใกล้กับข้อมูลทดสอบมากที่สุด k อันดับแรก มีคลาสใดที่ปรากฏบ่อยที่สุด

   3.1. กรณีที่ตอบได้มากที่สุด 1 คำตอบ คลาสที่ปรากฏบ่อยที่สุดนั้นจะเป็นคำตอบ

   3.2. กรณีที่ตอบได้มากกว่า 1 คำตอบ จะเลือกตอบคลาสที่ปรากฏบ่อยจากมากไปน้อยตามจำนวนคำตอบที่ต้องการ

2.1.8. วิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด 5 อันดับที่ใช้เป็นเกณฑ์มาตรฐานใน LSHTC

ผู้จัดการแข่งขัน LSHTC ได้แนะนำวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด 5 อันดับ [20] ที่ใช้เป็นเกณฑ์มาตรฐานในการแข่งขันไว้ ดังนี้

กำหนดให้ T แทน เซตของข้อมูลเรียนรู้ทั้งหมด

1. คำนวณค่า idf (inverse document frequency) บนชุดข้อมูลเรียนรู้ แล้วนำมาแปลงค่าของฟีเจอร์ทุกตัวในข้อมูลเรียนรู้และข้อมูลทดสอบจาก tf (term frequency) เป็น $tfidf$
2. สำหรับข้อมูลทดสอบ t แต่ละตัว
   2.1. หาฟีเจอร์ที่มีค่า $tfidf$ มากที่สุด 3 อันดับแรกของข้อมูลทดสอบ t เรียกเซตของฟีเจอร์นี้ว่า $3F$
   2.2. สร้างเซต S ซึ่งเป็นเซตย่อยของเซต T โดยข้อมูลเรียนรู้แต่ละตัวใน S ต้องมีอย่างน้อย 1 ฟีเจอร์ที่ปรากฏในเซต $3F$
   2.3. หาข้อมูลเรียนรู้จากเซต S ที่มีระยะทางใกล้กับข้อมูลทดสอบ t มากที่สุด 5 อันดับแรก จากนั้นคืนค่าคลาสที่ปรากฏบ่อยที่สุด 3 อันดับแรกเป็นคำตอบของข้อมูลทดสอบ t
      2.3.1.กรณีที่มีน้อยกว่า 3 คลาส จะตอบเท่ากับคลาสที่มี
      2.3.2.กรณีที่มีคลาสที่ปรากฏบ่อยที่สุดเท่ากันมากกว่า 3 คลาส (tie breaking cases) จะตอบคลาสเหล่านั้นทั้งหมด

      คำนวณระยะทางระหว่างข้อมูล 2 ตัวได้โดยใช้สมการที่ (29)

$$distance = \frac{|D_1| + |D_2| - 2 * commonFeaturesOf(inst1, inst2)}{|D_1| + |D_2| - commonFeaturesOf(inst1, inst2)} \quad (29)$$

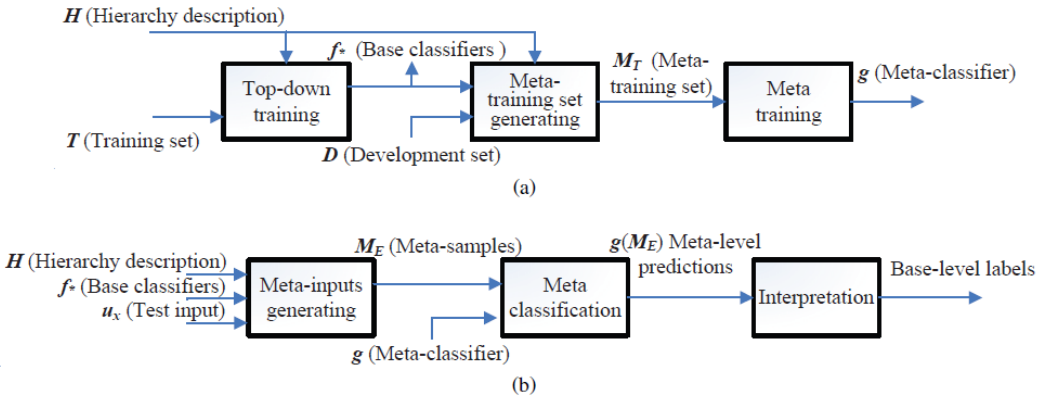
เมื่อ D_i คือ เซตของฟีเจอร์ทั้งหมดของข้อมูลตัวที่ i

$commonFeaturesOf(inst1, inst2)$ คือ จำนวนฟีเจอร์ร่วมของข้อมูลทั้ง 2 ตัว

2.2. งานวิจัยที่เกี่ยวข้อง

ขั้นตอนและวิธีจำแนกประเภทที่ผู้เข้าร่วมการแข่งขัน LSHTC นำเสนอแบ่งได้เป็น 2 กลุ่ม คือ วิธีการแบบบนลงล่าง และวิธีการแบบแฟลต

Xiao-Lin Wang และคณะ [21] นำเสนอการปรับปรุงอัลกอริทึมจำแนกประเภทแบบบนลงล่าง ชื่อ Meta-classification Top-down method (MetaTD) มีขั้นตอนการทำงานดังรูปที่ 7



รูปที่ 7 ขั้นตอนการทำงานของ Meta-classification Top-down method

(อ้างอิงจาก Figure 2 ใน [21])

ก่อนเริ่มขั้นตอนแรก งานวิจัยนี้จะสร้างชุดข้อมูลตรวจสอบ (Validation data set / Development set) เพื่อใช้ปรับค่าพารามิเตอร์ (Parameters) ของโมเดลก่อนนำไปใช้กับข้อมูลทดสอบ ในบทความนี้ไม่ได้ระบุชัดเจนว่าสร้างอย่างไร แต่โดยทั่วไปจะสร้างโดยแบ่งจากข้อมูลเรียนรู้

กำหนดให้ H แทน โครงสร้างลำดับชั้น

T แทน ชุดข้อมูลเรียนรู้

D แทน ชุดข้อมูลตรวจสอบ

E แทน ชุดข้อมูลทดสอบ

f_* แทน ตัวจำแนกประเภทขั้นพื้นฐาน (Base classifiers)

M_T แทน ชุดข้อมูลเรียนรู้ขั้น Meta (Meta-training set)

g แทน ตัวจำแนกประเภทขั้น Meta (Meta-classifier)

M_E แทน ข้อมูลทดสอบขั้น Meta (Meta-samples)

$g(M_E)$ แทน ผลการทำนายคลาสขั้น Meta (Meta-level predictions)

ขั้นตอนที่ (1) Top-down training เริ่มฝึกตัวจำแนกประเภท SVM^{light} [22] ที่ทุกโหนดในโครงสร้างฯ โดยกำหนดชุดข้อมูลเรียนรู้ด้วยวิธี Exclusive Parent Training policy เรียกโมเดลที่ได้ว่า f_* ตัวจำแนกประเภทขั้นพื้นฐาน

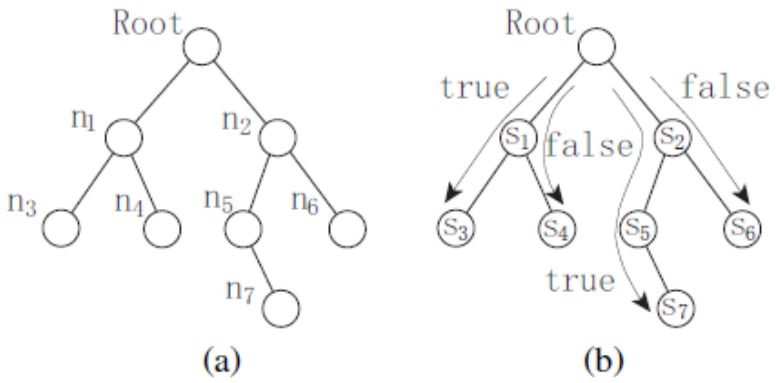
ขั้นตอนที่ (2) Meta-training set generating กำหนดค่า s เป็นค่า SVM scores[5] ขั้นต่ำ จากนั้นเริ่มทำนายคลาสชุดข้อมูลตรวจสอบด้วย f_* จากลำดับชั้นบนถึงลำดับชั้นล่าง จะได้ค่า SVM scores จากโมเดล โดยที่แต่ละโหนดจะเลือกเพียง r คลาสที่มีความมั่นใจมากที่สุดเพื่อทำนายคลาสที่ลำดับชั้นถัดไป เมื่อทำนายคลาสมาถึงโหนดใบแล้ว จะนำค่า SVM scores ตลอดเส้นทาง (Path) จากโหนดรากถึงโหนดใบมาสร้าง M_T ชุดข้อมูลเรียนรู้ขั้น Meta โดยฟีเจอร์ของขั้น Meta คือ แต่ละโหนดในเส้นทาง ค่าฟีเจอร์ คือ SVM scores ที่โหนดนั้น และคลาสคำตอบบ่งบอกว่าข้อมูลตรวจสอบนั้นเป็นสมาชิกของโหนดใบในเส้นทางนั้นใช่หรือไม่ นอกจากนี้ยังเพิ่มฟีเจอร์อีก 3 ฟีเจอร์ซึ่งคาดว่าจะช่วยให้ทำนายคลาสได้ถูกต้องมากขึ้น ฟีเจอร์ที่เพิ่มมา คือ

(i) average score มีค่าฟีเจอร์เท่ากับค่าเฉลี่ย SVM scores ตลอดเส้นทาง

(ii) minimum score มีค่าฟีเจอร์เท่ากับค่า SVM scores ต่ำสุดในเส้นทาง

(iii) pass-rate มีค่าฟีเจอร์เท่ากับสัดส่วนของจำนวน SVM scores ตลอดเส้นทางที่มีค่าเกินเกณฑ์ s ที่ตั้งไว้ ต่อจำนวนทั้งหมด

ตัวอย่างการทำงานขั้นตอนที่ (2) เมื่อข้อมูลตรวจสอบตัวหนึ่งมีคลาสคำตอบคือ n_3 และ n_7 และมีโครงสร้างลำดับชั้นดังรูปที่ 8 (a)



รูปที่ 8 ตัวอย่างการทำงานขั้นตอนที่ (2) Meta-training set generating
(a) โครงสร้างลำดับชั้น (b) เส้นทางการทำนายคลาสที่นำมาสร้างข้อมูลเรียนรู้ขั้น Meta
(อ้างอิงจาก Figure 3 ใน [21])

[5] SVM scores คือ ค่าที่ได้จาก SVM หลังการทำนายคลาส เป็นระยะห่างจากระนาบตัดสินใจแสดงถึงความมั่นใจการทำนายคลาสนั้น

รูปที่ 8 (b) แสดงค่า SVM scores s_i ที่แต่ละโหนด n_i เมื่อนำค่าเหล่านี้ไปสร้างข้อมูลเรียนรู้ขั้น Meta โดยกำหนดให้เลขโหนดในโครงสร้างฯ เป็นเลขฟีเจอร์ และฟีเจอร์ที่คำนวณเพิ่ม คือ ฟีเจอร์ 8 9 และ 10 จะได้ข้อมูลเรียนรู้ฯ 4 ตัวดังรูปที่ 9

No.	Basic			Extension		
1	1:s_1 [a]	3:s_3		8:a_{13} [b]	9:m_{13}	10:p_{13}
2	1:s_1	4:s_4		8:a_{14}	9:m_{14}	10:p_{14}
3	2:s_2	5:s_5	7:s_7	8:a_{257}	9:m_{257}	10:p_{257}
4	2:s_2	6:s_6		8:a_{26}	9:m_{26}	10:p_{26}

[a] dimension:value

[b] $a_{i_1 i_2 \ldots i_k}, m_{i_1 i_2 \ldots i_k}, p_{i_1 i_2 \ldots i_k}$ denote the average, minimum, and pass-rate of $s_{i_1}, s_{i_2} \ldots s_{i_k}$ respectively.

รูปที่ 9 ข้อมูลเรียนรู้ขั้น Meta ตัวที่ 1-4

ขั้นตอนที่ (3) Meta training ฝึก g ตัวจำแนกประเภทขั้น Meta ด้วย LIBLINEAR [23] และ M_T ชุดข้อมูลเรียนรู้ขั้น Meta

ขั้นตอนที่ (4) Meta-inputs generating ทำนายคลาสชุดข้อมูลทดสอบด้วย f_* ตัวจำแนกประเภทขั้นพื้นฐาน แล้วนำ SVM scores ที่ได้มาสร้าง M_E ข้อมูลทดสอบขั้น Meta ด้วยวิธีเดียวกับขั้นตอนที่ (2) ต่างกันเพียง เราจะไม่ทราบคลาสคำตอบ

ขั้นตอนที่ (5) Meta classification ทำนายคลาสของ M_E ข้อมูลทดสอบขั้น Meta ด้วย g จะได้ $g(M_E)$ ผลการทำนายคลาสขั้น Meta

ขั้นตอนที่ (6) Interpretation เลือกข้อมูลทดสอบขั้น Meta ที่มีค่า SVM scores มากกว่าเกณฑ์ที่ตั้งไว้มาแปลงเป็นโหนดใบเป็นคำตอบ การแปลงทำได้โดยการพิจารณาว่า ข้อมูลทดสอบขั้น Meta นั้นสร้างจากเส้นทางที่มีโหนดใดเป็นโหนดใบ

ชื่อในการแข่งขัน LSHTC ของผู้นำเสนอวิธีนี้คือ arthur ซึ่งได้ผลการประเมินประสิทธิภาพ LBMaF เป็นอันดับที่ 3 บนข้อมูลวิกิพีเดียขนาดกลาง แต่ไม่มีผลการประเมินบนข้อมูลวิกิพีเดียขนาดใหญ่ ตัวจำแนกประเภทขั้น Meta ของงานวิจัยนี้มีเพียงตัวเดียวและสร้างได้อย่างรวดเร็วเมื่อนำ LIBLINEAR มาใช้ แต่กระบวนการเตรียมชุดข้อมูลเรียนรู้ขั้น Meta นั้นใช้เวลานานมาก เพราะต้องฝึกตัวจำแนกประเภทที่ทุกโหนดทั้งโครงสร้างฯ และต้องทำนายชุดข้อมูลตรวจสอบเพื่อนำค่า SVM scores มาใช้ต่อ ทำให้วิธีการนี้ไม่อาจใช้กับข้อมูลที่มีโครงสร้างลำดับชั้นขนาดใหญ่ได้

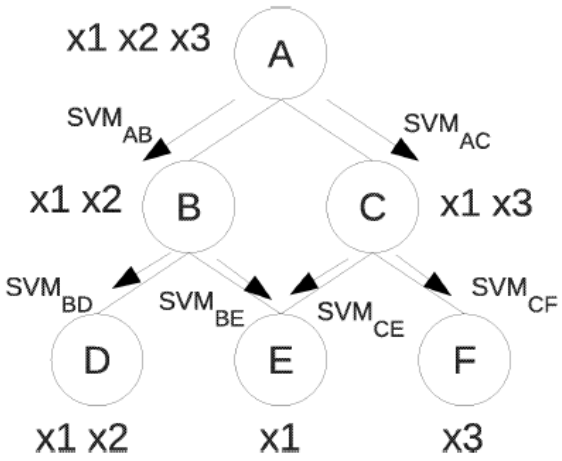
Yutaka Sasaki และ Davy Weissenbacher [12] นำเสนอการปรับปรุงอัลกอริทึมจำแนกประเภทแบบบนลงล่าง โดยวิธีที่นำเสนอประกอบด้วย 2 ขั้นตอนหลัก คือ (1) Training Stage และ (2) Classification Stage ก่อนเริ่มขั้นตอนแรก งานวิจัยนี้จะแปลงค่าฟีเจอร์ทั้งหมดในข้อมูลเรียนรู้และข้อมูลทดสอบด้วยสมการที่ (30) เพื่อหลีกเลี่ยงผลกระทบจากฟีเจอร์ที่มีค่ามาก

$$v_{new} = \frac{v}{v+1} \qquad (30)$$

เมื่อ v คือ ค่าฟีเจอร์เดิม และ v_{new} คือ ค่าฟีเจอร์ใหม่

ขั้นตอนที่ (1) Training Stage ทำการฝึกตัวจำแนกประเภทที่ทุกกิ่ง (Edge) ในโครงสร้างลำดับชั้น ซึ่งกิ่งเหล่านี้ คือ ความสัมพันธ์แบบแม่-ลูก (Parent-Child Relation) ของโหนดในโครงสร้างฯ โดยกำหนดชุดข้อมูลเรียนรู้ด้วยวิธี Exclusive Parent Training policy

ตัวอย่างการฝึกตัวจำแนก SVM ที่กิ่ง BD ซึ่งอยู่ระหว่างโหนดแม่ B และโหนดลูก D ดังรูปที่ 10 จะได้ว่า ชุดตัวอย่างบวก คือ ข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่โหนด D และชุดตัวอย่างลบ คือ ข้อมูลเรียนรู้ที่มีคลาสที่เฉพาะเจาะจงที่สุดที่โหนด E



รูปที่ 10 การฝึกตัวจำแนก SVM ที่ทุกกิ่ง
(อ้างอิงจาก Fig. 2 ใน [12])

ขั้นตอนที่ (2) Classification Stage กำหนดค่า bias β เป็นเกณฑ์ขั้นต่ำของ SVM score และค่า global threshold θ จากนั้นทำนายคลาสของข้อมูลทดสอบแต่ละตัวด้วยโมเดลที่แต่ละกิ่งจากบนลงล่างตามโครงสร้างลำดับชั้น โดยมีเงื่อนไขว่า ค่า SVM score ที่ได้จากโมเดลต้องมีค่า

มากกว่า β จึงจะทำนายต่อที่ชั้นถัดไป กรณีที่โหนดแม่ใดไม่มีโมเดลที่กิ่งที่มีค่า SVM score มากกว่าเกณฑ์แม้แต่โมเดลเดียว ให้เลือกทำนายต่อชั้นถัดไปที่โหนดลูกทางกิ่งที่มี SVM score สูงที่สุด

เมื่อทำนายคลาสจนถึงโหนดใบแล้ว แปลงค่า SVM score ในเส้นทางที่มีค่า SVM score ตั้งแต่กิ่งบนสุดจนถึงโหนดใบ ให้ค่าอยู่ในช่วง [0,1] โดยใช้ Sigmoid Function ตามสมการที่ (31)

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (31)$$

จากนั้นนำ SVM score มาคูณกันแยกตามแต่ละเส้นทางจนถึงโหนดใบ คลาสคำตอบของข้อมูลทดสอบ คือ โหนดใบแต่ละโหนดที่มีค่าผลคูณ SVM score มากกว่าหรือเท่ากับ θ

ชื่อในการแข่งขัน LSHTC ของผู้นำเสนอวิธีนี้คือ TTI ซึ่งได้ผลการประเมินประสิทธิภาพ LBMaF เป็นอันดับที่ 1 บนข้อมูลวิกิพีเดียขนาดกลาง แต่ไม่มีผลการประเมินบนข้อมูลวิกิพีเดียขนาดใหญ่ การกำหนดชุดข้อมูลเรียนรู้ที่ใช้ในงานวิจัยนี้ ช่วยลดปัญหาชุดข้อมูลเรียนรู้ไม่สมดุลได้ การกำหนดค่า β และ θ เป็นเกณฑ์ก็ช่วยกรองคลาสที่จะเป็นคำตอบได้เช่นกัน แต่จำนวนตัวจำแนกประเภทที่ต้องฝึกอาจมีมากเท่ากับจำนวนกิ่งในโครงสร้างลำดับชั้นก็เป็นได้ จึงยังไม่เหมาะที่จะใช้กับข้อมูลที่มีโครงสร้างฯ ใหญ่นัก

Xiaogang Han และคณะ [24] นำเสนอการปรับปรุงวิธีการ k-NN โดย

กำหนดให้ H แทน โครงสร้างลำดับชั้น

m แทน จำนวนโหนดใบ

C แทน เซตของโหนดใบ

n แทน จำนวนข้อมูลเรียนรู้

D แทน ชุดข้อมูลเรียนรู้

Q แทน ชุดข้อมูลทดสอบ

u แทน จำนวนฟีเจอร์

T แทน เซตของฟีเจอร์

P แทน เซตของโหนดแม่ทั้งหมดของโหนดใบ

$Parents(c)$ แทน เซตของโหนดแม่ของคลาส c

$Children(p)$ แทน เซตโหนดลูกของคลาส p

$D(c)$ แทน เซตข้อมูลเรียนรู้ของคลาส c

$Cat(d)$ แทน เซตคลาสของข้อมูลเรียนรู้ d

$T(d)$ แทน เซตฟีเจอร์ของข้อมูลเรียนรู้ d

ค่าความคล้ายระหว่างข้อมูลทดสอบกับข้อมูลเรียนรู้ที่ใช้ในงานวิจัยนี้มี 2 ค่า คือ ความคล้ายคลึงเชิงมุมโคไซน์ cosine similarity และ BM25 โดยมีวิธีการคำนวณดังนี้

ก่อนคำนวณค่า cosine similarity $cossim(d_i, d)$ จะแปลงค่าฟีเจอร์ข้อมูลเป็น $tfidf\ (\omega_{t,d})$ ด้วยสมการที่ (32) และ (33) ก่อน แล้วจึงคำนวณด้วยสมการที่ (34)

$$idf(t) = \log(\frac{n}{n_t}) \quad (32)$$

$$\omega_{t,d} = \log(tf(t,d) + 1) \cdot idf(t) \quad (33)$$

$$cossim(d_i, d) = \frac{d_i \cdot d}{\|d_i\| \cdot \|d\|} = \frac{\sum_{k=1}^{u} \omega_{k,i}\omega_k}{\sqrt{\sum_{k=1}^{u} {\omega_{k,i}}^2}\sqrt{\sum_{k=1}^{u} {\omega_k}^2}} \quad (34)$$

เมื่อ n_t คือ จำนวนข้อมูลเรียนรู้ที่มีฟีเจอร์ t

ส่วนค่า BM25 ต้องกำหนด parameters 2 ตัว คือ k_1 และ b แล้วคำนวณค่าตามสมการที่ (35) และ (36)

$$idf(t_j) = \log\left(\frac{n - n_t + 0.5}{n_t + 0.5}\right) \quad (35)$$

$$bm25sim(d, d_i) = \sum_{j=1}^{m} idf(t_j) \cdot \frac{tf(t_j, d_i) \cdot (k_1 + 1)}{tf(t_j, d_i) + k_1 \cdot \left(1 - b + b \cdot \frac{|d_i|}{avgdl_D}\right)} \cdot \frac{tf(t_j, d) \cdot (k_1 + 1)}{tf(t_j, d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{avgdl_Q}\right)} \quad (36)$$

เมื่อ $|d|$ และ $|d_i|$ คือ ความยาวของข้อมูลเรียนรู้ d

$avgdl_D$ และ $avgdl_Q$ คือ ความยาวเฉลี่ยของข้อมูลในชุดข้อมูลเรียนรู้ D และข้อมูลในชุดข้อมูลทดสอบ Q ตามลำดับ

ขั้นตอนที่ (1) สร้างเซต DS เก็บทูเพิล (Tuple) $\{(d_i, \vec{d}_i, c_j)|1 \leq i \leq n, c_j \in Cat(d_i)\}$ โดย $\vec{d}_i$ คือ เวกเตอร์ฟีเจอร์ของข้อมูลเรียนรู้ตัวที่ i (d_i) และ c_j คือ คลาสหนึ่งของ d_j

ต่อมาพิจารณาข้อมูลทดสอบ d แต่ละตัว แล้วทำขั้นตอนที่ (2) ถึง (4)

ขั้นตอนที่ (2) คำนวณค่าความคล้าย[6]ระหว่าง d กับข้อมูลเรียนรู้ทุกตัว เรียงลำดับข้อมูลเรียนรู้ด้วยค่าความคล้าย และสร้างเซต $KNN(d)$ ที่มีสมาชิกเป็นข้อมูลเรียนรู้ที่มีค่าความคล้ายมากที่สุด k อันดับแรก

ขั้นตอนที่ (3) สร้างเซต CD จากเซต $KNN(d)$ และเซต DS เพื่อเก็บทูเพิล $\{(d_i, \vec{d}_i, c_j)|d_i \in KNN(d)\}$ และสร้างเซตว่าง CS

ขั้นตอนที่ (4) พิจารณาแต่ละทูเพิลใน CD นำมาสร้างทูเพิล $(d_i, c_j, score_i)$ เก็บไว้ใน CS

ขั้นตอนที่ (5) รวมเซต CS ที่คำนวณค่า cosine similarity กับ BM25 เป็นเซตเดียว

ขั้นตอนที่ (6) กำหนดให้

$score(c)$ แทน เซตค่าความคล้ายของข้อมูลเรียนรู้ที่อยู่ในคลาส c

$pscores(c)$ แทน เซตค่าความคล้ายของโหนดแม่ของคลาส c คำนวณได้จากสมการที่ (37)

$$pscores(c) = scores\left(Children\left(Parents(c)\right)\right) = \{score_i|(d_i, c_j, score_i) \in CS, c_j \in Children(Parents(c))\} \quad (37)$$

ขั้นตอนนี้จะคำนวณฟีเจอร์ใหม่ 4 ฟีเจอร์ ดังนี้

(i) $\max(score(c))$ (ii) $\sum pscores(c)$

(iii) $\sum scores(c)$ (iv) $r(c) = \frac{|scores(c)|}{|D(c)|}$

จากนั้นคำนวณค่า $rs(c)$ $ranking\ score$ ด้วยสมการที่ (38) ให้ครบทุกคลาส c สุดท้ายจะเลือกคลาสที่มีค่า $rs(c)$ มากที่สุด M อันดับ เป็นคำตอบ

$$rs(c) = \theta_1 \log(\max(scores(c)) + \theta_2 \log\left(\sum pscores(c)\right) + \theta_3 \log\left(\sum scores(c)\right) + \theta_4 \log(r(c)) \quad (38)$$

[6] ค่าความคล้ายในขั้นตอนนี้ คือ cosine similarity หรือ BM25 ก็ได้

ชื่อในการแข่งขัน LSHTC ของผู้นำเสนอวิธีนี้คือ chrishan ซึ่งได้ผลการประเมินประสิทธิภาพ LBMaF เป็นอันดับที่ 5 ทั้งบนข้อมูลวิกพีเดียขนาดกลางและขนาดใหญ่ งานวิจัยนี้นำค่าความคล้ายที่ได้จากข้อมูลเรียนรู้ที่เป็นเพื่อนบ้านใกล้สุด k อันดับมาใช้จัดลำดับคลาสที่น่าจะเป็นคำตอบ แม้วิธีนี้จะใช้ทำนายคลาสข้อมูลขนาดใหญ่ได้ แต่พารามิเตอร์ที่ต้องปรับค่าก็มีจำนวนมาก จึงต้องทำกระบวนการทั้งหมดซ้ำหลายครั้ง เพื่อให้เลือกพารามิเตอร์ที่ดีที่สุดได้

Dong-Hyun Lee [25] นำเสนอ Multi-Stage Rocchio Classification (MSRC) โดย Rocchio Classification เป็นวิธีการจำแนกประเภทอย่างง่ายที่มีพื้นฐานที่การหาความคล้ายระหว่างข้อมูลทดสอบกับเซนทรอยด์ของคลาส แต่โดยปกติแล้ววิธีนี้ตอบได้แค่คลาสเดียว งานวิจัยนี้จึงเสนอวิธีการปรับปรุง Rocchio Classification ให้ทำงานหลายขั้นตอนเพื่อให้ทำนายคลาสได้มากขึ้น

ขั้นตอนที่ (1) ประยุกต์การวัดความคล้าย BM25 ที่คำนวณดังสมการที่ (39) และ (40) มาใช้แปลงค่า tf ของฟีเจอร์ จะได้สูตรการแปลงฟีเจอร์ข้อมูลเรียนรู้ดังสมการที่ (41) และสูตรการแปลงฟีเจอร์ข้อมูลทดสอบดังสมการ (42) โดยใช้ค่า $k = 1.5$ และ ค่า $b = 0.75$

$$IDF(q_i) = \log\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}\right) \quad (39)$$

$$score(D, Q) = \sum_{j=1}^{m} IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k+1)}{f(q_i, D) + k \cdot \left(1 - b + b \cdot \frac{L}{L_{avg}}\right)} \quad (40)$$

$$wtf_{t,d} = \frac{(k+1) \cdot tf_{t,d}}{tf_{t,d} + k \cdot \left(1 - b + b \cdot \frac{L_d}{L_{avg}}\right)} \cdot \log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \quad (41)$$

$$wtf_{t,d} = \frac{(k+1) \cdot tf_{t,d}}{tf_{t,d} + k \cdot \left(1 - b + b \cdot \frac{L_d}{L_{avg}}\right)} \quad (42)$$

ขั้นตอนที่ (2) คำนวณเซนทรอยด์เวกเตอร์ของแต่ละคลาสด้วยสมการ (43)

ขั้นตอนที่ (3) แปลงเซนทรอยด์เวกเตอร์ให้เป็นยูนิตเวกเตอร์[7]ด้วยสมการ (44) แล้วคำนวณ cosine similarity ระหว่างข้อมูลทดสอบกับเซนทรอยด์เวกเตอร์ของแต่ละคลาส ด้วยสมการ (45)

[7] ยูนิตเวกเตอร์ คือ เวกเตอร์ที่มีความยาวเท่ากับ 1

$$\overrightarrow{\mu_c} = \frac{1}{|D_c|} \sum_{d \in D_c} \overrightarrow{v_d} \quad (43)$$

$$\overrightarrow{\mu_c} = \frac{\sum_{d \in D_c} \overrightarrow{v_d}}{\left\| \sum_{d \in D_c} \overrightarrow{v_d} \right\|} \quad (44)$$

$$c^* = \underset{c}{\operatorname{argmin}} \, \overrightarrow{\mu_c} \cdot \overrightarrow{v_d} \quad (45)$$

ขั้นตอนที่ (4) เลือกคลาสที่มีค่าความคล้ายมากที่สุด K อันดับแรก นำแต่ละคลาสไปหา Label Power-set (LP) และทำขั้นตอนที่ (5) จนกว่าจะเสร็จจึงพิจารณาคลาสถัดไป

ขั้นตอนที่ (5) เพิ่มคลาสใน LP ทีละคลาส หาเซนทรอยด์เวกเตอร์ของ LP 2 รูปแบบ คือ (i) including centroid ข้อมูลเรียนรู้ที่นำมาคำนวณเซนทรอยด์มีเซตย่อยของคลาสคำตอบเป็น LP (ii) same centroid ข้อมูลเรียนรู้ที่จะนำมาคำนวณเซนทรอยด์มีคลาสคำตอบเป็น LP เท่านั้น

จากนั้นตรวจสอบเกณฑ์ดังสมการที่ (46) ทำซ้ำขั้นตอนที่ (5) จนกว่าค่าจะไม่ตรงตามเกณฑ์ สุดท้ายจะได้เซตคำตอบตามเซต LP นั้น

$$th_n \overrightarrow{v_d} \cdot \vec{\mu}_{including}(c_1, \dots, c_n) < \overrightarrow{v_d} \cdot \vec{\mu}_{same}(c_1, \dots, c_n) \quad (46)$$

งานวิจัยนี้ได้ผลการประเมินประสิทธิภาพ LBMaF เป็นอันดับที่ 2 บนข้อมูลวิกิพีเดียขนาดกลาง และได้อันดับที่ 1 บนข้อมูลวิกิพีเดียขนาดใหญ่ นับว่าเป็นผู้เข้าแข่งขันที่ได้ผลการประเมินโดยรวมที่ดีที่สุด วิธีการที่งานวิจัยนี้นำเสนอไม่ซับซ้อน แต่ได้ผลดีและมีการใช้ Label Power-set ที่เลือกรูปแบบเซตคำตอบที่เป็นไปได้บนชุดข้อมูลเรียนรู้มาเป็นคำตอบของชุดข้อมูลทดสอบ ซึ่งก็ตรงกับธรรมชาติของข้อมูลทั่วไปที่ข้อมูลทดสอบส่วนใหญ่มักจะตอบเหมือนหรือใกล้เคียงกับข้อมูลเรียนรู้ แต่ถ้าเซตคำตอบแต่ละเซตของข้อมูลเรียนรู้มีจำนวนสมาชิกมาก การใช้วิธีการนี้ก็อาจได้ประสิทธิภาพที่ไม่ดีนัก

บทที่ 3
การจำแนกข้อความขนาดใหญ่แบบหลายฉลากมีลำดับชั้น

ในบทนี้นำเสนอการจำแนกข้อความขนาดใหญ่แบบหลายฉลากมีลำดับชั้นที่พัฒนาขึ้นโดยปรับปรุงวิธีการแบบแฟลตและนำโครงสร้างลำดับชั้นมาช่วยกรองคำตอบให้ถูกต้องมากยิ่งขึ้น ขั้นตอนการทำงานมีดังนี้ (1) เตรียมข้อมูลให้อยู่ในรูปแบบที่พร้อมจะใช้งานในขั้นตอนถัดไป รวมถึงทำดัชนีคลาสและฟีเจอร์ เพื่อให้เข้าถึงได้ง่ายและเรียกใช้ได้อย่างรวดเร็ว (2) คำนวณหาเซนทรอยด์เวกเตอร์ของแต่ละคลาส เพื่อใช้เป็นตัวแทนของคลาส (3) ตัดฟีเจอร์ที่ปรากฏน้อยครั้งออกจากชุดข้อมูลเรียนรู้ (4) ฝึกตัวจำแนกประเภท SVM ที่โหนดลำดับชั้นบน และนำโมเดลที่ได้มาทำนายคลาสข้อมูลทดสอบ (5) ทำนายคลาสข้อมูลทดสอบด้วยการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด k อันดับที่นำเซนทรอยด์เวกเตอร์และการวัดความคล้ายคลึงเชิงมุมโคไซน์เข้ามาใช้ (6) นำผลการทำนายคลาสจากขั้นตอนที่สี่และห้ามารวมกัน แล้วพิจารณาตามข้อกำหนดเชิงลำดับชั้น จะได้เซตของคลาสที่เป็นคำตอบที่สมบูรณ์ ขั้นตอนการทำงานของวิธีที่นำเสนอแสดงได้ดังรูปที่ 11

รูปที่ 11 ขั้นตอนการทำงานของวิธีที่นำเสนอ

3.1. การเตรียมข้อมูล

ชุดข้อมูลที่ใช้ในงานวิจัยนี้เป็นข้อมูลวิกิพีเดียที่การแข่งขัน LSHTC จัดเตรียมไว้ โดยมีการเปลี่ยนฟีเจอร์จากคำศัพท์ในหน้าวิกิพีเดียให้เป็นตัวเลขทั้งหมด ค่าของฟีเจอร์ คือ ความถี่ของคำนั้นๆ ที่ปรากฏในข้อมูลเรียนรู้หนึ่งตัว (ข้อมูลเรียนรู้หนึ่งตัว คือ วิกิพีเดียหนึ่งหน้า หรือ เอกสารหนึ่งเอกสาร) แต่การนำค่าฟีเจอร์เหล่านี้มาใช้ทำนายคลาสทันที อาจได้ผลไม่ดีนัก เนื่องจากความสำคัญของฟีเจอร์ที่เป็นคำไม่ได้ขึ้นกับความถี่ที่คำนั้นปรากฏในข้อมูลเรียนรู้หนึ่งตัวเท่านั้น แต่ยังขึ้นกับจำนวนคำทั้งหมดในข้อมูลเรียนรู้นั้นและความถี่ที่คำนี้ปรากฏในชุดข้อมูลเรียนรู้ทั้งหมดด้วย เช่น

ข้อมูลเรียนรู้ ก มีคำทั้งหมด 100 คำ คำว่า มหาวิทยาลัย ปรากฏในข้อมูลเรียนรู้ ก จำนวน 5 ครั้ง
ข้อมูลเรียนรู้ ข มีคำทั้งหมด 1,000 คำ คำว่า นักศึกษา ปรากฏในข้อมูลเรียนรู้ ข จำนวน 10 ครั้ง

ถ้าพิจารณาแค่ความถี่ของคำ จะพบว่าคำว่า นักศึกษา มีความถี่มากกว่า น่าจะเป็นคำที่สำคัญกว่า แต่แท้จริงแล้วคำว่า มหาวิทยาลัย สำคัญกว่า เนื่องจากเป็นคำที่ปรากฏถึง 5 เปอร์เซ็นต์ในข้อมูลเรียนรู้ ก ในขณะที่คำว่า นักศึกษา ปรากฏเพียง 1 เปอร์เซ็นต์ในข้อมูลเรียนรู้ ข

จากเหตุผลข้างต้น ผู้วิจัยจึงแปลงค่า tf ของแต่ละฟีเจอร์ในชุดข้อมูลเรียนรู้ให้เป็น $tfidf$ ด้วยสมการที่ (47) (48) และ (49) จากนั้นแปลงค่า tf ของแต่ละฟีเจอร์ในชุดข้อมูลทดสอบด้วยสมการที่ (47) และ (49) โดยใช้ค่า $idf(f, D)$ ที่คำนวณบนชุดข้อมูลเรียนรู้

กำหนดให้ f แทน ฟีเจอร์

d แทน ข้อมูลเรียนรู้

$|d|$ แทน จำนวนคำทั้งหมดในข้อมูลเรียนรู้

D แทน ข้อมูลเรียนรู้ทั้งหมด

$|D|$ แทน จำนวนข้อมูลเรียนรู้ทั้งหมด

n_f แทน จำนวนข้อมูลเรียนรู้ที่มีฟีเจอร์ f

$tf(f, d)$ แทน ความถี่ของฟีเจอร์ f ในเอกสาร d

$$tf_{new}(f, d) = \frac{tf(f, d)}{|d|} \quad (47)$$

$$idf(f, D) = 1 + \ln\left(\frac{|D|}{n_f}\right) \quad (48)$$

$$tfidf(f, d, D) = tf_{new}(f, d) * idf(f, D) \quad (49)$$

นอกจากนี้ ผู้วิจัยได้ทำดัชนีคลาสและฟีเจอร์ด้วยการอ่านชุดข้อมูลเรียนรู้ และเขียนไฟล์เก็บไว้ว่าข้อมูลเรียนรู้แต่ละตัว มีคลาสอะไรบ้าง มีฟีเจอร์อะไรบ้าง และสถิติของคลาสและฟีเจอร์ทั้งหมดเป็นอย่างไร เพื่อให้เข้าถึงได้ง่ายและเรียกใช้ได้อย่างรวดเร็ว เป็นประโยชน์ต่อการทำงานขั้นถัดไป

3.2. คำนวณหาเซนทรอยด์เวกเตอร์

ในขั้นตอนนี้ ผู้วิจัยคำนวณหาเซนทรอยด์เวกเตอร์ของแต่ละคลาส เพื่อใช้เป็นตัวแทนของคลาส เซนทรอยด์เวกเตอร์ คือ เวกเตอร์ที่ศูนย์กลางของชุดตัวอย่างบวกของแต่ละคลาส

เกิดจากการหาค่าเฉลี่ยแต่ละฟีเจอร์ของชุดตัวอย่างบวกที่อยู่ในคลาสเดียวกัน งานวิจัยนี้คำนวณเซนทรอยด์เวกเตอร์ 2 รูปแบบ คือ

รูปแบบที่ 1 Normal Centroid หาค่าเฉลี่ยของแต่ละฟีเจอร์ตามปกติ จำนวนฟีเจอร์ของเซนทรอยด์เวกเตอร์นี้จะเท่ากับจำนวนฟีเจอร์ทั้งหมดที่ปรากฏในคลาส

รูปแบบที่ 2 Decreased Centroid ตัดฟีเจอร์ที่ปรากฏเพียงครั้งเดียวในคลาสที่จะคำนวณหาเซนทรอยด์ออกไป ฟีเจอร์ที่เหลือนำมาคำนวณหาค่าเฉลี่ยตามปกติ การคำนวณรูปแบบนี้ เกิดจากความคิดที่ว่าฟีเจอร์ที่ปรากฏเพียงครั้งเดียวอาจทำให้จุดศูนย์กลางของคลาสเบนไป จึงไม่ควรนำมาคำนวณหาเซนทรอยด์ จำนวนฟีเจอร์ของเซนทรอยด์เวกเตอร์นี้จะเท่ากับจำนวนฟีเจอร์ทั้งหมดที่ปรากฏในคลาสมากกว่าหนึ่งครั้ง

3.3. ตัดฟีเจอร์ที่ปรากฏน้อยครั้งออกจากชุดข้อมูลเรียนรู้

ฟีเจอร์ทั้งหมดในข้อมูลเรียนรู้ของวิกิพีเดียขนาดกลางมีจำนวน 346,299 ฟีเจอร์ และในข้อมูลเรียนรู้ของวิกิพีเดียขนาดใหญ่มีมากถึง 1,617,899 ซึ่งเป็นจำนวนที่มหาศาล เมื่อนำข้อมูลเรียนรู้ไปฝึกตัวจำแนกประเภท เช่น SVM ถ้าต้องอ่านข้อมูลที่มีฟีเจอร์มากเช่นนั้น อาจจะอ่านขึ้นมาเก็บไว้ในหน่วยความจำได้ไม่หมด หรือถ้าอ่านได้ครบ ก็อาจสร้างโมเดลได้ไม่ค่อยดีนัก ผู้วิจัยจึงทดลองลดจำนวนฟีเจอร์ ด้วยการตัดฟีเจอร์ที่ปรากฏน้อยครั้งออกไปให้ได้มากที่สุด โดยต้องไม่มีข้อมูลเรียนรู้ตัวใดถูกตัดฟีเจอร์ออกจนหมด

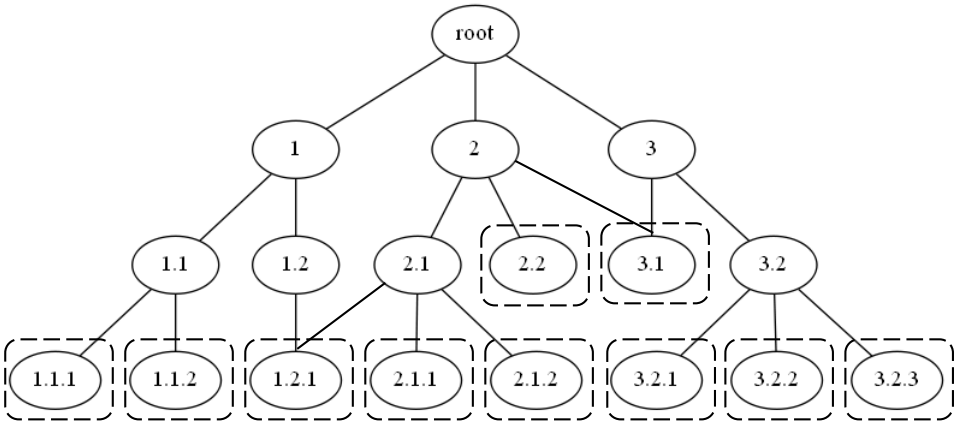
เมื่อทดสอบด้วยข้อมูลวิกิพีเดียขนาดกลาง ปรากฏว่าถ้าตัดฟีเจอร์ที่ปรากฏน้อยกว่า 5 ครั้งออกไปจะลดจำนวนฟีเจอร์ได้มากที่สุดโดยข้อมูลเรียนรู้ยังอยู่ครบทุกตัว ฟีเจอร์จะเหลือเพียง 75,448 ฟีเจอร์ คิดเป็น 21.79 เปอร์เซ็นต์ของฟีเจอร์ทั้งหมด สำหรับข้อมูลวิกิพีเดียขนาดใหญ่ แม้ว่าจะตัดเพียงฟีเจอร์ที่ปรากฏแค่ครั้งเดียวออกไป ข้อมูลเรียนรู้บางตัวก็ถูกตัดฟีเจอร์ออกจนหมดแล้ว ในกรณีนี้เพื่อลดจำนวนฟีเจอร์ ผู้วิจัยจะยอมให้ข้อมูลเรียนรู้ถูกตัดทิ้งไปบ้าง โดยจะเลือกตัดฟีเจอร์ที่ปรากฏน้อยกว่า 5 ครั้งออกไปเช่นเดียวกับข้อมูลวิกิพีเดียขนาดกลาง ฟีเจอร์จะเหลือเพียง 310,037 ฟีเจอร์ คิดเป็น 19.16 เปอร์เซ็นต์ของฟีเจอร์ทั้งหมด และข้อมูลเรียนรู้ที่ถูกตัดทิ้งมีจำนวนแค่ 15 ตัว จากข้อมูลเรียนรู้ทั้งหมด 2,365,436 ตัว

3.4. ฝึกตัวจำแนกประเภท SVM ที่โหนดลำดับชั้นบน

โหนดลำดับชั้นบนที่กล่าวถึงในขั้นตอนนี้ สำหรับข้อมูลวิกิพีเดียขนาดกลางหมายถึง โหนดลูกของโหนดราก ซึ่งโหนดรากของข้อมูลชุดนี้จะมีเพียงโหนดเดียว แต่สำหรับข้อมูลวิกิพีเดียขนาดใหญ่ โหนดลำดับชั้นบนจะหมายถึงโหนดรากเลย

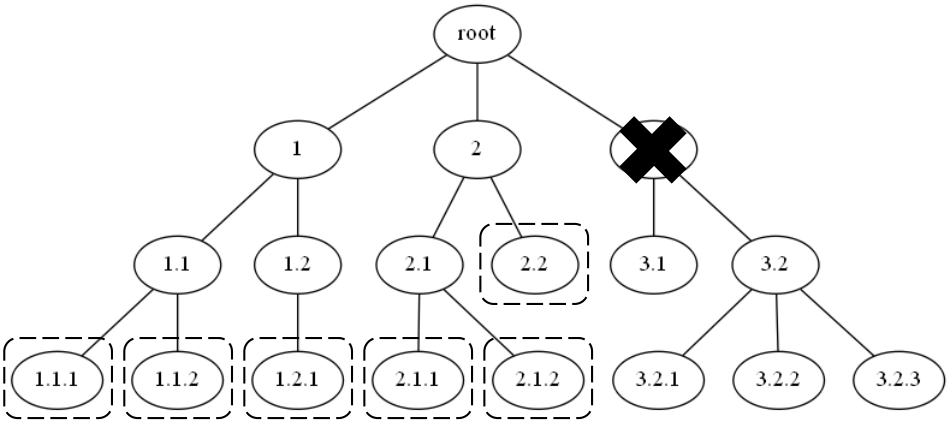
การฝึกตัวจำแนกที่โหนดลำดับชั้นบนมีข้อดีคือ ชุดตัวอย่างบวกของแต่ละคลาสมีเพียงพอ โมเดลทำนายค่อนข้างแม่นยำ และส่งผลอย่างมากต่อความแม่นยำในการจำแนกประเภทของทั้งโครงสร้างลำดับชั้น เพราะถ้าทำนายคลาสที่ชั้นแรกผิด การทำนายในชั้นต่อๆ มาก็จะผิดตามไปด้วย ในทางกลับกันถ้าทำนายถูกก็จะช่วยจำกัดขอบเขตของคำตอบให้น้อยลงได้ เนื่องจากถ้าตัวจำแนกประเภทที่โหนดลำดับชั้นบนทำนายว่าเป็นบวกที่โหนดใด คำตอบที่เป็นไปได้ก็จะเหลือเพียงโหนดลูกหลานของโหนดนี้เท่านั้น

ในขั้นตอนนี้ผู้วิจัยจะฝึกตัวจำแนกประเภท SVM เชิงเส้น (Linear Support Vector Machine) ด้วย LIBLINEAR [23] เพื่อทำนายคลาสที่โหนดลำดับชั้นบนตามที่กล่าวมา โดยเรียกวิธีนี้ว่า Top-Level Pruning (TLP) ยกตัวอย่างการทำงานได้ดังนี้ เมื่อเรามีโครงสร้างลำดับชั้นตามรูปที่ 12 และใช้ TLP ตัวจำแนกประเภทจะถูกฝึกและทำนายคลาสข้อมูลทดสอบที่ลำดับชั้นบนของโครงสร้างฯ คือ ที่โหนด 1 2 และ 3 ถ้าตัวจำแนกประเภทที่โหนด 3 ทำนายว่าข้อมูลทดสอบนั้นไม่ใช่คลาส 3 เราจะตัดโหนด 3 และโหนดลูกหลานทิ้งไปได้ ทำให้คลาสที่มีโอกาสเป็นคำตอบเหลือเพียงคลาสดังรูปที่ 13



รูปที่ 12 ตัวอย่างโครงสร้างลำดับชั้น

โหนดที่อยู่ในสี่เหลี่ยมเส้นประคือ โหนดใบที่มีโอกาสเป็นคำตอบของการทำนายคลาส



รูปที่ 13 ตัวอย่างการใช้ Top-Level Pruning

โหนดใบที่มีโอกาสเป็นคำตอบลดจำนวนลง

ขั้นตอนนี้ เมื่อฝึกตัวจำแนกประเภทจนครบด้วยข้อมูลวิกิพีเดียขนาดกลางและขนาดใหญ่ จะได้โมเดลทำนาย 4 โมเดล และ 11,814 โมเดล ตามลำดับ เมื่อนำข้อมูลทดสอบมาทำนายคลาสกับโมเดลจะได้เซตคลาสที่มีโอกาสเป็นคำตอบ

การนำ TLP ไปใช้จริง ถ้าเรากำหนดเกณฑ์อย่างเข้มงวดตามตัวอย่างข้างต้น มีโอกาสที่จะเหลือคลาสที่มีโอกาสเป็นคำตอบน้อยและไม่ครอบคลุมคำตอบที่ถูกต้อง ดังนั้นผู้วิจัยจึงกำหนดเงื่อนไขว่า คลาสจะไม่ถูกตัดทิ้ง ถ้าคลาสนั้นมีโหนดที่ลำดับชั้นบนทำนายผลเป็นบวกเกินครึ่งหนึ่ง ยกตัวอย่างการทำงานได้ดังนี้

ข้อมูลทดสอบ t มีผลการทำนายคลาสที่โหนดลำดับชั้นบนเป็นบวกที่คลาส 1 และ 2 พิจารณาโหนดใบทั้งหมดว่ามีโหนดลำดับชั้นบนโหนดใดบ้าง และผ่านเกณฑ์ที่โหนดลำดับชั้นบนทำนายผลเป็นบวกเกินครึ่งหนึ่งหรือไม่ ดังตารางที่ 2

ตารางที่ 2 โหนดลำดับชั้นบนของโหนดใบแต่ละโหนด และการประเมินว่าผ่านเกณฑ์หรือไม่

โหนดใบ	โหนดลำดับชั้นบน	ผ่านเกณฑ์
1.1.1	1	✓
1.1.2	1	✓
1.2.1	1 และ 2	✓
2.1.1	2	✓
2.1.2	2	✓

โหนดใบ	โหนดลำดับชั้นบน	ผ่านเกณฑ์
2.2	2	✓
3.1	2 และ 3	×
3.2.1	3	×
3.2.2	3	×
3.2.3	3	×

จากตารางที่ 2 จะได้ว่าคลาสที่มีโอกาสเป็นคำตอบที่เหลืออยู่ ได้แก่ คลาส 1.1.1 1.1.2 1.2.1 2.1.1 2.1.2 และ 2.2

3.5. ทำนายคลาสข้อมูลทดสอบด้วยการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด k อันดับที่นำเซนทรอยด์เวกเตอร์และการวัดความคล้ายคลึงเชิงมุมเข้ามาใช้[8]

ผู้วิจัยศึกษาวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด 5 อันดับที่ใช้เป็นเกณฑ์มาตรฐานใน LSHTC ที่ได้กล่าวถึงในบทที่ 2 หัวข้อที่ 2.1.8. ทดลองทำซ้ำตามขั้นตอน และปรับปรุงการจัดอันดับคลาสคำตอบเพิ่มเติมที่ขั้นตอนที่ 2.3 ดังนี้

กำหนดให้ T แทน เซตของข้อมูลเรียนรู้ทั้งหมด

1. แปลงค่าของฟีเจอร์ทุกตัวในข้อมูลเรียนรู้และข้อมูลทดสอบจาก tf เป็น $tfidf$
2. สำหรับข้อมูลทดสอบ t แต่ละตัว
   2.1. หาฟีเจอร์ที่มีค่า $tfidf$ มากที่สุด 3 อันดับแรกของข้อมูลทดสอบ t และเรียกเซตของฟีเจอร์นี้ว่า $3F$
   2.2. สร้างเซต S ซึ่งเป็นเซตย่อยของเซต T โดยข้อมูลเรียนรู้แต่ละตัวใน S ต้องมีอย่างน้อย 1 ฟีเจอร์ที่ปรากฏในเซต $3F$
   2.3. คำนวณระยะทางระหว่างข้อมูลเรียนรู้จากเซต S กับข้อมูลทดสอบ t ได้ด้วยสมการที่ (29) แล้วหาข้อมูลเรียนรู้ที่อยู่ใกล้ที่สุด k อันดับแรก จากนั้นนำคลาสของข้อมูลเรียนรู้เหล่านี้ มาเรียงลำดับตามเกณฑ์ต่อไปนี้
      i. เรียงลำดับด้วยความถี่ที่คลาสนั้นปรากฏจากมากไปน้อย
      ii. เรียงลำดับด้วยผลรวมของระยะทางระหว่างข้อมูลเรียนรู้จากเซต S ที่มีคลาสที่กำลังพิจารณากับข้อมูลทดสอบ t จากน้อยไปมาก
      iii. เรียงลำดับด้วยจำนวนข้อมูลเรียนรู้ที่มีคลาสที่กำลังพิจารณาเป็นคำตอบจากน้อยไปมาก
      iv. เรียงลำดับด้วยเลขคลาสจากน้อยไปมาก

[8] ขั้นตอนนี้จะใช้ฟีเจอร์ทั้งหมด ต่างจากขั้นตอนฝึกตัวจำแนกประเภท SVM ที่โหนดลำดับชั้นบนซึ่งตัดฟีเจอร์ที่ปรากฏน้อยกว่า 5 ครั้งออกไป

สุดท้ายคืนค่าคลาสที่ปรากฏบ่อยที่สุด x อันดับแรกเป็นคำตอบของข้อมูลทดสอบ t

2.3.1.กรณีที่มีน้อยกว่า x คลาส จะตอบเท่ากับคลาสที่มี

2.3.2.กรณีที่มีคลาสที่ปรากฏบ่อยที่สุดเท่ากันมากกว่า x คลาส จะตอบคลาสเหล่านั้นทั้งหมด (ผู้วิจัยขอเรียกกรณีนี้ว่า x+ เช่น 2+ คือ กรณีที่มีคลาสที่ปรากฏบ่อยที่สุดเท่ากันมากกว่า 2 คลาส เป็นต้น)

ผู้วิจัยทำการทดสอบด้วยค่า k เท่ากับ 5 และ 7 และค่า x เท่ากับ 1 ถึง k ผลการทำนายคลาสของข้อมูลแสดงไว้ในภาคผนวก ก จากนั้นผู้วิจัยทดลองปรับปรุงใหม่ตั้งแต่ขั้นตอนที่ 2.3 โดยนำเซนทรอยด์เวกเตอร์และการวัดความคล้ายคลึงเชิงมุมเข้ามาใช้ ดังนี้

2.3. หาข้อมูลเรียนรู้จากเซต S ที่มีระยะทางใกล้กับข้อมูลทดสอบ t มากที่สุด k อันดับแรก สร้างเซต C ซึ่งเป็นเซตของคลาสของข้อมูลเรียนรู้เหล่านี้

2.4. นำเซนทรอยด์เวกเตอร์ของแต่ละคลาสในเซต C มาวัดความคล้ายคลึงเชิงมุมโคไซน์กับ t โดยความคล้ายคลึงเชิงมุมโคไซน์คำนวณได้ด้วยสมการที่ (50)

$$Cosine\ similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \qquad (50)$$

เมื่อ A และ B คือ เวกเตอร์ที่จะคำนวณความคล้ายคลึงเชิงมุมโคไซน์

$\|A\|$ และ $\|B\|$ คือ ขนาดของเวกเตอร์ A และ B ตามลำดับ

2.5. คืนค่าคลาสที่มีความคล้ายคลึงเชิงมุมโคไซน์มากที่สุด x อันดับแรกเป็นคำตอบของข้อมูลทดสอบ t โดยมีเงื่อนไขเพิ่มเติมว่าค่าความคล้ายฯ ต้องมากกว่า 0.2 ซึ่งเป็นค่าเฉลี่ยของค่าความคล้ายฯ ทั้งหมด เพื่อป้องกันการเลือกคลาสที่ไม่คล้ายกับข้อมูลทดสอบมาเป็นคำตอบ ถ้าพบว่าทุกคลาสมีค่าความคล้ายฯ น้อยกว่าค่าเฉลี่ย ในข้อมูลวิกิพีเดียขนาดกลางจะเลือกโหนดรากเป็นคำตอบ ในขณะที่ข้อมูลวิกิพีเดียขนาดใหญ่จะเลือกคลาสที่มีชุดตัวอย่างบวกมากที่สุดเป็นคำตอบ

2.5.1.กรณีที่มีน้อยกว่า x คลาส จะตอบเท่ากับคลาสที่มี

2.5.2.กรณีที่มีคลาสที่ความคล้ายคลึงเชิงมุมโคไซน์มากที่สุดเท่ากันมากกว่า x คลาส จะตอบคลาสเหล่านั้นทั้งหมด

ผู้วิจัยทำการทดสอบด้วยค่า k เท่ากับ 5 7 และ 10 ค่า x เท่ากับ 1 ถึง k และเลือกใช้เซนทรอยด์เวกเตอร์ 2 รูปแบบที่คำนวณไว้ตามบทที่ 3 หัวข้อที่ 3.2 พบว่าบนข้อมูลวิกิพีเดียขนาดกลางได้ประสิทธิภาพการจำแนกมากที่สุดเมื่อ กำหนดค่า k = 10 ค่า x = 2+ และเลือกใช้ Decreased Centroid ในขณะที่บนข้อมูลวิกิพีเดียขนาดใหญ่ได้ประสิทธิภาพการจำแนกมากที่สุดเมื่อ กำหนดค่า k = 10 ค่า x = 3 และเลือกใช้ Decreased Centroid ซึ่งจำนวน 2 และ 3 เท่ากับจำนวนคลาสโดยเฉลี่ยของข้อมูลเรียนรู้หนึ่งตัวในข้อมูลวิกิพีเดียขนาดกลางและขนาดใหญ่ ตามลำดับ ผลการทดลองเมื่อกำหนดตัวแปรเป็นค่าต่างๆ แสดงไว้ในภาคผนวก ข

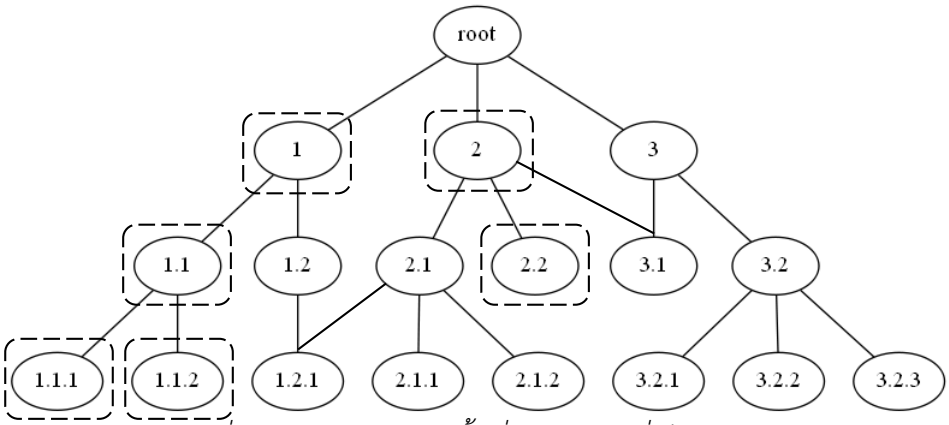
3.6. รวมผลการทำนายคลาสและปรับคลาสคำตอบตามข้อกำหนดเชิงลำดับชั้น

ในขั้นตอนสุดท้าย สร้างเซตคำตอบของข้อมูลทดสอบด้วยคลาสที่ได้จากการทำนายในขั้นตอนที่ห้า โดยมีเงื่อนไขว่าคลาสนั้นๆ ต้องอยู่ในเซตคลาสที่มีโอกาสเป็นคำตอบจากขั้นตอนที่สี่ด้วย ยกตัวอย่างเช่น สมมติคลาสที่ได้จากการทำนายในขั้นตอนที่ห้า คือ คลาส 1.1.1 1.1.2 2.2 และ 3.1

เซตคลาสที่มีโอกาสเป็นคำตอบจากขั้นตอนที่สี่ ประกอบด้วย คลาส 1.1.1 1.1.2 1.2.1 2.1.1 2.1.2 และ 2.2

เซตคำตอบที่สร้างขึ้นจะมีสมาชิก 3 ตัว คือ คลาส 1.1.1 1.1.2 และ 2.2

สุดท้ายพิจารณาตามข้อกำหนดเชิงลำดับชั้น จะได้เซตของคลาสที่เป็นคำตอบที่สมบูรณ์ คือ คลาส 1 1.1 1.1.1 1.1.2 2 และ 2.2 ดังรูปที่ 14



รูปที่ 14 โครงสร้างลำดับชั้นที่แสดงคลาสที่เป็นคำตอบ

3.7. การวิเคราะห์ความซับซ้อนเชิงเวลา

ผู้วิจัยประเมินเวลาที่ใช้ในการประมวลผล ทั้งเวลาในการฝึกตัวจำแนกประเภท (Train) และการทำนายคลาส (Test) โดยอ้างอิงจากบทความวิชาการของผู้เข้าแข่งขัน LSHTC แต่ละคนเท่าที่มีการเปิดเผยในเว็บไซต์ เปรียบเทียบกับวิธีที่นำเสนอ ได้ผลดังตารางที่ 3 โดย

กำหนดให้ n แทน จำนวนข้อมูลเรียนรู้
n_{avg} แทน จำนวนข้อมูลเรียนรู้เฉลี่ยต่อหนึ่งคลาส
n' แทน จำนวนข้อมูลเรียนรู้ที่ถูกเลือก
l แทน จำนวนคลาสที่เป็นโหนดใบ
d แทน จำนวนฟีเจอร์ทั้งหมด
d_{avg} แทน จำนวนฟีเจอร์เฉลี่ยของข้อมูลหนึ่งตัว
e แทน จำนวนกิ่งในโครงสร้างฯ
k แทน จำนวนคลาสที่ถูกเลือก
s แทน จำนวนคลาสที่มากที่สุดในเซตคำตอบของข้อมูลเรียนรู้ (จำนวนขั้นตอนที่วนซ้ำ)

ตารางที่ 3 ผลการประเมินเวลาที่ใช้ในการประมวลผลของอัลกอริทึมแบบต่างๆ

ชื่อผู้เข้าร่วมการแข่งขัน LSHTC / อัลกอริทึม	**ความซับซ้อนเชิงเวลา (Time Complexity)**	
	Train (ต่อ 1 คลาส)	Test (ต่อข้อมูลเรียนรู้ 1 ตัว)
arthur	$O(nd^2)$	$O(d+l)$
chrishan	-	$O(nd + n\log n)$
dhlee	-	$O(skn_{avg}(l+d_{avg}) + kld_{avg})$
TTI	$O(nd^2)$ (ต่อ 1 กิ่ง)	$O(ed+l)$
LSHTC's k-NN Baseline	-	$O(n'd + n'\log n')$
วิธีการแบบแฟลต (One vs. Rest)	$O(nd)$	$O(d)$
วิธีที่นำเสนอ	$O(nd)$	$O(n'd + n'\log n')$

จากตารางที่ 3 จะเห็นว่าบางอัลกอริทึมไม่ได้แสดงเวลาในการฝึกตัวจำแนกประเภท เนื่องจากอัลกอริทึมเหล่านี้มีรูปแบบการทำงานแบบ k-NN คือ ไม่ได้ฝึกตัวจำแนกประเภท และการทำงานทั้งหมดจะเกิดขึ้นพร้อมการทำนายคลาสเลย ส่วนอัลกอริทึมที่แสดงเวลาในการฝึกตัวจำแนกฯ ทุกวิธีในตารางนี้เป็นการฝึก SVM จึงมีความซับซ้อนเชิงเวลาเท่ากัน คือ $O(nd^2)$ ยกเว้นอัลกอริทึมที่ใช้ LIBLINEAR [23] จะลดเวลาลงได้ เหลือเพียง $O(nd)$

เปรียบเทียบความซับซ้อนเชิงเวลาของวิธีที่นำเสนอกับวิธีของ dhlee ที่ได้ประสิทธิภาพดีเป็นอันดับที่ 1 บนข้อมูลทั้งสองชุดได้ดังนี้ วิธีที่นำเสนอมีความซับซ้อนเชิงเวลาขึ้นกับจำนวนข้อมูลเรียนรู้ที่ถูกเลือก และจำนวนฟีเจอร์ เมื่อทำการจำแนกประเภทเสร็จทุกขั้นตอนตามวิธีที่นำเสนอ เราจะเลือกจำนวนข้อมูลเรียนรู้ที่อยู่ใกล้ที่สุด k อันดับ และจำนวนคลาสคำตอบที่ต้องการได้โดยไม่ต้องคำนวณใหม่

วิธีของ dhlee มีความซับซ้อนเชิงเวลาขึ้นกับจำนวนคลาสที่เป็นโหนดใบ จำนวนข้อมูลเรียนรู้เฉลี่ยต่อหนึ่งคลาส จำนวนฟีเจอร์เฉลี่ยของข้อมูลหนึ่งตัว จำนวนคลาสที่ถูกเลือก และจำนวนคลาสที่มากที่สุดในเซตคำตอบของข้อมูลเรียนรู้ โดยรวมแล้วขึ้นกับค่าเฉลี่ยหรือจำนวนที่มีน้อย อาจบอกได้ว่าถ้าชุดข้อมูลมีค่าเหล่านั้นน้อย วิธีการนี้ก็จะได้ผลดีและทำงานได้เร็วกว่าวิธีที่นำเสนอ ในทางตรงกันข้าม เมื่อข้อมูลยิ่งซับซ้อน เช่น เซตคำตอบที่เป็นไปได้ของข้อมูลเรียนรู้มีเยอะมาก การใช้วิธีการนี้ก็จะไม่เหมาะนัก ในขณะที่ปัจจัยเหล่านี้ไม่ส่งผลกระทบต่อวิธีการที่ผู้วิจัยนำเสนอ

เมื่อเปรียบเทียบจำนวนเซนทรอยด์เวกเตอร์มากที่สุดที่ทั้งสองวิธีต้องคำนวณ ได้ดังตารางที่ 4 วิธีการของ dhlee จะคำนวณเซนทรอยด์เวกเตอร์มากกว่าวิธีที่ผู้วิจัยเสนอค่อนข้างมาก ขึ้นกับค่า k และจำนวน Label-Power set ของคลาสเหล่านั้น โดยตัวเลขในวงเล็บในตารางที่ 4 แสดงจำนวนเซตคลาสคำตอบทั้งหมดที่เป็นไปได้ของข้อมูลชุดนั้น

ตารางที่ 4 เปรียบเทียบจำนวนเซนทรอยด์เวกเตอร์มากที่สุดที่วิธีที่นำเสนอและวิธีการของ dhlee ต้องคำนวณ

	วิธีที่นำเสนอ	**วิธีการของ dhlee**
จำนวนเซนทรอยด์เวกเตอร์มากที่สุดที่ต้องคำนวณ	จำนวนคลาสที่เป็นโหนดใบ	จำนวนคลาสที่เป็นโหนดใบ + จำนวน Label-Power set ของคลาสที่เลือกมา k ตัว
ข้อมูลวิกิพีเดียขนาดกลาง	36,504	36,504 (+ 171,495)
ข้อมูลวิกิพีเดียขนาดใหญ่	325,056	325,056 (+ 1,470,337)

บทที่ 4

การทดลองและวิเคราะห์ผล

4.1. ชุดข้อมูลที่ใช้ในงานวิจัย

ข้อมูลที่นำมาใช้ทดสอบการจำแนกประเภทในการวิจัยนี้ คือ ข้อมูลวิกิพีเดียขนาดใหญ่ (Wiki-Large) จาก LSHTC4 ประกอบด้วย ชุดข้อมูลเรียนรู้ ชุดข้อมูลทดสอบ และโครงสร้างลำดับชั้น ชุดข้อมูลทั้งสองชุดจัดรูปแบบตามชุดข้อมูลของ libSVM [26] แต่ละแถวในไฟล์ชุดข้อมูล คือ ข้อมูลหนึ่งตัวที่อยู่ในรูปแบบ sparse vector ประกอบด้วย คลาส ฟีเจอร์และค่าของฟีเจอร์ ดังรูปที่ 15 ค่าของฟีเจอร์ คือ ความถี่ของฟีเจอร์นั้นๆ ในข้อมูลแต่ละตัว และด้วยรูปแบบ sparse vector ทำให้ฟีเจอร์ที่ปรากฏในเวกเตอร์มีค่าของฟีเจอร์มากกว่า 0 เสมอ

$$\text{class1, class2, class3 ... feature1:value1 ... feature}_m\text{:value}_m$$

รูปที่ 15 ข้อมูลที่อยู่ในรูปแบบ sparse vector

การแข่งขันนี้ไม่ได้ให้คลาสคำตอบของข้อมูลทดสอบแต่ละตัวมาพร้อมชุดข้อมูล คลาสของข้อมูลทดสอบจึงใส่เลข 0 เอาไว้ ข้อมูลเรียนรู้และข้อมูลทดสอบแต่ละตัว อาจจัดอยู่ได้ในหลายคลาส คลาสเหล่านั้นมีความสัมพันธ์ตามโครงสร้างลำดับชั้นที่กำหนดมาให้ โดยโครงสร้างฯ เป็นกราฟที่มีวัฏจักร ก่อนทำนายคลาสจึงต้องกำจัดวัฏจักรก่อน เพื่อให้ทำนายคลาสได้ถูกต้อง

นอกจากนี้ผู้วิจัยได้นำข้อมูลวิกิพีเดียขนาดกลาง (Wiki-Medium) มาใช้ทดสอบวิธีการจำแนกประเภทที่นำเสนอก่อนจะนำไปทดสอบกับข้อมูลวิกิพีเดียขนาดใหญ่ สถิติของข้อมูลวิกิพีเดียทั้งสองชุดแสดงได้ดังตารางที่ 5

ตารางที่ 5 สถิติของข้อมูลวิกิพีเดียขนาดกลางเปรียบเทียบกับข้อมูลวิกิพีเดียขนาดใหญ่

ชุดข้อมูล	จำนวนข้อมูลเรียนรู้	จำนวนข้อมูลทดสอบ	จำนวนคลาสที่เป็นโหนดใบ	จำนวนฟีเจอร์	ความลึกของโครงสร้างลำดับชั้น
วิกิพีเดียขนาดกลาง	456,886	81,262	36,504	346,299	12
วิกิพีเดียขนาดใหญ่	2,365,436	452,167	325,056	1,617,899	14

4.2. การวัดประสิทธิภาพการทำนายคลาส

การแข่งขัน LSHTC ไม่ได้ให้คลาสคำตอบของข้อมูลทดสอบแต่ละตัวมาพร้อมชุดข้อมูลการประเมินประสิทธิภาพการทำนายคลาสทำได้ด้วยการส่งไฟล์คลาสที่ได้จากการทำนายให้เว็บ [10] ตรวจสอบเท่านั้น แม้ว่าการแข่งขันจะจบแล้ว แต่ผู้ที่ต้องการประเมินประสิทธิภาพก็ยังคงส่งไฟล์ไปตรวจสอบได้เช่นเดิม

ตัววัดประสิทธิภาพการจำแนกข้อมูลที่เว็บประเมินให้ ประกอบด้วย $Accuracy, EBP$ $EBR, EBF, LBMaP, LBMaR, LBMaF, LBMiP, LBMiR$ และ$LBMiF$ โดยตัววัดฯ ที่การแข่งขันสนใจมากที่สุด คือ $LBMaF$

วิธีการอื่นที่นำมาเปรียบเทียบกับวิธีการที่นำเสนอ ได้แก่ วิธีการแบบแฟลตที่สร้างตัวจำแนกประเภทที่โหนดใบทุกโหนดด้วยข้อมูลเรียนรู้ทั้งหมด วิธี k-NN ที่ใช้เป็นเกณฑ์มาตรฐานของ LSHTC และวิธีการจำแนกประเภทของผู้เข้าแข่งขัน LSHTC ที่ได้ค่า $LBMaF$ สูงที่สุด 5 อันดับแรก (ผลการประเมินฯ ของผู้เข้าแข่งขันทั้งหมดแสดงไว้ในภาคผนวก ค) ผลการประเมินประสิทธิภาพการจำแนกประเภทชุดข้อมูลวิกิพีเดียขนาดกลางแสดงไว้ในตารางที่ 6

ตารางที่ 6 ผลการประเมินประสิทธิภาพการจำแนกประเภทชุดข้อมูลวิกิพีเดียขนาดกลาง

ชื่อผู้เข้าร่วมการแข่งขัน LSHTC / อัลกอริทึม	$LBMaPr$	$LBMaRe$	$LBMaF$
TTI (LSHTC)[9]	50.64%	30.69%	28.35%
dhlee (LSHTC)	47.58%	31.80%	28.24%
arthur (LSHTC)	57.33%	28.76%	26.74%
วิธีที่นำเสนอ	39.91%	29.42%	25.70%
coolvegpuff (LSHTC)	52.61%	25.67%	25.07%
chrishan (LSHTC)	42.62%	29.96%	24.54%
k-NN Baseline	25.22%	23.54%	17.58%
วิธีการแบบแฟลต	45.89%	7.57%	8.89%

[9] ชื่อที่ระบุ (LSHTC) ไว้ท้ายชื่อ หมายถึง ผู้เข้าร่วมการแข่งขัน LSHTC

จากตารางที่ 6 วิธีที่นำเสนอมีค่า $LBMaF$ เท่ากับ 25.70% อยู่อันดับที่ 4 เมื่อพิจารณาค่า $LBMaPr$ และ $LBMaRe$ พบว่าค่า $LBMaRe$ ของวิธีที่นำเสนอใกล้เคียงกับวิธีอื่นที่อันดับสูงกว่า ส่วน $LBMaPr$ ของวิธีที่นำเสนอมีค่าค่อนข้างต่ำกว่า แสดงว่าคลาสที่วิธีที่นำเสนอทำนายได้ยังไม่ตรงกับคำตอบที่ถูกต้องมากเท่าวิธีอื่น

เปรียบเทียบวิธีที่นำเสนอกับวิธีการแบบแฟลต และวิธี k-NN จะเห็นว่าค่า $LBMaF$ ของวิธีที่นำเสนอมีค่ามากกว่าวิธีมาตรฐานทั้งสองวิธี แต่ก็น่าสังเกตว่าค่า $LBMaF$ ของวิธีการแบบแฟลตที่น้อยเกิดจากการตอบคลาสได้น้อย แต่คลาสที่ตอบได้ก็ถูกต้องค่อนข้างมากเช่นกัน เห็นได้จากค่า $LBMaRe$ น้อย แต่ค่า $LBMaPr$ มาก ผลการประเมินประสิทธิภาพการจำแนกประเภทชุดข้อมูลวิกิพีเดียขนาดใหญ่แสดงไว้ในตารางที่ 7

ตารางที่ 7 ผลการประเมินประสิทธิภาพการจำแนกประเภทชุดข้อมูลวิกิพีเดียขนาดใหญ่

ชื่อผู้เข้าร่วมการแข่งขัน LSHTC / อัลกอริทึม	$LBMaPr$	$LBMaRe$	$LBMaF$
dhlee (LSHTC)[10]	40.86%	30.78%	25.64%
วิธีที่นำเสนอ	33.35%	28.16%	23.48%
anttip (LSHTC)	33.31%	24.55%	22.45%
coolvegpuff (LSHTC)	32.49%	24.32%	21.74%
daq (LSHTC)	32.09%	20.31%	17.38%
chrishan (LSHTC)	53.79%	17.60%	17.01%
k-NN Baseline	30.33%	17.70%	14.86%
วิธีการแบบแฟลต[11]	NA	NA	NA

[10] ชื่อที่ระบุ (LSHTC) ไว้ท้ายชื่อ หมายถึง ผู้เข้าร่วมการแข่งขัน LSHTC

[11] ใช้เวลาสร้างตัวจำแนกนานมาก ไม่อาจสร้างได้ครบในเวลาที่จำกัด จึงไม่มีผลประเมินประสิทธิภาพ

จากตารางที่ 7 วิธีที่นำเสนอมีค่า $LBMaF$ เท่ากับ 23.48% อยู่อันดับที่ 2 เมื่อพิจารณาค่า $LBMaPr$ และ $LBMaRe$ จะพบว่าค่า $LBMaPr$ ของวิธีที่นำเสนอใกล้เคียงกับวิธีอื่นที่อันดับต่ำกว่า ในขณะที่ค่า $LBMaRe$ มีค่ามากกว่า แสดงว่าวิธีที่นำเสนอทำนายคลาสได้ครอบคลุมคำตอบมากกว่าวิธีอื่น จึงทำให้ได้ค่า $LBMaF$ มากกว่า

เมื่อพิจารณาผลการประเมินประสิทธิภาพโดยรวมแล้ว พบว่าผู้เข้าแข่งขันที่นำเสนอวิธีการจำแนกประเภทที่ประมวลผลข้อมูลได้ทั้งสองชุด นอกเหนือจากวิธีที่ผู้วิจัยนำเสนอมีเพียง 4 คนเท่านั้น ได้แก่ dhlee anttip[12] coolvegpuff และ chrishan โดยวิธีการจำแนกฯ ถ้ามองแค่ 5 วิธีนี้ dhlee จะมีค่า LBMaF มากเป็นอันดับ 1 บนข้อมูลทั้งสองชุด ในขณะที่วิธีที่นำเสนอได้อันดับ 2 บนข้อมูลทั้งสองชุดเช่นกัน

[12] ผลการประเมินฯ ของ anttip บนข้อมูลวิกิพีเดียขนาดกลางได้อันดับที่ 7 จึงไม่ได้แสดงไว้ในตารางที่ 6

บทที่ 5
สรุปผลการวิจัย

5.1. สรุปผลการวิจัย

การจำแนกประเภทแบบหลายฉลากมีลำดับชั้น เป็นการจำแนกประเภทที่รวมลักษณะเฉพาะของปัญหาสองรูปแบบคือ ข้อมูลแต่ละตัวอาจจัดอยู่ได้ในหลายคลาส และคลาสเหล่านี้มีความสัมพันธ์เป็นโครงสร้างลำดับชั้น ซึ่งข้อมูลในชีวิตจริงมักจะมีลักษณะซับซ้อนเช่นนี้ เช่น บทความ หน้าเว็บ ฟังก์ชันการทำงานของโปรตีน เป็นต้น

การจำแนกประเภทข้อความแบบหลายฉลากมีลำดับชั้น เป็นหัวข้อการวิจัยที่ได้รับความสนใจอย่างมากในปัจจุบัน เพราะโครงสร้างลำดับชั้นใช้อธิบายความสัมพันธ์ของข้อมูลประเภทข้อความได้ดี และข้อมูลประเภทข้อความที่เราพบอยู่ทุกวันก็คือ ข้อมูลบนเว็บไซต์นั่นเอง เมื่อมนุษย์สร้างข้อมูลได้อย่างเสรีดังเช่นในปัจจุบัน ข้อมูลและเว็บไซต์ก็เพิ่มจำนวนขึ้นอย่างรวดเร็ว เว็บไซต์อย่างเว็บไดเรกทอรีและวิกิพีเดียที่ใช้งานกันอย่างแพร่หลายและแทบจะตลอดเวลา จึงจำเป็นต้องมีระบบการจำแนกประเภทอย่างอัตโนมัติเมื่อมีหน้าเว็บใหม่เพิ่มเข้ามาในฐานข้อมูล ปัญหานี้ถือเป็นการจำแนกข้อความขนาดใหญ่แบบหลายฉลากมีลำดับชั้น

งานวิจัยหลายงานนำเสนอวิธีแก้ปัญหาดังกล่าว แต่วิธีเหล่านั้นใช้กับการประมวลผลข้อมูลที่มีขนาดใหญ่ไม่ได้ เนื่องจากการประมวลผลอาจต้องใช้พื้นที่เก็บข้อมูลขนาดใหญ่มาก อาจใช้เวลาประมวลผลนานเกินไป หรือแทบไม่มีความแม่นยำในการจำแนกประเภท วิธีการส่วนใหญ่ที่พอจะรองรับข้อมูลขนาดใหญ่ได้ก็ไม่ได้นำโครงสร้างลำดับชั้นมาใช้ให้เกิดประโยชน์

งานวิจัยนี้จึงได้นำเสนอการจำแนกข้อความขนาดใหญ่แบบหลายฉลากมีลำดับชั้นที่ปรับปรุงวิธีการ k-NN ซึ่งเป็นวิธีการแบบแฟลต และนำโครงสร้างลำดับชั้นมาใช้ด้วยการฝึกตัวจำแนกประเภท SVM ที่โหนดชั้นบนของโครงสร้างฯ เพื่อช่วยกรองคำตอบให้มีความถูกต้องแม่นยำมากขึ้น นอกจากนี้ยังมีการตัดฟีเจอร์ที่ปรากฏน้อยครั้งออกไปเพื่อช่วยลดจำนวนฟีเจอร์ และนำฟีเจอร์สำคัญของข้อมูลทดสอบมาช่วยเลือกข้อมูลเรียนรู้เพื่อลดข้อมูลที่จะต้องพิจารณาอีกด้วย

ผลการประเมินประสิทธิภาพของวิธีที่นำเสนอบนข้อมูลวิกิพีเดียขนาดกลางและขนาดใหญ่เปรียบเทียบกับวิธีการแบบแฟลต วิธี k-NN และอัลกอริทึมของผู้เข้าแข่งขัน LSHTC คนอื่น พบว่าวิธีที่นำเสนอมีค่า $LBMaF$ มากเป็นอันดับที่ 4 บนข้อมูลขนาดกลาง และมากเป็นอันดับที่ 2 บนข้อมูลขนาดใหญ่ ถ้าพิจารณาเฉพาะอัลกอริทึมที่ทำงานได้บนข้อมูลทั้งสองชุด จะพบว่าวิธีที่นำเสนอมี

ประสิทธิภาพดีเป็นอันดับที่ 2 บนข้อมูลทั้งสองชุด และวิธีที่ผู้เข้าแข่งขัน dhlee นำเสนอได้ประสิทธิภาพดีที่สุดบนข้อมูลทั้งสองชุด เมื่อเปรียบเทียบความซับซ้อนเชิงเวลาดังเช่นในบทที่ 3 หัวข้อที่ 3.7 จะเห็นว่าถ้าเป็นข้อมูลทั่วไปที่จำนวนข้อมูลเรียนรู้เฉลี่ยต่อหนึ่งคลาสและจำนวนฟีเจอร์เฉลี่ยของข้อมูลหนึ่งตัวมีค่าไม่มากนัก วิธีการของ dhlee จะได้ผลดีและทำงานได้เร็วกว่าวิธีที่นำเสนอ แต่วิธีการของเขาจะใช้เวลานานขึ้นเรื่อยๆ ถ้าเซตของคลาสคำตอบที่เป็นไปได้ในข้อมูลเรียนรู้มีจำนวนมากขึ้น นอกจากนี้จำนวนเซนทรอยด์ที่ต้องคำนวณจะเพิ่มขึ้นด้วยเช่นกัน ในขณะที่ปัจจัยนี้ไม่ส่งผลกระทบต่อวิธีที่ผู้วิจัยนำเสนอ

5.2. ข้อจำกัดของงานวิจัย

ข้อจำกัดของงานวิจัยนี้ประกอบด้วย

(1) โครงสร้างลำดับชั้นต้องไม่มีวัฏจักร หรือถ้ามี ต้องกำจัดวัฏจักรก่อนนำวิธีที่นำเสนอไปใช้

(2) ส่วนหนึ่งของวิธีที่นำเสนอต้องใช้ฟีเจอร์ที่มีค่า tfidf มากที่สุด 3 อันดับแรกของข้อมูลทดสอบแต่ละตัวในชุดข้อมูลทดสอบ เพราะฉะนั้นเมื่อนำไปใช้กับข้อมูลชุดอื่น จะต้องทราบฟีเจอร์เหล่านี้ด้วย หรืออาจปรับใช้เป็นการเลือกฟีเจอร์ที่น่าจะสำคัญก็ได้เช่นกัน

(3) การใช้ LIBLINEAR ทำให้ฝึกตัวจำแนกได้เร็วขึ้นมาก แต่ก็ต้องใช้หน่วยความจำ (RAM) เพิ่มขึ้นเช่นกัน

5.3. แนวทางการวิจัยในอนาคต

แนวทางการวิจัยในอนาคต มีดังนี้

(1) การฝึกตัวจำแนกประเภทที่โหนดใบหรือโหนดภายใน หลังจากฝึกตัวจำแนกประเภทที่ลำดับชั้นบนแล้ว โดยกำหนดชุดข้อมูลเรียนรู้ด้วยวิธี Exclusive Top-Level Training Policy (ETT) ตามที่ผู้วิจัยได้นำเสนอไว้ใน [27] จะช่วยเพิ่มความแม่นยำในการจำแนกประเภทได้

(2) ปรับปรุงการคำนวณเซนทรอยด์เวกเตอร์หรือคำนวณหาตัวแทนคลาสด้วยวิธีอื่น เพื่อใช้เป็นตัวแทนคลาสได้ดียิ่งขึ้น

5.4. ผลงานตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์ฉบับนี้ ตีพิมพ์เป็นบทความทางวิชาการ 1 หัวข้อ ได้แก่

“An Improvement of Flat Approach on Hierarchical Text Classification using Top-Level Pruning Classifiers” โดย “Natchanon Phachongkitphiphat” และ “Peerapon Vateekul” ในงานประชุมวิชาการนานาชาติ “The 11th International Joint Conference on Computer Science and Software Engineering (JCSSE)” ณ โรงแรมสยามเบย์ชอร์ เมืองพัทยา จังหวัดชลบุรี ประเทศไทย ระหว่างวันที่ 14 ถึงวันที่ 16 พฤษภาคม 2557

ดัชนีศัพท์

[A]

Ancestor Class / Ancestor Node	บรรพบุรุษ, โหนดบรรพบุรุษ

[B]

Big-bang Approach / Global Approach	วิธีการแบบบิ๊กแบง
Binary Classification	การจำแนกสองประเภท
Branch / Edge	กิ่งในโครงสร้างลำดับชั้น

[C]

Categorization / Classification	การจำแนกประเภท
Candidate Category / Candidate Class	คลาสที่มีโอกาสเป็นคำตอบ
Centroid Vector	เซนทรอยด์เวกเตอร์
Child Class / Child Node	โหนดลูก
Class / Label	ประเภท, ฉลาก
Classifier	ตัวจำแนกประเภท
Class Hierarchy	โครงสร้างลำดับชั้น
Class Selection	การเลือกประเภท
Cosine Similarity	การวัดความคล้ายคลึงเชิงมุมโคไซน์
Cycle	วัฏจักร

[D]

Default Predicted Class	ประเภทที่จำแนกได้โดยปริยาย

Default True Class	ประเภทที่เป็นคำตอบที่ถูกต้องโดยปริยาย
Depth	ความลึก
Development Set / Validation Data Set	ชุดข้อมูลตรวจสอบ
Directed Acyclic Graph (DAG)	กราฟอวัฏจักรระบุทิศทาง, โครงสร้างแบบกราฟ

[E]

Error Propagation	ความผิดพลาดที่ถูกส่งต่อจากโหนดแม่สู่โหนดลูก
Enzyme	เอนไซม์

[F]

Feature	ฟีเจอร์
Feature Selection	การเลือกฟีเจอร์
Flat Approach	วิธีการแบบแฟลต

[G]

Genre	แนวเพลง

[H]

Hierarchical Classification (HC)	การจำแนกประเภทแบบมีลำดับชั้น
Hierarchical Constraint	ข้อกำหนดเชิงลำดับชั้น

Hierarchical Multi-Label Classification (HMC)	การจำแนกประเภทแบบหลายฉลากมีลำดับชั้น

[I]

Imbalanced Training Sets	ชุดตัวอย่างสอนไม่สมดุล
Information Retrieval Systems	ระบบค้นคืนสารสนเทศ
Internal Node	โหนดภายใน

[K]

k-Nearest Neighbors (k-NN)	การจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด K อันดับ

[L]

Large-scale Data	ข้อมูลขนาดใหญ่
Leaf Node	โหนดใบ

[M]

Multiclass Classification	การจำแนกหลายประเภท
Multi-label Classification	การจำแนกประเภทแบบหลายฉลาก
Music Genre Classification	การจำแนกแนวเพลง

[N]

Negative Training Set	ชุดตัวอย่างลบ

[P]

Parent Class / Parent Node	โหนดแม่
Performance Evaluation	การวัดประสิทธิภาพการทำงาน
Positive Training Set	ชุดตัวอย่างบวก
Protein Function	ฟังก์ชันการทำงานของโปรตีน
Protein Function Prediction	การทำนายฟังก์ชันการทำงานของโปรตีน

[R]

Root Node	โหนดราก

[S]

Sibling Node / Sibling Class	โหนดพี่น้อง
Support Vector Machine (SVM)	ซัพพอร์ตเวกเตอร์แมชชีน

[T]

Term Frequency	ความถี่ของคำ
Test Data	ข้อมูลทดสอบ
Test Data Set	ชุดข้อมูลทดสอบ
Text Categorization	การจำแนกประเภทข้อความ
Top-down Approach / Local Approach	วิธีการแบบบนลงล่าง
Training Data Set	ชุดข้อมูลเรียนรู้
Training Data	ข้อมูลเรียนรู้
Training Policy	การกำหนดชุดข้อมูลเรียนรู้
Tree	โครงสร้างแบบต้นไม้

[W]

Web directory	เว็บไดเรกทอรี
Wikipedia	วิกิพีเดีย

รายการอ้างอิง

[1] *Categorization*. Available: https://en.wikipedia.org/wiki/Categorization

[2] *Classification*. Available: https://en.wikipedia.org/wiki/Classification

[3] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan, "Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies," *VLDB Journal,* vol. 7, pp. 163-178, 1998.

[4] C. N. Silla Jr and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery,* vol. 22, pp. 31-72, 2011.

[5] A. J. Barrett, "Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). Enzyme Nomenclature. Recommendations 1992. Supplement 4: corrections and additions (1997)," *Eur J Biochem,* vol. 250, pp. 1-6, Nov 15 1997.

[6] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry*, et al.*, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium," *Nat Genet,* vol. 25, pp. 25-9, May 2000.

[7] *Wikipedia*. Available: http://www.wikipedia.org/

[8] G. R. Xue, D. Xing, Q. Yang, and Y. Yu, "Deep classification in large-scale text hierarchies," Singapore, 2008, pp. 619-626.

[9] W. Bi and J. T. Kwok, "Multi-label classification on tree- and DAG-structured hierarchies," Bellevue, WA, 2011, pp. 17-24.

[10] *Pascal Challenge on Large Scale Hierarchical Text Classification*. Available: lshtc.iit.demokritos.gr/

[11] I. Partalas, A. Kosmopoulos, N. Baskiotis, T. Artieres, G. Paliouras, E. Gaussier*, et al.*, "Lshtc: A benchmark for large-scale text classification," *arXiv preprint arXiv:1503.08581,* 2015.

[12] Y. Sasaki and D. Weissenbacher, "TTI's System for the LSHTC3 Challenge," presented at the Third Pascal Challenge on Large Scale Hierarchical Text Classification, 2012.

[13] T. Fagni and F. Sebastiani, "On the selection of negative examples for hierarchical text categorization," *Proceedings of the 3rd Language & Technology Conference (LTC'07),* pp. 24-28, 2007.

[14] P. Vateekul, "Hierarchical Multi-Label Classification: Going Beyond Generalization Trees," *Open Access Dissertations,* p. Paper 723, 2012.

[15] *Precision and recall*. Available: http://en.wikipedia.org/wiki/Precision_and_recall

[16] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," *Inf. Retr.,* vol. 1, pp. 69-90, 1999.

[17] S. Kiritchenko, S. Matwin, and F. Famili, "Functional annotation of genes using hierarchical text categorization," in *BioLINK SIG: Linking Literature, Information and Knowledge for Biology*, 2005.

[18] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos, "Evaluation Measures for Hierarchical Classification: a unified view and novel approaches," *CoRR,* vol. abs/1306.6802, 2013.

[19] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "On finding lowest common ancestors in trees," presented at the Proceedings of the fifth annual ACM symposium on Theory of computing, Austin, Texas, USA, 1973.

[20] *kNN Benchmark Code*. Available: https://www.kaggle.com/c/lshtc/forums/t/6974/knn-benchmark-code

[21] X. Wang, H. Zhao, and B.-L. Lu, "Enhance Top-down method with Meta-Classification for Very Large-scale Hierarchical Classification," in *IJCNLP*, 2011, pp. 1089-1097.

[22] T. Joachims, "11 Making Large-Scale SVM Learning Practical."

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of machine learning research,* vol. 9, pp. 1871-1874, 2008.

[24] X. Han, S. Li, and Z. Shen, "A k-NN Method for Large Scale Hierarchical Text Classification at LSHTC3," presented at the Third Pascal Challenge on Large Scale Hierarchical Text Classification, 2012.

[25] D.-H. Lee, "Multi-Stage Rocchio Classification for Large-scale Multilabeled Text data," presented at the Third Pascal Challenge on Large Scale Hierarchical Text Classification, 2012.

[26] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.,* vol. 2, pp. 1-27, 2011.

[27] N. Phachongkitphiphat and P. Vateekul, "An Improvement of Flat Approach on Hierarchical Text Classification Using Top-Level Pruning Classifiers," in *The 11th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Chonburi, Thailand, 2014, pp. 86-90.

ภาคผนวก

ภาคผนวก ก
ผลการวัดประสิทธิภาพการทำนายคลาสด้วยวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด k อันดับที่ปรับปรุงการจัดอันดับคลาสคำตอบ

ผู้วิจัยทำการทดลองบนข้อมูลวิกิพีเดียขนาดกลางตามบทที่ 3 หัวข้อที่ 3.5 โดยกำหนดค่าเพื่อนบ้านใกล้สุด k เท่ากับ 5 และ 7 และจำนวนคลาสคำตอบ x เท่ากับ 1 ถึง k

เมื่อกำหนดค่า k เท่ากับ 5 จะได้ผลการวัดประสิทธิภาพฯ ดังตารางที่ 8

ตารางที่ 8 ผลการวัดประสิทธิภาพฯ เมื่อกำหนดค่า k เท่ากับ 5

x	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
1	30.06	46.20	30.06	34.48	37.35	15.77	15.96	46.20	23.77	31.39
2	29.52	35.80	40.34	35.94	27.58	23.42	19.90	34.30	34.42	34.36
3	26.56	30.02	44.76	33.86	22.41	27.62	20.27	26.93	39.34	31.97
4	24.51	26.84	47.38	32.03	19.28	30.15	19.67	22.39	42.21	29.25
5	23.29	25.07	49.21	30.80	17.28	31.89	19.01	19.52	44.26	27.09
1+	31.93	43.82	39.64	36.87	16.20	21.61	15.01	18.92	33.64	24.22
2+	25.77	29.27	49.48	32.36	14.23	30.79	16.61	15.21	44.04	22.61
3+	22.64	24.31	52.35	29.74	13.71	34.15	16.89	14.23	47.75	21.93
4+	21.77	22.96	53.26	28.97	13.56	35.40	16.98	13.95	49.22	21.74
5+	21.55	22.61	53.53	28.76	13.52	35.82	17.01	13.85	49.71	21.67

เมื่อกำหนดค่า k เท่ากับ 7 จะได้ผลการวัดประสิทธิภาพฯ ดังตารางที่ 9

ตารางที่ 9 ผลการวัดประสิทธิภาพฯ เมื่อกำหนดค่า k เท่ากับ 7

x	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
1	30.34	46.77	30.34	34.83	38.88	14.84	15.06	46.77	24.06	31.78
2	29.32	35.81	41.05	36.15	28.30	22.88	19.45	34.70	35.06	34.88
3	25.86	29.37	45.66	33.63	22.47	27.42	19.96	27.09	40.20	32.36
4	23.30	25.59	48.34	31.27	19.06	30.28	19.43	22.29	43.17	29.40
5	21.61	23.27	50.16	29.48	16.73	32.23	18.61	19.12	45.23	26.88
6	20.48	21.78	51.61	28.19	15.14	33.70	17.86	16.91	46.81	24.85
7	19.72	20.80	52.72	27.26	14.02	34.82	17.24	15.32	48.06	23.23
1+	31.79	44.66	38.07	36.64	14.99	19.60	13.58	18.16	31.89	23.14
2+	25.12	29.12	49.48	31.73	12.19	29.86	14.79	13.21	43.66	20.29
3+	20.61	22.29	53.95	27.62	11.34	34.90	14.89	11.72	48.98	18.92
4+	18.96	19.94	55.59	26.00	11.04	37.09	14.90	11.22	51.28	18.42
5+	18.40	19.15	56.23	25.42	10.97	38.08	14.94	11.03	52.32	18.22
6+	18.20	18.86	56.48	25.21	10.93	38.50	14.95	10.95	52.78	18.14
7+	18.13	18.75	56.57	25.13	10.93	38.68	14.96	10.91	52.99	18.10

ภาคผนวก ข
ผลการวัดประสิทธิภาพการทำนายคลาสด้วยวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุด k อันดับที่วัดความคล้ายกับเซนทรอยด์เวกเตอร์

ผู้วิจัยทำการทดลองบนข้อมูลวิกิพีเดียขนาดกลางตามบทที่ 3 หัวข้อที่ 3.5 โดยกำหนดค่าเพื่อนบ้านใกล้สุด k เท่ากับ 5 7 และ 10 จำนวนคลาสคำตอบ x เท่ากับ 1 ถึง k และเลือกใช้เซนทรอยด์เวกเตอร์ 2 รูปแบบที่คำนวณไว้ตามบทที่ 3 หัวข้อที่ 3.2

เมื่อกำหนดค่า k เท่ากับ 5 และวัดความคล้ายกับ Normal Centroid จะได้ผลการวัดประสิทธิภาพฯ ดังตารางที่ 10

ตารางที่ 10 ผลการวัดประสิทธิภาพฯ เมื่อกำหนดค่า k เท่ากับ 5 และวัดความคล้ายกับ Normal Centroid

x	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
1	32.85	50.79	32.85	37.73	47.77	21.80	21.93	50.79	26.12	34.50
2	32.00	39.22	44.34	39.35	34.55	28.63	24.42	37.80	37.94	37.87
3	28.47	32.44	48.78	36.65	26.79	31.79	23.56	29.48	43.07	35.00
4	25.93	28.56	50.92	34.16	22.03	33.46	22.07	24.26	45.74	31.70
5	24.28	26.23	52.05	32.32	19.01	34.38	20.71	20.83	47.25	28.92
1+	32.86	50.78	32.87	37.75	47.76	21.87	21.99	50.78	26.15	34.52
2+	32.00	39.22	44.35	39.35	34.53	28.65	24.43	37.79	37.94	37.86
3+	28.47	32.43	48.78	36.65	26.77	31.80	23.57	29.47	43.08	35.00
4+	25.93	28.56	50.92	34.16	22.02	33.47	22.07	24.26	45.74	31.70
5+	24.28	26.23	52.05	32.32	19.00	34.39	20.70	20.83	47.25	28.92

เมื่อกำหนดค่า k เท่ากับ 5 และวัดความคล้ายกับ Decreased Centroid จะได้ผลการวัดประสิทธิภาพ ดังตารางที่ 11

ตารางที่ 11 ผลการวัดประสิทธิภาพฯ เมื่อกำหนดค่า k เท่ากับ 5 และวัดความคล้ายกับ Decreased Centroid

x	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
1	32.67	50.42	32.67	37.51	47.88	22.25	22.42	50.42	25.94	34.25
2	31.83	39.01	44.13	39.16	35.04	28.70	24.52	37.59	37.72	37.65
3	28.38	32.34	48.65	36.55	27.53	31.64	23.60	29.38	42.92	34.88
4	25.88	28.51	50.83	34.10	22.76	33.25	22.13	24.20	45.63	31.63
5	24.25	26.20	51.99	32.28	19.63	34.23	20.79	20.80	47.18	28.87
1+	32.69	50.42	32.69	37.52	47.86	22.31	22.48	50.41	25.96	34.27
2+	31.83	39.01	44.14	39.16	35.03	28.73	24.54	37.58	37.73	37.65
3+	28.38	32.34	48.65	36.55	27.51	31.67	23.61	29.37	42.93	34.88
4+	25.88	28.51	50.83	34.10	22.76	33.26	22.13	24.20	45.64	31.63
5+	24.25	26.20	51.99	32.28	19.61	34.23	20.79	20.80	47.18	28.87

เมื่อกำหนดค่า k เท่ากับ 7 และวัดความคล้ายกับ Normal Centroid จะได้ผลการวัดประสิทธิภาพ ดังตารางที่ 12

ตารางที่ 12 ผลการวัดประสิทธิภาพฯ เมื่อกำหนดค่า k เท่ากับ 7 และวัดความคล้ายกับ Normal Centroid

x	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
1	32.90	50.97	32.90	37.82	48.61	22.15	22.21	50.97	26.22	34.63
2	31.64	38.94	44.78	39.31	35.37	29.30	24.84	37.89	38.28	38.08
3	27.74	31.73	49.71	36.39	27.31	32.80	24.00	29.55	43.85	35.31
4	24.84	27.43	52.31	33.60	22.38	34.78	22.50	24.25	46.95	31.98
5	22.82	24.67	53.84	31.36	19.05	36.03	21.01	20.64	48.82	29.01
6	21.42	22.84	54.77	29.66	16.73	36.84	19.69	18.09	50.08	26.58
7	20.44	21.60	55.40	28.41	15.12	37.44	18.61	16.24	50.96	24.63
1+	32.92	50.97	32.92	37.83	48.59	22.23	22.28	50.96	26.24	34.65
2+	31.64	38.93	44.79	39.31	35.35	29.32	24.86	37.88	38.29	38.08
3+	27.73	31.73	49.71	36.39	27.28	32.81	24.00	29.54	43.85	35.30
4+	24.84	27.43	52.31	33.60	22.37	34.79	22.50	24.24	46.95	31.98
5+	22.82	24.67	53.85	31.36	19.04	36.03	21.01	20.64	48.83	29.01
6+	21.42	22.84	54.77	29.66	16.73	36.84	19.69	18.09	50.08	26.58
7+	20.44	21.60	55.40	28.41	15.11	37.45	18.61	16.24	50.96	24.63

เมื่อกำหนดค่า k เท่ากับ 7 และวัดความคล้ายกับ Decreased Centroid จะได้ผลการวัดประสิทธิภาพ ดังตารางที่ 13

ตารางที่ 13 ผลการวัดประสิทธิภาพฯ เมื่อกำหนดค่า k เท่ากับ 7 และวัดความคล้ายกับ Decreased Centroid

x	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
1	32.62	50.47	32.62	37.48	48.19	22.71	22.79	50.47	25.96	34.28
2	31.40	38.64	44.49	39.04	35.49	29.45	24.94	37.59	37.98	37.78
3	27.60	31.58	49.50	36.24	27.96	32.72	24.05	29.39	43.62	35.12
4	24.77	27.35	52.18	33.50	23.17	34.67	22.62	24.16	46.78	31.86
5	22.78	24.63	53.76	31.31	19.81	35.89	21.16	20.60	48.72	28.95
6	21.39	22.81	54.70	29.63	17.43	36.68	19.85	18.06	49.99	26.54
7	20.41	21.58	55.33	28.38	15.74	37.29	18.81	16.21	50.86	24.59
1+	32.64	50.46	32.64	37.50	48.16	22.77	22.84	50.46	25.98	34.30
2+	31.40	38.64	44.50	39.04	35.49	29.47	24.96	37.57	37.98	37.78
3+	27.60	31.58	49.51	36.24	27.94	32.74	24.06	29.39	43.63	35.12
4+	24.77	27.34	52.18	33.50	23.17	34.68	22.62	24.16	46.78	31.86
5+	22.78	24.63	53.76	31.31	19.80	35.90	21.15	20.59	48.72	28.95
6+	21.39	22.81	54.70	29.63	17.42	36.69	19.85	18.06	50.00	26.53
7+	20.41	21.57	55.33	28.37	15.73	37.29	18.80	16.21	50.86	24.59

เมื่อกำหนดค่า k เท่ากับ 10 และวัดความคล้ายกับ Normal Centroid จะได้ผลการวัดประสิทธิภาพ ดังตารางที่ 14

ตารางที่ 14 ผลการวัดประสิทธิภาพฯ เมื่อกำหนดค่า k เท่ากับ 10 และวัดความคล้ายกับ Normal Centroid

x	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
1	32.86	50.97	32.86	37.79	49.19	22.46	22.42	50.97	26.22	34.63
2	31.22	38.60	45.03	39.16	35.85	29.89	25.07	37.85	38.44	38.14
3	26.97	31.03	50.27	36.03	27.81	33.65	24.30	29.48	44.27	35.39
4	23.82	26.42	53.27	33.01	22.74	35.92	22.82	24.18	47.72	32.10
5	21.56	23.40	55.19	30.54	19.30	37.39	21.31	20.56	49.97	29.13
6	19.91	21.29	56.44	28.56	16.82	38.44	19.91	17.93	51.52	26.61
7	18.69	19.78	57.34	27.00	14.97	39.23	18.67	15.97	52.67	24.51
8	17.77	18.67	57.99	25.76	13.57	39.86	17.62	14.46	53.54	22.76
9	17.07	17.82	58.46	24.78	12.48	40.32	16.73	13.26	54.20	21.30
10	16.52	17.17	58.78	23.99	11.62	40.66	15.96	12.29	54.67	20.07
1+	32.88	50.97	32.88	37.80	49.15	22.54	22.49	50.95	26.24	34.64
2+	31.22	38.59	45.04	39.16	35.83	29.92	25.08	37.83	38.45	38.14
3+	26.97	31.03	50.27	36.02	27.78	33.65	24.30	29.47	44.27	35.39
4+	23.82	26.42	53.28	33.01	22.73	35.93	22.83	24.18	47.73	32.10
5+	21.56	23.40	55.20	30.54	19.28	37.40	21.30	20.55	49.97	29.13
6+	19.91	21.29	56.44	28.55	16.81	38.44	19.91	17.93	51.53	26.60
7+	18.69	19.78	57.34	27.00	14.96	39.23	18.67	15.97	52.68	24.51
8+	17.77	18.67	57.99	25.76	13.57	39.87	17.63	14.45	53.54	22.76
9+	17.07	17.82	58.46	24.78	12.48	40.33	16.73	13.26	54.20	21.30
10+	16.52	17.17	58.78	23.99	11.62	40.66	15.96	12.29	54.67	20.07

เมื่อกำหนดค่า k เท่ากับ 10 และวัดความคล้ายกับ Decreased Centroid จะได้ผลการวัดประสิทธิภาพ ดังตารางที่ 15

ตารางที่ 15 ผลการวัดประสิทธิภาพฯ เมื่อกำหนดค่า k เท่ากับ 10 และวัดความคล้ายกับ Decreased Centroid

x	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
1	32.48	50.31	32.48	37.33	48.34	23.13	23.08	50.31	25.88	34.18
2	30.87	38.19	44.59	38.77	35.55	30.13	25.20	37.44	38.03	37.73
3	26.80	30.84	50.02	35.82	28.12	33.69	24.34	29.28	43.97	35.15
4	23.70	26.29	53.04	32.86	23.32	35.82	22.87	24.04	47.45	31.92
5	21.49	23.33	55.02	30.45	19.97	37.26	21.40	20.48	49.78	29.02
6	19.86	21.24	56.31	28.49	17.50	38.25	20.03	17.88	51.36	26.52
7	18.65	19.74	57.24	26.94	15.62	38.96	18.83	15.93	52.52	24.44
8	17.74	18.63	57.89	25.71	14.22	39.60	17.84	14.41	53.39	22.70
9	17.04	17.79	58.37	24.74	13.05	40.09	16.95	13.23	54.07	21.25
10	16.50	17.15	58.71	23.96	12.13	40.43	16.18	12.27	54.57	20.04
1+	32.49	50.31	32.50	37.34	48.28	23.20	23.13	50.29	25.90	34.19
2+	30.87	38.19	44.60	38.77	35.54	30.16	25.23	37.43	38.04	37.73
3+	26.80	30.83	50.02	35.82	28.10	33.72	24.35	29.27	43.98	35.15
4+	23.70	26.29	53.05	32.86	23.30	35.83	22.87	24.04	47.46	31.91
5+	21.49	23.32	55.02	30.45	19.95	37.26	21.39	20.48	49.78	29.02
6+	19.86	21.24	56.31	28.49	17.49	38.25	20.03	17.87	51.36	26.52
7+	18.65	19.74	57.24	26.94	15.62	38.96	18.82	15.92	52.52	24.44
8+	17.74	18.63	57.90	25.71	14.22	39.61	17.85	14.41	53.39	22.70
9+	17.04	17.79	58.37	24.74	13.05	40.09	16.95	13.22	54.08	21.25
10+	16.50	17.15	58.71	23.96	12.13	40.43	16.18	12.27	54.57	20.03

ภาคผนวก ค

ผลการวัดประสิทธิภาพการทำนายคลาสบนข้อมูลวิกิพีเดียขนาดกลาง เปรียบเทียบวิธีการแบบแฟลต วิธี k-NN อัลกอริทึมของผู้เข้าแข่งขัน LSHTC และวิธีที่นำเสนอ

ผลการประเมินประสิทธิภาพฯ บนข้อมูลวิกิพีเดียขนาดกลาง เรียงลำดับวิธีการด้วยค่า LBMaF จากมากไปน้อย แสดงได้ดังตารางที่ 16 และตารางที่ 17

ตารางที่ 16 ผลการประเมินประสิทธิภาพฯ บนข้อมูลวิกิพีเดียขนาดกลาง เรียงลำดับวิธีการด้วยค่า LBMaF จากมากไปน้อย (ส่วนที่ 1)

ชื่อผู้เข้าร่วมการแข่งขัน LSHTC / อัลกอริทึม	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
TTI	42.00	50.48	50.84	47.71	50.64	30.69	28.35	47.69	46.82	47.25
dhlee	38.48	49.36	42.56	43.52	47.58	31.80	28.24	48.51	37.17	42.09
Arthur	43.82	55.16	49.63	49.37	57.33	28.76	26.74	56.58	43.82	49.39
วิธีที่นำเสนอ	30.84	38.49	42.51	37.93	39.91	29.42	25.70	38.52	36.42	37.44
coolvegpuff	42.91	55.00	47.63	48.24	52.61	25.67	25.07	55.24	42.12	47.79
chrishan	41.17	51.76	51.16	47.68	42.62	29.96	24.54	39.37	44.71	41.87
brouardc	35.36	47.87	43.23	41.82	35.82	26.52	23.98	37.93	36.87	37.39
Anttip	40.77	50.38	43.26	44.60	48.90	24.47	23.85	51.15	37.23	43.09
szarak	37.11	46.61	48.57	43.66	35.22	27.56	21.95	33.95	42.37	37.69

ตารางที่ 17 ผลการประเมินประสิทธิภาพฯ บนข้อมูลวิกิพีเดียขนาดกลาง เรียงลำดับวิธีการด้วยค่า LBMaF จากมากไปน้อย (ส่วนที่ 2)

ผลลัพธ์ของการจำแนกประเภทหลายป้ายกำกับ LSHTC / อัลกอริทึม	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
daq	35.26	45.40	36.82	38.96	39.98	20.29	19.29	43.37	31.55	36.53
KULeuven	29.76	37.10	36.73	34.08	27.88	20.16	18.25	27.92	33.49	30.45
k-NN Baseline	24.91	28.30	41.64	31.76	25.22	23.54	17.58	25.09	36.65	29.79
glouppe	4.73	5.02	34.23	8.46	32.39	15.96	17.58	5.74	31.43	9.71
SSir	32.70	43.39	39.79	38.73	46.13	16.11	16.81	44.06	34.65	38.79
TUD_KE	24.48	32.30	29.52	28.39	24.99	13.11	13.03	29.49	25.97	27.62
hautcs	32.04	40.21	32.89	34.73	53.96	10.70	10.37	43.26	26.34	32.74
วิธีการแบบแฟลต	20.34	26.70	21.93	22.72	45.89	7.57	8.89	30.01	19.03	23.29
Peaceguard	24.99	40.48	24.99	29.17	47.01	5.39	5.55	40.48	20.82	27.50
dicaro	6.26	8.29	10.62	8.05	14.26	2.51	2.11	5.67	9.72	7.16

ภาคผนวก ง
ผลการวัดประสิทธิภาพการทำนายคลาสบนข้อมูลวิกิพีเดียขนาดใหญ่ เปรียบเทียบวิธีการแบบแฟลต วิธี k-NN อัลกอริทึมของผู้เข้าแข่งขัน LSHTC และวิธีที่นำเสนอ

ผลการประเมินประสิทธิภาพฯ บนข้อมูลวิกิพีเดียขนาดใหญ่ เรียงลำดับวิธีการด้วยค่า LBMaF จากมากไปน้อย แสดงได้ดังตารางที่ 18

ตารางที่ 18 ผลการประเมินประสิทธิภาพฯ บนข้อมูลวิกิพีเดียขนาดใหญ่ เรียงลำดับวิธีการด้วยค่า LBMaF จากมากไปน้อย

ชื่อวิธีการ / อัลกอริทึมของผู้เข้าแข่งขัน LSHTC	%									
	Acc	EBP	EBR	EBF	LBMaP	LBMaR	LBMaF	LBMiP	LBMiR	LBMiF
dhlee	34.02	47.84	40.60	41.49	40.86	30.78	25.64	41.55	29.53	34.52
วิธีที่นำเสนอ	14.55	18.19	42.21	20.78	33.35	28.16	23.48	4.73	29.64	8.16
anttip	33.23	42.33	40.17	39.10	33.31	24.55	22.45	34.38	30.62	32.40
coolvegpuff	38.06	57.10	46.25	45.91	32.49	24.32	21.74	29.35	35.57	32.16
daq	33.26	49.40	40.08	40.07	32.09	20.31	17.38	32.86	29.28	30.96
chrishan	34.58	60.70	36.70	41.98	53.79	17.60	17.01	55.14	25.08	34.48
k-NN Baseline	27.24	36.28	38.69	34.72	30.33	17.70	14.86	32.56	28.08	30.16
วิธีการแบบแฟลต[13]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

[13] ใช้เวลาสร้างตัวจำแนกนานมาก ไม่อาจสร้างได้ครบในเวลาที่จำกัด จึงไม่มีผลประเมินประสิทธิภาพ

ประวัติผู้เขียนวิทยานิพนธ์

นายณัฐชนน ผจงกิจพิพัฒน์ เกิดวันที่ 16 กันยายน 2534 สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2555 และเข้าศึกษาหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2556