การจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน

Web Page Classification Using Feature Reduction And Support Vector Machine

พรพล ธรรมรงค์รัตน์

Pornpon Thamrongrat

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Prince of Songkla University
2552

ชื่อวิทยานิพนธ์ การจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน

ผู้เขียน นายพรพล ธรรมรงค์รัตน์

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

…….....

(ผู้ช่วยศาสตราจารย์ ดร.วิภาดา เวทย์ประสิทธิ์)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

…….....

(ดร.ลัดดา ปรีชาวีรกุล)

คณะกรรมการสอบ

……..ประธานกรรมการ

(ดร.ฐิมาพร เพชรแก้ว)

…….....กรรมการ

(ดร.ลัดดา ปรีชาวีรกุล)

…….....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศิริรัตน์ วณิชโยบล)

…….....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วิภาดา เวทย์ประสิทธิ์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

.....

(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ การจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน
ผู้เขียน นายพรพล ธรรมรงค์รัตน์
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2551

บทคัดย่อ

จำนวนเว็บเพจที่มากขึ้นบนโลกอินเทอร์เน็ตส่งผลให้การสืบค้นเอกสารที่ตรงกับความต้องการของผู้ใช้มีประสิทธิภาพลดลง การจำแนกเว็บเพจเป็นแนวทางหนึ่งในการแก้ปัญหานี้ วิทยานิพนธ์นี้ได้นำเสนอแนวคิดใหม่ในการจำแนกประเภทเว็บเพจ โดยสร้างแบบจำลองการจำแนกประเภทเว็บเพจโดยการให้คะแนนเสียงและซัพพอร์ตเวกเตอร์แมชชีน (Web Page Classification Using Voting Algorithm and Multi-Class Support Vector Machine: WPC_VAMSVM) และพัฒนาโปรแกรมสำหรับจำแนกประเภทเว็บเพจโดยใช้ Visual C#.Net สำหรับเตรียมข้อมูลเว็บเพจ ซึ่งสร้างลักษณะเฉพาะ ทำงานร่วมกับโปรแกรม WEKA แบบ Command Line Interface สำหรับลดขนาดลักษณะเฉพาะ และโปรแกรม MATLAB สำหรับจำแนกประเภทและให้คะแนนเสียงเว็บเพจ ขั้นตอนการทำงานประกอบด้วย 4 ขั้นตอน คือ 1) การเตรียมข้อมูลเว็บเพจ 2) การสร้างลักษณะเฉพาะ 3) การลดขนาดลักษณะเฉพาะ 4) การจำแนกประเภทและการให้คะแนนเสียง ผลการทดลองโดยใช้ชุดข้อมูลมาตรฐานจาก CMU พบว่าการใช้ลักษณะเฉพาะจากข้อความและหัวเรื่องร่วมกับอัลกอริทึมการให้คะแนนเสียงแบบ 1vsAll_Voting ให้ค่าความถูกต้องสูงที่สุด

Thesis Title	Web Page Classification Using Feature Reduction And Support Vector Machine
Author	Mr. Pornpon Thamrongrat
Major Program	Computer Science
Academic Year	2008

ABSTRACT

The increasing numbers of web pages on the cyber world result to the less effectiveness of document retrieval that matches the need of users. The classification of web pages is one of the solutions to solve this problem. This thesis proposes new idea for the web page classification by using feature reduction and Support Vector Machine. The model of Web Page Classification Using Voting Algorithm and Multi-Class Support Vector Machine (WPC_VAMSVM) has been constructed and developed for the web page classification. Visual C#.Net is used for programming in order to web page preprocessing process, Command Line Interface of WEKA is used for feature reduction process, and MATLAB is used for classification and voting process. The model of web page classification composes of 4 steps; step 1) web page preprocessing, step 2) feature generation, step 3) feature selection, and step 4) classification and new voting technique. The experimental result with the CMU benchmark dataset show that using text and title feature with 1vsAll_Voting algorithm gives the highest F-measure value.

สารบัญ

สารบัญ (ต่อ)

หน้า

รายการตาราง

รายการภาพประกอบ

รายการภาพประกอบ (ต่อ)

รายการภาพประกอบ (ต่อ)

บทที่ 1

บทนำ

เทคโนโลยีทางด้านอินเทอร์เน็ตเติบโตอย่างรวดเร็ว เว็บเพจถูกสร้างขึ้นจำนวนมากเพื่อตอบสนองวัตถุประสงค์ที่แตกต่างกัน ส่งผลให้เกิดข้อมูลสารสนเทศที่หลากหลายและมีปริมาณมากตามไปด้วย ทำให้เกิดคำถามที่ว่าเราจะสามารถหาสารสนเทศที่เป็นประโยชน์จากเว็บเพจมากมายเหล่านี้ได้อย่างไร คำถามนี้เป็นปัญหาที่ได้รับการถกเถียงกันในวงกว้าง โดยพบว่าวิธีหนึ่งที่จะตอบปัญหานี้คือ การจำแนกประเภทของเว็บเพจให้มีความถูกต้อง (Chen and Huang, 2004)

การจำแนกประเภทของเว็บเพจเป็นแนวทางหนึ่งที่นำมาช่วย เพื่อให้เกิดประโยชน์สูงสุดสำหรับข้อมูลที่มีอยู่บนอินเทอร์เน็ต ซึ่งหากมีการจำแนกประเภทของข้อมูลและจัดกลุ่มเว็บเพจไว้แล้วแบบอัตโนมัติ จะทำให้เราสามารถสืบค้นข้อมูลจากเว็บเพจได้ถูกต้องและรวดเร็วยิ่งขึ้น

อย่างไรก็ตาม ถึงแม้ว่าการจำแนกประเภทเว็บเพจได้ให้ประสิทธิภาพที่ดี แต่เว็บเพจประกอบด้วยตัวอักษรซึ่งมีความยาวที่แตกต่างกันจำนวนมาก ส่งผลให้ขนาดของลักษณะเฉพาะของข้อมูลเข้า (Input Feature) มีขนาดใหญ่ตามไปด้วย สิ่งนี้เป็นอุปสรรคในการจำแนกประเภท การลดขนาดลักษณะเฉพาะของข้อมูลเข้า (Feature Reduction) จึงเป็นแนวทางหนึ่งในการแก้ปัญหาดังกล่าว โดยมีจุดมุ่งหมายเพื่อลดความซับซ้อนของระบบและเพิ่มประสิทธิภาพของการจำแนกประเภท

การทำให้ค่าน้ำหนักคำ (Word Weighting) เป็นการสร้างตัวแทนเนื้อหาของเอกสาร เพื่อให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถเรียนรู้ได้ ซึ่งตัวแทนของเอกสารมักจะอยู่ในรูปของเวกเตอร์ของน้ำหนักคำ เช่น Boolean, TF-IDF และ TFC เป็นต้น

เว็บเพจที่มีจำนวนลักษณะเฉพาะ (Feature) จำนวนมาก จะส่งผลต่อประสิทธิภาพของการจำแนกหมวดหมู่เว็บเพจ เนื่องจากอัลกอริทึมที่ใช้ในการเรียนรู้เพื่อสร้างตัวจำแนกหมวดหมู่โดยทั่วไปไม่สามารถรองรับการทำงานกับจำนวนลักษณะเฉพาะของเอกสารที่สูงมากได้ดี การลดขนาดลักษณะเฉพาะจึงเป็นขั้นตอนหนึ่งที่ต้องทำก่อนการสร้างตัวจำแนกเอกสาร หลักการของซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine: SVM) คือการสร้างสมการเส้นตรงเพื่อแบ่งเขตข้อมูล 2 กลุ่มออกจากกันโดย SVM จะพยายามสร้างเส้นแบ่งตรงกึ่งกลางระหว่างกลุ่มให้มีระยะห่างระหว่างขอบเขตของทั้งสองกลุ่มมากที่สุด SVM จะใช้ฟังก์ชั่นแม็ปสำหรับย้ายข้อมูลจาก Input Space ไปยัง Feature Space และสร้างฟังก์ชั่นวัดความคล้ายที่เรียกว่า Kernel Function บน Feature Space

งานวิจัยนี้ได้ใช้เทคนิคการลดขนาดลักษณะเฉพาะและใช้เทคนิคซัพพอร์ตเวกเตอร์แมชชีนร่วมกับอัลกอริทึมการให้คะแนนเสียงมาสร้างแบบจำลอง และพัฒนาโปรแกรมสำหรับจำแนกประเภทเว็บเพจ เพื่อให้ได้ผลการจำแนกประเภทที่ถูกต้องมากยิ่งขึ้น

1.1 การตรวจเอกสาร

เทคนิคที่ใช้ในการสร้างแบบจำลองสำหรับจำแนกประเภทเว็บเพจ คือ การให้น้ำหนักคำ (Word Weighting) เทคนิคการลดขนาดลักษณะเฉพาะ (Feature Reduction) ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine) และการให้คะแนนเสียง (Voting) ดังรายละเอียดต่อไปนี้

1.1.1 การให้น้ำหนักคำ (Word Weighting)

การให้น้ำหนักคำ (Word Weighting) เป็นการสร้างตัวแทนของเนื้อหาของเอกสาร สำหรับใช้ในกระบวนการเรียนรู้ ลักษณะของตัวแทนเอกสารอยู่ในรูปแบบเวกเตอร์ของน้ำหนักคำ ตัวอย่างวิธีการให้น้ำหนักคำ (วัลลภ, 2548) เช่น TF-IDF (Term Frequency-Inverse Document Frequency) เป็นวิธีที่คำนวณน้ำหนักจากความถี่ของการปรากฏของคำในเอกสาร และพิจารณาความถี่ของคำนั้น ๆ ที่ปรากฏในเอกสารอื่นร่วมด้วย โดยมีแนวคิดว่า คำที่ปรากฏในเอกสารน้อยฉบับ จะมีค่าน้ำหนักสูง ส่วนคำที่ปรากฏในเอกสารหลายฉบับจะมีค่าน้ำหนักต่ำ เนื่องจากเป็นคำที่ไม่แสดงถึงลักษณะเฉพาะของเอกสารนั้น TFC Weighting (Term Frequency -Cosine) เป็นวิธีคำนวณที่คล้ายกับ TF-IDF แต่จะมีการพิจารณาจำนวนคำที่แตกต่างกันในแต่ละเอกสารร่วมด้วย จึงนำค่า Cosine Normalization เพื่อปรับมาตรฐานให้ทุกเอกสารมีจำนวนคำเท่ากัน LTC Weighting (Log-TF-IDF-Cosine) เป็นวิธีที่คำนวณน้ำหนักที่คล้ายกับวิธี TFC แต่จะมีการพิจารณาความถี่ที่มีค่าต่างกันมาก โดยการใช้ฟังก์ชัน log เข้ามาเพื่อปรับความแตกต่าง Boolean Weighting เป็นวิธีที่คำนวณน้ำหนักการปรากฏของคำที่อยู่ในเอกสาร ถ้ามีคำที่พิจารณาปรากฏอยู่ในเอกสารตั้งแต่ 1 ครั้งขึ้นไป จะให้ค่าน้ำหนักเป็น 1 ถ้าคำดังกล่าว ไม่ปรากฏอยู่ในเอกสารจะให้ค่าน้ำหนักเป็น 0 ค่าน้ำหนักนี้เรียกอีกอย่างว่า ค่าคุณลักษณะความจริง (Boolean Feature) ซึ่งมีค่าเป็นไบนารี Word Frequency Weighting เป็นวิธีที่คำนวณน้ำหนักการปรากฏของคำที่พิจารณา ปรากฏอยู่ในเอกสารนั้น ๆ โดยตรง

1.1.2 การลดขนาดลักษณะเฉพาะ (Feature Reduction)

การลดขนาดลักษณะเฉพาะ (Feature Reduction) เป็นการลดขนาดของข้อมูลที่ไม่จำเป็นออกเพื่อเพิ่มความถูกต้องของการจำแนกประเภท เอกสารที่มีจำนวนลักษณะเฉพาะมาก จะส่งผลต่อประสิทธิภาพของการจำแนก เนื่องจากวิธีการที่ใช้ในการเรียนรู้เพื่อสร้างตัวจำแนกไม่สามารถรับรองการทำงานกับจำนวนลักษณะของเอกสารที่สูงมากได้ดี ตัวอย่าง

เทคนิคของการเลือกลักษณะเฉพาะ เช่น ReliefF ใช้การประเมินค่าความแตกต่างของลักษณะเฉพาะกับตัวอย่างใกล้เคียง (*K* nearest Neighbors) ในคลาสเดียวกันหรือต่างคลาสกันจำนวน *K* ตัว ถ้าค่าความแตกต่างเป็น 1 แสดงว่ามีความแตกต่างกันมาก ถ้าค่าที่ได้เป็น 0 จะมีความเหมือนกันมาก งานวิจัยที่ใช้หลักการนี้เช่น งานวิจัยด้านการทำเหมืองข้อมูล (Huang *et al*., 2004; Symeonidis *et al*., 2007; Xin J. *et al*., 2007) Information Gain ใช้การประเมินค่าของลักษณะเฉพาะโดยวัดจากค่า Information Gain ซึ่งเป็นวิธีที่ง่ายและรวดเร็ว งานวิจัยที่ใช้หลักการนี้ เช่น งานวิจัยด้านการทำเหมืองข้อมูลโดยฐานข้อมูล UCI ประกอบด้วยฐานข้อมูลย่อย 9 ฐานข้อมูล (Huang *et al*., 2004) Gain Ratio ใช้การประเมินค่าของลักษณะเฉพาะโดยวัดค่า Gain Ratio ให้กับคลาสนั้น ๆ ซึ่งเป็นวิธีที่ง่ายและรวดเร็ว งานวิจัยที่ใช้หลักการนี้ เช่น การระบุเสียงพูดใช้ฐานข้อมูลเสียง 2001 NIST SRE (Ganchev *et al*., 2006) Chi Square ใช้การประเมินค่าของลักษณะเฉพาะโดยวัดค่า Chi-Square ทางสถิติ งานวิจัยที่ใช้หลักการนี้ เช่น งานวิจัยด้านชีวสารสนเทศ (Koh and Wong, 2007)

1.1.3 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine)

ซัพพอร์ตเวกเตอร์แมชชีน (Gunn, S. R., 1998) ถูกออกแบบมาเพื่อแก้ปัญหาการจำแนกประเภทแบบไบนารี หลักการของ SVM คือการสร้างสมการเส้นตรงเพื่อแบ่งเขตข้อมูล 2 กลุ่มออกจากกันโดย SVM จะพยายามสร้างเส้นแบ่งตรงกึ่งกลางระหว่างกลุ่มให้มีระยะห่างระหว่างขอบเขตของทั้งสองกลุ่มมากที่สุด SVM จะใช้ฟังก์ชั่นแม็ปสำหรับย้ายข้อมูลจาก Input Space ไปยัง Feature Space และสร้างฟังก์ชั่นวัดความคล้ายที่เรียกว่า Kernel Function บน Feature Space งานวิจัยที่ใช้ซัพพอร์ตเวกเตอร์แมชชีนมาใช้ในการจำแนกประเภทเว็บเพจ เช่น การสกัดลักษณะเฉพาะจากเว็บเพจโดยใช้ SVM เพื่อประเมินประสิทธิผลของการจำแนกประเภทเว็บเพจ ด้วยเคอร์แนลฟังก์ชัน (Kernel Function) ที่ต่างกัน (Xue *et al*., 2006) Multi-Class Support Vector Machine เป็นวิธีการที่ใช้เพื่อแก้ปัญหาการจำแนกประเภทที่มีจำนวนคลาสมากกว่าสองได้ ซึ่งมีวิธีการดังนี้

- 1vsAll (One-against-all) เป็นวิธีการที่ง่ายที่สุดในการทำมัลติคลาส (Multi-Class) เมื่อทำการจำแนกจำนวน M คลาส แต่ละคลาสต้องทำการเปรียบเทียบค่า discriminant function ด้วยจำนวน M-1 คลาส โดยคลาสที่มีค่า discriminant function มากที่สุด จะถูกเลือกเป็นผลลัพธ์สำหรับการจำแนกประเภท

- 1vs1 (One-against-one) เป็นอีกวิธีการหนึ่งในการทำมัลติคลาส (Multi-Class) ที่ตั้งอยู่บนหลักการของไบนารี เมื่อทำการจำแนกจำนวน M คลาส ต้องทำการเปรียบเทียบค่า discriminant function ระหว่างหนึ่งคลาสต่อหนึ่งคลาส ด้วยจำนวน M(M-1)/2 คลาส โดยคลาสที่มีค่า discriminant function มากกว่า จะถูกเลือกเป็นผลลัพธ์ คลาสที่มีจำนวนการถูกเลือกมากที่สุดจะเป็นผลลัพธ์ของการจำแนกประเภท แต่หากมีคลาสที่เป็นผลลัพธ์

มากกว่า 1 คลาส จะพิจารณาจากผลรวมของค่าสัมประสิทธิ์ของ discriminant function ที่มากกว่า

- Directed Acyclic Graphs SVM (DAGSVM) เป็นวิธีการที่ใช้หลักการของ One-against-one แต่วิธีนี้จะใช้การเปรียบเทียบเพียง M-1 ครั้งเท่านั้น โดยมีโหนดภายในจำนวน M(M-1)/2 และโหนดใบ M จำนวน การเปรียบเทียบจะเปรียบเทียบระหว่างหนึ่งคลาสต่อหนึ่งคลาสโดยจะไม่สนใจคลาสอื่นที่เหลือ ผลลัพธ์ที่ได้จะดำเนินตามด้านซ้ายหรือขวาของโหนด ซึ่งขึ้นอยู่กับผลลัพธ์ของการเปรียบเทียบในครั้งนั้น ๆ เมื่อเส้นทางการเปรียบเทียบพบโหนดใบใด ก็จะถือว่าคลาสนั้นเป็นผลลัพธ์ของการจำแนกประเภท

1.1.4 การให้คะแนนเสียง (Voting)

การให้คะแนนเสียงเป็นวิธีการในการเพิ่มประสิทธิภาพของการจำแนกประเภท โดยนำผลลัพธ์ที่ได้จากการจำแนกประเภทก่อนหน้าซึ่งมาจาก 2 วิธีการหรือมากกว่านั้น มาให้คะแนนเสียงของผลลัพธ์แล้วเลือกคำตอบของการจำแนกประเภทตามวิธีการหรืออัลกอริทึมของวิธีการให้คะแนนเสียงที่นำเสนอ งานวิจัยที่ใช้หลักการนี้ เช่น มีการนำผลลัพธ์ที่ได้จากการจำแนกด้วย SVM โดยมีลักษณะเฉพาะสองชนิดคือ ลักษณะของคีย์เวิร์ด และลักษณะของข้อความ ผ่านอัลกอริทึมการให้คะแนนเสียงเพื่อเพิ่มประสิทธิภาพของการจำแนกประเภทของเว็บเพจ (Rung-Ching and Chung-Hsun, 2006) มีวิธีการให้คะแนนเสียง จากผลลัพธ์ที่ได้จากการจำแนกด้วย SVM โดยมีลักษณะเฉพาะที่แตกต่างกัน 5 ชนิดคือ 1) Text 2) Title+Heading 3) URL+Anchor 4) Title +Heading+URL+ Anchor+Text 5) Title+Heading+URL+Anchor โดยถือว่าถ้าการจำแนกทั้ง 5 วิธี ยอมรับว่าเป็นคลาสนั้น ก็จะให้ผลลัพธ์ของการจำแนกคือคลาสดังกล่าว (Rui F. *et. al*, 2006)

1.2 วัตถุประสงค์ของโครงการ

1.2.1 สร้างแบบจำลองสำหรับการจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน

1.2.2 พัฒนาโปรแกรมสำหรับการจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน

1.3 ขอบเขตการดำเนินงาน

1.3.1 พัฒนาอัลกอริทึมและสร้างแบบจำลองในการจำแนกประเภทเว็บเพจโดยการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน

1.3.2 พัฒนาโปรแกรมเพื่อจำแนกประเภทเว็บเพจโดยการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน

1.3.3 นำชุดข้อมูลมาทดสอบ ได้แก่

- http://www.yahoo.com
- http://www.cs.cmu.edu/~WebKB

1.3.4 ประเมินประสิทธิภาพของแบบจำลองด้วยค่า F-measure

1.4 ขั้นตอนการดำเนินงานและระยะเวลาการดำเนินงาน

1.4.1 ศึกษางานวิจัยและเอกสารที่เกี่ยวข้องสำหรับการจำแนกประเภทเว็บเพจ

1.4.2 ศึกษาเทคโนโลยีและเครื่องมือสนับสนุน

1.4.3 วิเคราะห์และออกแบบโปรแกรมสำหรับการจำแนกประเภทเว็บเพจโดยที่ขั้นตอนการจำแนกประเภทเว็บเพจประกอบด้วย 4 ขั้นตอนหลัก

1.4.4 เตรียมข้อมูลสำหรับนำมาทดสอบ ซึ่งเป็นข้อมูลที่ใช้ในการวัดประสิทธิภาพของโปรแกรมในการจำแนกประเภทเว็บเพจ

1.4.5 เลือกลักษณะเฉพาะที่สำคัญเพื่อลดจำนวนข้อมูลเข้าและเลือกลักษณะเฉพาะที่สำคัญสำหรับจำแนกประเภทเว็บเพจ

1.4.6 จำแนกประเภทเว็บเพจด้วยเครื่องมือที่ได้พัฒนาแล้ววิเคราะห์และประเมินประสิทธิภาพของเครื่องมือ

1.4.7 พัฒนาโปรแกรมสำหรับการจำแนกประเภทเว็บเพจ

1.4.8 ทดสอบและติดตั้งโปรแกรมสำหรับการจำแนกประเภทเว็บเพจ

1.4.9 จัดทำเอกสารประกอบโปรแกรมสำหรับการจำแนกประเภทเว็บเพจ

ตารางที่ 1.1 ตารางแสดงระยะเวลาการดำเนินงาน

กิจกรรม/ขั้นตอนการดำเนินงาน	เดือน												
	2551									2552			
	4	5	6	7	8	9	10	11	12	1	2	3	4
1. ศึกษาและทำความเข้าใจลักษณะของเว็บเพจและการลดขนาดลักษณะเฉพาะวิธีต่าง ๆ													

ตารางที่ 1.1 ตารางแสดงระยะเวลาการดำเนินงาน (ต่อ)

กิจกรรม/ขั้นตอนการดำเนินงาน	เดือน												
	2551									2552			
	4	5	6	7	8	9	10	11	12	1	2	3	4
2. ศึกษาเทคนิคการเตรียมข้อมูลจากเว็บเพจ และวิธีการจำแนกประเภท													
3. ศึกษาเทคโนโลยีและเครื่องมือสนับสนุน													
4. วิเคราะห์และออกแบบระบบ													
5. พัฒนาระบบ													
6. ทดสอบและติดตั้งระบบ													
7. จัดทำเอกสารประกอบระบบและเขียนผลงานวิจัย													
8. จัดทำเอกสารวิทยานิพนธ์													

1.5 สถานที่และเครื่องมือที่ใช้งานวิจัย

1.5.1 สถานที่ทำวิจัย

ห้องปฏิบัติการคอมพิวเตอร์ CS207 ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

1.5.2 เครื่องมือที่ใช้ทำวิจัย

1) ด้านฮาร์ดแวร์

- เครื่องคอมพิวเตอร์จำนวน 1 เครื่อง
- หน่วยความจำ 512 MB
- ฮาร์ดดิสก์ 80 GB
- เครื่องพิมพ์จำนวน 1 เครื่อง

2) ด้านซอฟต์แวร์

- โปรแกรมประยุกต์ WEKA 3-6
- โปรแกรมประยุกต์ MATLAB 7.0
- โปรแกรมประยุกต์ C#.Net
- โปรแกรม compile java (j2se)
- ระบบปฏิบัติการ Microsoft Windows XP

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้แบบจำลองสำหรับจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน

1.6.2 ได้โปรแกรมสำหรับจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน

บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับการจำแนกประเภทเว็บเพจ

ทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับการจำแนกประเภทเว็บประกอบด้วย การหารากศัพท์ของคำ การกำจัดคำหยุด การให้ค่าน้ำหนักคำ การลดขนาดลักษณะเฉพาะ ซัพพอร์ตเวกเตอร์แมชชีน ต้นไม้ตัดสินใจ C4.5 RBFNN และการประเมินประสิทธิภาพ

2.1 การหารากศัพท์ของคำ (Stemming)

รากศัพท์ คือรูปแบบเดิมของคำที่ยังไม่ได้เติมคำอุปสรรค (Prefixes) หรือคำปัจจัย (Suffixes) การหารากศัพท์ จึงเป็นการหารูปเดิมของคำ เพื่อปรับรวมให้เป็นคำเดียวกัน การหารากศัพท์เป็นกระบวนการที่ควรทำก่อนการนำคำเหล่านี้มาใช้เพื่อจำแนกประเภทเว็บเพจ การหารากศัพท์ของคำภาษาอังกฤษนี้มีขั้นตอนที่เป็นรูปแบบซึ่งสามารถเขียนเป็นอัลกอริทึมในการหารากศัพท์ได้ เนื่องจากไวยากรณ์ของภาษาอังกฤษมีกฎเกณฑ์ที่แน่นอน สำหรับงานวิจัยนี้ได้ใช้ Porter Algorithm (Porter M. F., 1997) ในการหารากศัพท์ ประกอบด้วย 5 ขั้นตอนดังภาพประกอบ 2.1

Step 1a

Suffix	Replacement	Examples
sses	ss	caresses --> caress
ies	i	ponies --> poni
ss	ss	carress --> carress
s	NULL	cats --> cat>

Step 1b

Suffix	Replacement	Examples
eed	ee	feed --> feed
ed	NULL	plastered --> plaster
		bled --> bled
ing	NULL	motoring --> motor

ภาพประกอบ 2.1 การหารากศัพท์ด้วยวิธี Porter

Step 1b1

Suffix	Replacement	Examples
at	ate	conflat(ed) --> conflate
bl	ble	troubl(ing)--> trouble
iz	ize	siz(ed) --> size
NULL	single letter	hopp(ing) --> hop
		tann(ed) --> tan
		fall(ing) --> fall
		hiss(ing) --> hiss
		fizz(ing) --> fizz
NULL	e	fail(ing) --> fail
		fil(ing) --> file

Step 1c

Suffix	Replacement	Examples
y	i	happy --> happi
		sky --> sky

Step 2

Suffix	Replacement	Examples
ational	ate	relational --> relate
tional	tion	conditional --> condition
enci	ence	valenci --> valence
anci	ance	hesitanci --> hesitance
izer	ize	digitizer --> digitize
abli	able	conformabli --> conformable
alli	al	radicalli --> radical
entli	ent	differentli --> different
eli	e	vileli --> vile
ousli	ous	analogousli --> analogous
ization	ize	vietnamization --> vietnamize
ation	ate	predication --> predicate

ภาพประกอบ 2.1 การหารากศัพท์ด้วยวิธี Porter (ต่อ)

ator	ate	operator --> operate
alism	al	feudalism --> feudal
iveness	ive	decisiveness --> decisive
fulness	ful	hopefulness --> hopeful
ousness	ous	callousness --> callous
aliti	al	formaliti --> formal
iviti	ive	sensitiviti --> sensitive
biliti	ble	sensibiliti --> sensible

Step 3

Suffix	Replacement	Examples
icate	ic	triplicate --> triplic
ative	NULL	formative --> form
alize	al	formalize --> formal
iciti	ic	electriciti --> electric
ical	ic	electrical --> electric
ful	NULL	hopeful --> hope
ness	NULL	goodness --> good

Step 4

Suffix	Replacement	Examples
al	NULL	revival --> reviv
ance	NULL	allowance --> allow
ence	NULL	inference --> infer
er	NULL	airliner --> airlin
ic	NULL	gyroscopic --> gyroscop
able	NULL	adjustable --> adjust
ible	NULL	defensible --> defens
ant	NULL	irritant --> irrit
ement	NULL	replacement --> replac
ment	NULL	adjustment --> adjust
ent	NULL	dependent --> depend
ion	NULL	adoption --> adopt

ภาพประกอบ 2.1 การหารากศัพท์ด้วยวิธี Porter (ต่อ)

ou	NULL	homologou --> homolog
ism	NULL	communism --> commun
ate	NULL	activate --> activ
iti	NULL	angulariti --> angular
ous	NULL	homologous --> homolog
ive	NULL	effective --> effect
ize	NULL	bowdlerize --> bowdler

Step 5a

Suffix	Replacement	Examples
e	NULL	probate --> probat
e	NULL	cease --> ceas

Step 5b

Suffix	Replacement	Examples
NULL	single letter	controll --> control
		roll --> roll

ภาพประกอบ 2.1 การหารากศัพท์ด้วยวิธี Porter (ต่อ)

2.2 การกำจัดคำหยุด (Stopping)

การกำจัดคำหยุด เป็นการนำคำที่ไม่มีนัยสำคัญออกไป โดยที่ไม่ทำให้ความหมายของเอกสารเปลี่ยนแปลง คำที่ไม่มีนัยสำคัญนี้ หมายถึง คำที่ใช้กันโดยทั่วไปไม่มีความหมายสำคัญต่อเอกสารนั้น ๆ เมื่อตัดออกจากเอกสารแล้วจะไม่ทำให้ใจความสำคัญของเอกสารเปลี่ยนแปลง ซึ่งได้แก่คำดังตารางที่ 2.1 (Frankes and Yates, 1992)

ตารางที่ 2.1 Stoplist Word

a	been	get	least	our	them	whether
about	before	getting	left	ourselves	then	which
after	being	go	less	out	there	while
again	between	goes	let	over	these	who
ago	but	going	like	per	they	whoever
all	by	gone	make	put	this	whom

ตารางที่ 2.1 Stoplist Word (ต่อ)

almost	came	got	many	putting	those	whose
also	can	gotten	may	same	through	why
always	cannot	had	maybe	saw	till	will
am	come	has	me	see	to	with
an	could	have	mine	seen	too	within
and	did	having	more	shall	two	without
another	do	he	most	she	unless	won't
any	does	her	much	should	until	would
anybody	doing	here	my	so	up	wouldn't
anyhow	done	him	myself	some	upon	yet
anyone	down	his	never	somebody	us	you
anything	each	how	no	someone	very	your
anyway	else	i	none	something	was	
are	even	if	not	stand	we	
as	ever	in	now	such	went	
at	every	into	of	sure	were	
away	everyone	is	off	take	what	
back	everything	isn't	on	than	whatever	
be	for	it	one	that	what's	
became	from	just	onto	the	when	
because	front	last	or	their	where	

2.3 การให้น้ำหนักคำ (Word Weighting)

การให้น้ำหนักคำเป็นขั้นตอนการแปลงเอกสารเพื่อให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถเรียนรู้ได้ การกระทำนี้จะเป็นการสร้างตัวแทนเนื้อหาของเอกสาร สำหรับใช้ในกระบวนการเรียนรู้ สำหรับวิทยานิพนธ์นี้ได้ใช้วิธีการ TF-IDF ซึ่งเป็นวิธีการที่นิยมใช้กันอย่างแพร่หลาย TF-IDF (Joachims T., 1997) เป็นวิธีที่คำนวณน้ำหนักจากความถี่ของการปรากฏของคำ t_k ในเอกสาร d_j และพิจารณาความถี่ของคำ t_k ที่ปรากฏในเอกสารอื่นร่วมด้วย

โดยมีแนวคิดว่า คำที่ปรากฏในเอกสารน้อยฉบับ จะมีค่าน้ำหนักสูง ส่วนคำที่ปรากฏในเอกสารหลายฉบับจะมีค่าน้ำหนักต่ำ เนื่องจากเป็นคำที่ไม่แสดงถึงลักษณะเฉพาะของเอกสารนั้น ดังสมการที่ (2.1)

$$w_{kj} = \#(t_k, d_j) \times \log\left(\frac{|Tr|}{\#Tr(t_k)}\right) \quad (2.1)$$

โดย $|Tr|$ คือ จำนวนเอกสารทั้งหมด

$\#Tr(t_k)$ คือ จำนวนเอกสารที่มีคำ t_k

$\#(t_k, d_j)$ คือ จำนวนการปรากฏของคำ t_k ที่ปรากฏอยู่ในเอกสาร d_j

2.4 การลดขนาดลักษณะเฉพาะ

เว็บเพจที่มีขนาดใหญ่ หมายถึง เว็บเพจที่มีจำนวนลักษณะเฉพาะมาก ซึ่งจำนวนลักษณะเฉพาะมีผลต่อประสิทธิภาพของการจำแนกหมวดหมู่เว็บเพจ เนื่องจากอัลกอริทึมที่ใช้ในการเรียนรู้เพื่อสร้างตัวจำแนกหมวดหมู่โดยทั่วไปไม่สามารถรองรับการทำงานกับจำนวนลักษณะเฉพาะของเอกสารที่สูงมากได้ดี และเอกสารที่มีจำนวนลักษณะเฉพาะมากอาจก่อให้เกิดปัญหา Overfitting ซึ่งเป็นปรากฏการณ์ที่ตัวจำแนกหมวกหมู่ค้นพบลักษณะโดยบังเอิญของเอกสารตัวอย่าง แทนที่จะค้นพบลักษณะพื้นฐานที่จำเป็นของเอกสารตัวอย่าง ทำให้ตัวจำแนกหมวดหมู่ทำงานผิดพลาด การลดขนาดลักษณะเฉพาะจึงเป็นขั้นตอนหนึ่งที่ต้องทำก่อนการสร้างตัวจำแนกเอกสาร แต่การลดขนาดของลักษณะเฉพาะต้องพิจารณาด้วยความรอบคอบ เนื่องจากการลดขนาดลักษณะเฉพาะอาจเป็นการกำจัดลักษณะเฉพาะที่สำคัญต่อการจำแนกหมวกหมู่ออกไปด้วย เทคนิคการลดขนาดลักษณะเฉพาะในงานวิจัยนี้ประกอบด้วย 4 วิธี ดังต่อไปนี้

2.4.1 ReliefF Feature Reduction (Robnik-Sikonja M. and Kononenko I., 2003) ใช้การประเมินค่าความแตกต่างของลักษณะเฉพาะกับตัวอย่างใกล้เคียง (K nearest Neighbours) ในคลาสเดียวกันหรือต่างคลาสกันจำนวน *K* ตัว ถ้าค่าความแตกต่างเป็น 1 แสดงว่ามีความแตกต่างกันมาก ถ้าค่าที่ได้เป็น 0 จะมีความเหมือนกันมาก โดยสามารถแสดงขั้นตอนวิธีของวิธีการนี้ ดังภาพประกอบ 2.2

Algorithm ReliefF

Input: for each training instance a vector of attribute values and the class value

Output: the vector W of estimations of the qualities of attributes

1: set all weights $W[A] := 0.0$;

2: for i :=1 to m do begin

3: randomly select an instance R_i;

4: find k nearest hits H_j;

5: for each class C ≠ class(R_i) do

6: from class C find k nearest misses $M_j(C)$;

7: for A:=1 to a do

8: $W[A] := W[A] - \sum_{j=1}^{k} \mathrm{diff}(A, Ri, Hj)/(m \cdot k) + \sum_{C \neq \mathrm{class}(R_i)} \left(\frac{P(C)}{1 - P(\mathrm{class}(R_i))} \sum_{j=1}^{k} \mathrm{diff}(A, R_i, M_j(C)) \right) / (m \cdot k)$

9: end;

ภาพประกอบ 2.2 ขั้นตอนวิธีของวิธีการ ReliefF

2.4.2 Information Gain Feature Reduction ใช้การประเมินค่าของลักษณะเฉพาะโดยวัด Information Gain (Genchev *et al*., 2006; Ian and Frank, 2005b) ซึ่งเป็นตัววัดความสัมพันธ์ของลักษณะเฉพาะให้กับคลาสนั้น ๆ การหาค่า IG (Information Gain) สามารถคำนวณได้ดังสมการที่ (2.2)

$$IG = H(Y) - H(Y|X) \tag{2.2}$$

กำหนดให้ Y คือ คลาส และ X คือลักษณะเฉพาะ

$H(Y)$ คือ ค่าเอนโทรปีของ Y

$H(Y|X)$ คือ ค่าเอนโทรปีของ Y เมื่อมีเงื่อนไข X

การหาค่า $H(Y)$ แสดงได้ดังสมการที่ (2.3) และการหาค่า $H(Y|X)$ แสดงได้ดังสมการที่ (2.4)

$$H(Y) = -\sum_{y \in Y} p(y) \log_2 p((y)) \tag{2.3}$$

$$H(Y \mid X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y \mid x) \log_2 p((y \mid x)) \tag{2.4}$$

โดยที่ $p(y)$ คือ ความน่าจะเป็นของ y

$p(x)$ คือ ความน่าจะเป็นของ x

$p(y|x)$ คือ ความน่าจะเป็นของ y เมื่อรู้ x

2.4.3 Gain Ratio Feature Reduction ใช้การประเมินค่าของลักษณะเฉพาะโดยวัดค่า Gain Ratio (Ganchev *et al*., 2006; Ian and Frank, 2005b) ซึ่งวัดความสัมพันธ์ของลักษณะเฉพาะอีกประเภทหนึ่งแต่จะมีการปรับสเกลตามค่าของข้อมูลในลักษณะเฉพาะที่สนใจให้กับคลาสนั้น ๆ การคำนวณค่า Gain Ratio เป็นดังสมการที่ (2.5)

$$GR = \frac{IG}{H(X)} \tag{2.5}$$

โดยที่ $H(X)$ คือ ค่าเอนโทรปีของ X

2.4.4 Chi Sqaure Feature Reduction ใช้การประเมินค่าของลักษณะเฉพาะโดยวัดค่า Chi-Square ซึ่งวัดค่าความสัมพันธ์ระหว่างลักษณะเฉพาะกับคลาสเพื่อจัดลำดับลักษณะเฉพาะตามค่านัยสำคัญทางสถิติ โดยค่าไคส์แควร์ของแต่ละลักษณะเฉพาะ (Li, G. L., and Leong T. Y., 2005) หาได้ดังสมการที่ (2.6)

$$\chi^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \tag{2.6}$$

โดยที่ A_{ij} คือ ความถี่จริงของตัวอย่างที่มีค่าที่ i และคลาสที่ j

E_{ij} คือ ความถี่คาดหวังของตัวอย่างที่มีค่าที่ i และคลาสที่ j หรือ ความถี่คาดหวังของ A_{ij} คำนวณดังสมการที่ (2.7)

m คือ จำนวนค่าของลักษณะเฉพาะ

n คือ จำนวนของคลาส

$$E_{ij} = R_i \times \frac{C_j}{N} \tag{2.7}$$

โดยที่ R_i คือ จำนวนตัวอย่างทั้งหมดที่มีค่าลักษณะเฉพาะที่ i

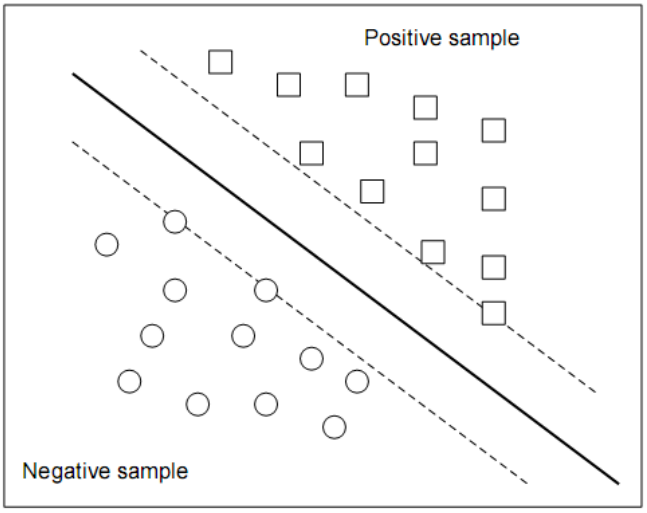
C_j คือ จำนวนตัวอย่างทั้งหมดที่อยู่ในคลาสที่ j

N คือ จำนวนของตัวอย่างทั้งหมด

ค่าไคส์แควร์สำหรับแต่ละลักษณะเฉพาะหาค่าได้จากความแตกต่างระหว่างค่าความถี่คาดหวังและค่าความถี่จริง ลักษณะเฉพาะที่มีค่าไคส์แควร์มากจะมีนัยสำคัญสูง โดยลักษณะเฉพาะจะถูกจัดเรียงตามค่าไคส์แควร์จากค่ามากไปน้อย

2.5 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine)

แนวคิดหลักของ Support Vector Machine (SVM) (Rui *et al*, 2006) นำมาใช้เพื่อหาระนาบการตัดสินใจในการแบ่งข้อมูลออกเป็นสองส่วน ใช้สำหรับข้อมูลที่มีมิติของข้อมูลสูง แสดงดังภาพประกอบ 2.3



ภาพประกอบ 2.3 ระนาบตัดสินใจของ SVM

กำหนดให้ $(x_i, y_i), \ldots, (x_n, y_n)$ เป็นตัวอย่างที่ใช้สำหรับการสอน *n* คือ จำนวนข้อมูลตัวอย่าง *m* คือ จำนวนมิติข้อมูลเข้า และ *y* คือ ผลลัพธ์มีค่า +1 หรือ -1 ดังสมการที่ (2.8)

$$(x_i, y_i),\ldots,(x_n, y_n) \text{ เมื่อ } x \in R^m \ , y \in \{+1,-1\} \tag{2.8}$$

สำหรับปัญหาเชิงเส้น มิติข้อมูลขนาดสูงได้ถูกแบ่งเป็น 2 กลุ่มโดยระนาบตัดสินใจ ซึ่งคำนวณได้ดังสมการที่ (2.9)

$$(w \cdot x) + b = 0 \tag{2.9}$$

เมื่อ *w* คือ ค่าน้ำหนัก และ *b* คือ ค่า bias สมการที่ (2.10) ใช้สำหรับจำแนกประเภทของข้อมูล

$$(w \cdot x) + b > 0 \text{ ถ้า } y_i = +1 \text{ และ } (w \cdot x) + b < 0 \text{ ถ้า } y_i = -1 \tag{2.10}$$

2.5.1 One-against-all Support Vector Machine (SVM)

เป็นวิธีการที่ง่ายที่สุดในการทำมัลติคลาส (Multi-Class) (Vapnik V., 1998) เมื่อทำการจำแนกจำนวน M คลาส แต่ละคลาสต้องทำการเปรียบเทียบค่า discriminant function ด้วยจำนวน M-1 คลาส โดยคลาสที่มีค่า discriminant function มากที่สุด จะถูกเลือกเป็นผลลัพธ์สำหรับการจำแนกประเภท วิธีการ winner-takes-all จะถูกใช้สำหรับระบุคลาสของ

ข้อมูล *x* คลาสที่มีค่าของ discriminant function มากที่สุดจะได้รับเลือกเป็นผลลัพธ์ของการจำแนกประเภท โดย class *k* ที่มีผลลัพธ์เป็น *y* สำหรับข้อมูล *x* สามารถคำนวณได้ดังสมการ (2.11)

$$y = \begin{cases} +1 & if \quad c = k \\ -1 & if \quad c \neq k \end{cases} \tag{2.11}$$

2.5.2 One-against-one Support Vector Machine (SVM)

เป็นอีกวิธีการหนึ่งในการทำมัลติคลาส (Multi-Class) (Knerr S., 1990) ที่ตั้งอยู่บนหลักการของไบนารี เมื่อทำการจำแนกจำนวน M คลาส ต้องทำการเปรียบเทียบค่า discriminant function ระหว่างหนึ่งคลาสต่อหนึ่งคลาส ด้วยจำนวน M(M-1)/2 คลาส โดยคลาสที่มีค่า discriminant function มากกว่า จะถูกเลือกเป็นผลลัพธ์ คลาสที่มีจำนวนการถูกเลือกมากที่สุดจะเป็นผลลัพธ์ของการจำแนกประเภท แต่หากมีคลาสที่เป็นผลลัพธ์มากกว่า 1 คลาส จะพิจารณาจากผลรวมของค่าสัมประสิทธิ์ของ discriminant function ที่มากกว่า ผลลัพธ์ *y* สำหรับข้อมูลสอน *x* สามารถคำนวณได้ดังสมการ (2.12)

$$y = \begin{cases} +1 & if \quad c = k \\ -1 & if \quad c = m \end{cases} \tag{2.12}$$

วิธีการ Maxwins จะถูกใช้สำหรับระบุคลาสของข้อมูล *x* โดยค่า discriminant function ของ SVM สำหรับคู่ของ classe (*k,m*) หมายถึงคลาส *k* และ *m* ถ้าค่าที่ได้เป็นค่าบวกหมายถึงว่า class *k* ได้รับการเลือกแต่ถ้าเป็นลบหมายถึง class *m* ได้รับการเลือก

2.6 ต้นไม้ตัดสินใจ C4.5

การสร้างต้นไม้ตัดสินใจ C4.5 (Quinla J. R., 1993; Vidulin V., 2007) ใช้ค่ามาตรฐานอัตราส่วนเกน (Gain Ratio) เพื่อเลือกลักษณะเฉพาะที่จะใช้เป็นรากหรือโหนด ถ้าให้ชุดของข้อมูล *M* ประกอบด้วยค่าที่เป็นไปได้ คือ $\{m_1, m_2, \ldots, m_n\}$ และให้ความน่าจะเป็นที่จะเกิดค่า m_i มีค่าเท่ากับ $P(m_i)$ จะได้ว่าค่า เกนสารสนเทศ (Information Gain) ของ *M* เขียนแทนด้วย *I(M)* คำนวณได้ดังสมการที่ (2.13)

$$I(M) = \sum_{i=1}^{n} - P(m_i) \log_2 P(m_i) \tag{2.13}$$

ถ้าให้ข้อมูลสอน คือ *T* และลักษณะเฉพาะที่เป็นโหนด คือ *x* และมีค่าทั้งหมดที่เป็นไปได้ *n* ค่า โหนดปัจจุบันจะแบ่งตัวอย่าง *T* ออกตามกิ่งเป็น $\{t_1, t_2, \ldots, t_n\}$ ตามค่าที่เป็นไปได้ของ *x* ดังนั้นจึงสามารถคำนวณค่าเกนสารสนเทศ (Information Gain) หลังจากแบ่งตามลักษณะเฉพาะ *x* ได้ดังสมการที่ (2.14)

$$I_x(T) = \sum_{i=1}^{n} \frac{|t_i|}{|T|} I(t_i) \quad (2.14)$$

ค่ามาตรฐานเกน (Gain) ของลักษณะเฉพาะ *x* ได้ดังสมการที่ (2.15)

$$Gain(x) = I(T) - I_X(T) \quad (2.15)$$

จากนั้นคำนวณค่าสารสนเทศของการแบ่งแยก (Split Information) ของลักษณะเฉพาะแต่ละตัว ถ้าให้ *T* คือ ชุดของตัวอย่าง เมื่อแบ่งตัวอย่างนี้ตามลักษณะเฉพาะ *x* จะได้ชุดของตัวอย่างย่อยในแต่ละกิ่ง คือ $\{t_1, t_2, \ldots, t_n\}$ จำนวน *n* ชุด ตามค่าที่เป็นไปได้ในคุณสมบัติ *x* เมื่อคำนวณค่าสารสนเทศของการแบ่งแยกได้ดังสมการที่ (2.16)

$$Split\ Information = -\sum_{i=1}^{n} \frac{|t_i|}{|T|} \log_2 \frac{|t_i|}{|T|} \quad (2.16)$$

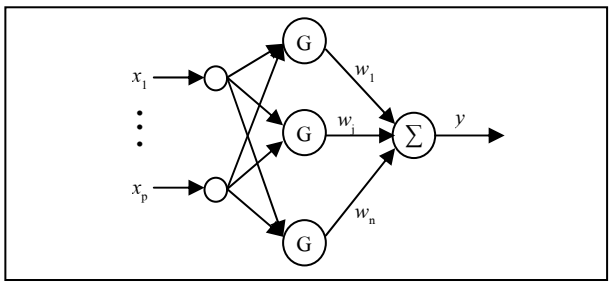
คำนวณค่ามาตรฐานอัตราส่วนเกน (Gain Ratio) ได้ดังสมการที่ (2.17)

$$Gain\ Ratio = Gain - Split\ Information \quad (2.17)$$

ท้ายสุดจึงเลือกค่า *Gain Ratio* สูงสุดเป็นลักษณะเฉพาะเริ่มต้น และเลือกลักษณะเฉพาะถัดไปตามค่า *Gain Ratio* น้อยลงตามลำดับ

2.7 Radial Basis Function Neural Networks (RBFNN)

RBF Neural Networks (Xuhong W. and Jinhua X., 2004; Jujie, C., Rongbing, J., 2004) ประกอบด้วยชั้นข้อมูลเข้า (Input Layer) ชั้นซ่อน (Hidden Layer) และชั้นข้อมูลออก (Output Layer) ดังภาพประกอบ 2.4 ซึ่งมีเกาซ์เซียนฟังก์ชัน (Gaussian Function) เป็นฟังก์ชันกระตุ้นในชั้นซ่อน ดังสมการที่ (2.18)



ภาพประกอบ 2.4 RBF Neural Networks

$$\phi_j(x) = \exp\left[-\frac{\|x - c_j E\|}{2\sigma_j^2} \right] \text{เมื่อ } j = 1,2,\ldots,n \quad (2.18)$$

โดยที่ ϕ คือ ข้อมูลออกของนิวรอลที่ *j* ในชั้นซ่อน *x* คือเวกเตอร์ข้อมูลเข้า c_j และ σ_j คือ ศูนย์กลาง และช่วงกว้างของนิวรอลที่ *j* ตามลำดับ ข้อมูลออกของโครงข่าย RBF คำนวณดังสมการที่ (2.19)

$$y = i_c(k+1) = \sum_{j=1}^{n} w_j \phi_j(x) \quad (2.19)$$

โดยที่ n คือ จำนวนของนิวรอลในชั้นซ่อน w_j คือน้ำหนักระหว่างชั้นซ่อนและชั้นข้อมูลออก และ y คือผลลัพธ์

2.8 การประเมินประสิทธิภาพ

การจำแนกหมวดหมู่เอกสารจะใช้วิธีการเรียนรู้โดยอาศัยตัวอย่าง ส่วนการจัดกลุ่มเอกสารจะใช้วิธีการเรียนรู้โดยไม่อาศัยตัวอย่าง เอกสารที่ใช้ในกระบวนการเรียนรู้นั้นจะถูกเรียกว่า คลังเอกสารเริ่มต้น (Initial Corpus) การวัดประสิทธิภาพนิยมใช้วิธีทางด้านการค้นคืนสารสนเทศ ซึ่งผลของการจำแนกประเภทเว็บเพจสามารถเขียนเป็นตารางตัดสินใจการจำแนกประเภท ได้ดังตารางที่ 2.2 และประเมินค่าด้วยการวัดค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และค่า F (F-measure)

ตารางที่ 2.2 ตารางตัดสินใจการจำแนกประเภท

ค่าที่จำแนก โดยตัวจำแนกประเภท	**ค่าของ C_i**	
	ใช่	**ไม่ใช่**
ใช่	TP_i	FP_i
ไม่ใช่	FN_i	TN_i

กำหนดให้ i คือ ลำดับของ Class (C) ที่ i จากตารางที่ 2.2 จะได้

TP_i (True Positive) คือ จำนวนเว็บเพจที่อยู่ใน C_i และตัวจำแนกประเภททำนายว่าอยู่ใน C_i

FP_i (Fasle Positive) คือ จำนวนเว็บเพจที่ไม่อยู่ใน C_i แต่ตัวจำแนกประเภททำนายว่าอยู่ใน C_i

FN_i (Fasle Negative) คือ จำนวนเว็บเพจที่อยู่ใน C_i แต่ตัวจำแนกประเภททำนายว่าไม่อยู่ใน C_i

TN_i (True Negative) คือ จำนวนเว็บเพจที่ไม่อยู่ใน C_i และตัวจำแนกประเภททำนายว่าไม่อยู่ใน C_i

สามารถคำนวณค่าความแม่นยำ (P_i) ค่าระลึก (R_i) และ ค่า F-measure (F_i) ได้ดังสมการที่ (2.20) ถึง (2.22)

$$P_i = \frac{TP_i}{TP_i + FP_i} \quad (2.20)$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \quad (2.21)$$

$$F_i = \frac{2P_i R_i}{P_i + R_i} \tag{2.22}$$

ข้อสังเกต การวัดประสิทธิผลของการจำแนกประเภท ควรพิจารณาค่าความแม่นยำ และค่าความระลึกร่วมกัน กับค่า F-measure เนื่องจากในการจำแนกประเภทนั้น กรณีที่มีค่า FN_i มาก แม้จะมีค่าความแม่นยำสูง (P_i) อาจมีค่าความระลึก (R_i) ที่ต่ำได้ และในกรณีที่ถ้ามีค่า FP_i มากแม้มีค่าความแม่นยำต่ำ (P_i) อาจให้ค่าความระลึก (R_i) ที่สูง เป็นต้น

บทที่ 3

แบบจำลองการจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน

วิทยานิพนธ์นี้มุ่งเน้นไปที่การสร้างแบบจำลองสำหรับเพิ่มความถูกต้องของการจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน โดยสร้างแบบจำลองที่มีการให้คะแนนเสียงร่วมกับมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีน (Web Page Classification Using Voting Algorithm And Multi-Class Support Vector Machine: WPC_VAMSVM) ออกแบบเพื่อทดสอบประสิทธิภาพของการจำแนกประเภทเว็บที่มีการให้คะแนนเสียงของผลลัพธ์ของการจำแนกประเภทโดยใช้มัลติคลาสซัพพอร์ตเวกเตอร์แมชชีน ซึ่งแสดงได้ดังภาพประกอบ 3.1 แบ่งการทำงานออกเป็น 4 ขั้นตอนหลักคือ ขั้นตอนที่ 1 การเตรียมข้อมูลเว็บเพจ (Web Page Preprocessing) ขั้นตอนที่ 2 การสร้างลักษณะเฉพาะ (Feature Generation) ขั้นตอนที่ 3 การลดขนาดลักษณะเฉพาะ (Feature Reduction) และขั้นตอนที่ 4 การจำแนกประเภทและการให้คะแนนเสียง (Classification and Voting)

ขั้นตอนที่ 1: การเตรียมข้อมูลเว็บเพจ (Web Page Preprocessing)
1.1 สกัดข้อความ (Text) และหัวเรื่อง (Title) จากหน้าเว็บเพจ
1.2 หารากศัพท์ของคำ (Stemming)โดยใช้อัลกอริทึม Porter
1.3 กำจัดคำหยุด (Stopping)
ขั้นตอนที่ 2: การสร้างลักษณะเฉพาะ (Feature Generation)
2.1 สร้างลักษณะเฉพาะของข้อความ
2.1.1 สร้าง term document matrix จากข้อความ
2.1.2 ให้ค่าน้ำหนักคำด้วยวิธี TF-IDF
2.1.3 เลือกคำที่มี document frequency มากกว่าค่า threshold
2.2 สร้างลักษณะเฉพาะของหัวเรื่อง
2.2.1 สร้าง term document matrix จากหัวเรื่อง
2.2.2 ให้ค่าน้ำหนักคำด้วยวิธี TF-IDF
2.3 รวมลักษณะเฉพาะจากข้อความและหัวเรื่องเข้าด้วยกัน

ภาพประกอบ 3.1 แบบจำลองการจำแนกประเภทเว็บเพจโดยใช้การให้คะแนนเสียงและมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีน

ขั้นตอนที่ 3: การลดขนาดลักษณะเฉพาะ (Feature Reduction)
3.1 ทำการเลือกลักษณะเฉพาะโดย 2 วิธีดังต่อไปนี้ ReliefF feature selection, Information Gain feature selection, Gain Ratio feature selection และ Chi Square feature selection 3.2 กำหนดลักษณะเฉพาะที่ได้จากขั้นตอนที่ 3.1 เป็น Leftfeature และ Rightfeature
ขั้นตอนที่ 4: การจำแนกประเภทและการให้คะแนนเสียง (Classification and Voting)
4.1 เลือกวิธีการจำแนกประเภทและการให้คะแนนเสียง (a) One-vs-One (1vs1) • จำแนกประเภททั้ง Leftfeature และ Rightfeature โดยวิธีการ 1vs1 SVM • ทำนายประเภทของเว็บเพจโดยใช้อัลกอริทึม 1vs1_Voting (b) One-vs-All (1vsAll) • จำแนกประเภททั้ง Leftfeature และ Rightfeature โดยวิธีการ 1vsAll SVM • ทำนายประเภทของเว็บเพจโดยใช้อัลกอริทึม 1vsAll_Voting 4.2 ประเมินประสิทธิภาพของการจำแนกด้วยค่า F-measure

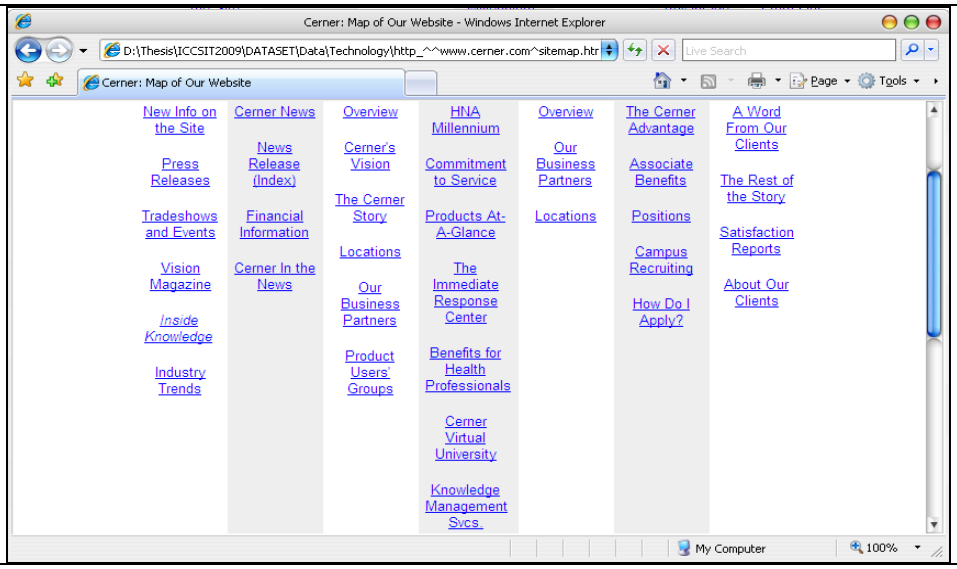
ภาพประกอบ 3.1 แบบจำลองการจำแนกประเภทเว็บเพจโดยใช้การให้คะแนนเสียง และมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีน (ต่อ)

แบบจำลอง WPC_VAMSVM ขั้นตอนที่ 1 คือการเตรียมข้อมูลเว็บเพจ (Web Page Preprocessing) มีจุดประสงค์เพื่มเตรียมข้อมูลให้อยู่ในรูปแบบที่พร้อมจะประมวลผล เช่น มีการสกัดเฉพาะข้อความและหัวเรื่องจากหน้าเว็บเพจเพื่อเตรียมข้อมูลเหล่านี้สำหรับการสร้างลักษณะเฉพาะต่อไป ขั้นตอนที่ 2 การสร้างลักษณะเฉพาะ (Feature Generation) มีจุดประสงค์เพื่อนำข้อมูลที่ได้จากข้อความและหัวเรื่องมาสร้างเป็นลักษณะเฉพาะเพื่อเป็นตัวแทนสำหรับการจำแนกประเภท ขั้นตอนที่ 3 การลดขนาดลักษณะเฉพาะ (Feature Reduction) มีจุดประสงค์เพื่อลดขนาดของลักษณะเฉพาะและเพิ่มประสิทธิภาพให้แก่การจำแนกประเภทเว็บเพจ ขั้นตอนที่ 4 การจำแนกประเภทและการให้คะแนนเสียง (Classification and Voting) มีจุดประสงค์เพื่อจำแนกประเภทเว็บเพจและนำผลลัพธ์ที่ได้จากการจำแนกประเภทมาพิจารณาผลลัพธ์สุดท้ายโดยวิธีการให้คะแนนเสียง รายละเอียดการทำงานของแบบจำลอง WPC_VAMSVM ในแต่ละขั้นตอนมีดังนี้

3.1 ขั้นตอนการเตรียมข้อมูลเว็บเพจ (Web Page Preprocessing)

ขั้นตอนที่ 1 เป็นขั้นตอนการเตรียมข้อมูลเว็บเพจให้พร้อมที่จะนำไปประมวลผล เนื่องจากหน้าเว็บเพจประกอบไปด้วยข้อมูลซึ่งอยู่ในรูปแบบที่แตกต่างกัน

ขั้นตอนที่ 1.1 ทำการสกัดข้อความและหัวเรื่องจากหน้าเว็บเพจ ตัวอย่างข้อมูลเว็บเพจที่มีการแสดงดังภาพประกอบ 3.2 และสามารถสกัดข้อความ ดังภาพประกอบ 3.3 และหัวเรื่องแสดงดังภาพประกอบ 3.4



ภาพประกอบ 3.2 ตัวอย่างหน้าเว็บเพจ

New Info on the Site Press Releases Tradeshows and Events Vision Magazine Inside Knowledge Industry Trends Cerner News News Release (Index) Financial Information Cerner In the News Overview Cerner's Vision The Cerner Story Locations Our Business Partners Product Users' Groups HNA Millennium Commitment to Service Products At-A-Glance The Immediate Response Center Benefits for Health Professionals Cerner Virtual University Knowledge Management Svcs. Laboratory Consolidation Product Users' Groups Cerner Direct Overview Our Business Partners Locations

ภาพประกอบ 3.3 ตัวอย่างข้อความที่สกัดได้จากหน้าเว็บเพจ

Cencer: Map of Our Website

ภาพประกอบ 3.4 ตัวอย่างหัวเรื่องที่สกัดได้จากเว็บเพจ

ขั้นตอนที่ 1.2 หารากศัพท์ของคำที่ได้จากทั้งข้อความและหัวเรื่องโดยใช้อัลกอริทึม Porter

ขั้นตอนที่ 1.3 กำจัดคำที่เป็น Stoplist ออก เนื่องจากคำที่เป็น Stoplist เป็นคำที่ฟุ่มเฟือยและไม่ได้นำมาวิเคราะห์ เช่น a, and, on, at, their เป็นต้น

3.2 ขั้นตอนการสร้างลักษณะเฉพาะ (Feature Generation)

ขั้นตอนที่ 2 เป็นขั้นตอนการสร้างลักษณะเฉพาะที่ได้จากขั้นตอนที่ 1 เพื่อใช้เป็นข้อมูลเพื่อการจำแนกประเภทต่อไป

ขั้นตอนที่ 2.1 นำข้อความมาสร้างเป็นลักษณะเฉพาะซึ่งอยู่ในรูปแบบของเมตริกส์เอกสาร ดังภาพประกอบ 3.5 โดยกำหนดค่าน้ำหนักของคำโดยใช้วิธีการ TF-IDF จากนั้นจึงเลือกคำที่มีค่าความถี่เอกสาร (Document Frequency) มากกว่าค่าที่กำหนด (Threshold)

Web Page	**Feature**			
	$Word_1$	$Word_2$	…	$Word_n$
Web_1	W_{11}	W_{12}	….	W_{1n}
Web_2	W_{21}	W_{22}	….	W_{2n}
Web_3	W_{31}	W_{32}	….	W_{3n}
….	….	….	….	….
Web_m	W_{m1}	W_{m2}	….	W_{mn}

ภาพประกอบ 3.5 เมตริกซ์เอกสาร (Document matrix)

ขั้นตอนที่ 2.2 นำหัวเรื่องมาสร้างเป็นลักษณะเฉพาะซึ่งอยู่ในรูปแบบของเมตริกส์เอกสาร แล้วจึงกำหนดค่าน้ำหนักของคำโดยใช้วิธีการ TF-IDF

ขั้นตอนที่ 2.3 รวมลักษณะเฉพาะข้อความจากขั้นตอน 2.1 และลักษณะเฉพาะจากหัวเรื่องจากขั้นตอน 2.2 เข้าด้วยกัน

3.3 ขั้นตอนการลดขนาดลักษณะเฉพาะ (Feature Reduction)

ขั้นตอนที่ 3 เป็นขั้นตอนการลดขนาดลักษณะเฉพาะเพื่อให้มีขนาดที่เหมาะสมต่อการจำแนกประเภท

ขั้นตอนที่ 3.1 ลดขนาดลักษณะเฉพาะด้วยการเลือกเพียง 2 วิธีการจาก ReliefF, Information Gain, Gain Ratio และ Chi Square

ขั้นตอนที่ 3.2 กำหนดลักษณะเฉพาะที่ได้จาก 2 วิธีการข้างต้น เป็น Leftfeature และ Rightfeature

3.4 ขั้นตอนการจำแนกประเภทและการให้คะแนนเสียง (Classification and Voting)

ขั้นตอนที่ 4 เป็นขั้นตอนที่นำข้อมูลจากขั้นตอนก่อนหน้ามาทำการจำแนกประเภทเว็บเพจด้วยซัพพอร์ตเวกเตอร์แมชชีนและอัลกอริทึมการให้คะแนนเสียง

ขั้นตอนที่ 4.1 ทั้ง Leftfeature และ Rightfeature จะถูกใช้สำหรับการจำแนกประเภทโดยวิธีการ SVM ซึ่งสามารถเลือกได้ 2 วิธีคือแบบ 1vs1 หรือ 1vsAll ผลลัพธ์ที่ได้หลังจากจำแนกประเภทจะเป็นข้อมูลเข้าสำหรับอัลกอริทึมการให้คะแนนเสียงต่อไป

(a) One-vs-One (1vs1)

วิธีการนี้ทั้ง Leftfeature และ Rightfeature จะถูกใช้สำหรับการจำแนกประเภทโดยวิธีการ SVM แบบ 1vs1 ซึ่งจะถูกเรียกว่า Leftclassifier และ Rightclassifier ตามลำดับ ต่อมาผลลัพธ์ที่ได้จากการจำแนกประเภททั้งสองจะถูกนำมาทำนายผลลัพธ์สุดท้ายโดยใช้อัลกอริทึม 1vs1_Voting ดังภาพประกอบ 3.6 ซึ่งประกอบด้วย 2 ส่วนคือส่วน Numvote value และ Sum weight value ซึ่งสามารถแสดงตัวอย่างการทำงาน โดยกำหนดให้ผลลัพธ์ของการจำแนกประเภทด้วย SVM ดังภาพประกอบ 3.7 การทำงานของอัลกอริทึม 1vs1_Voting ดังภาพประกอบ 3.8 โดยส่วนแรก Numvote value จะพิจารณาค่า *total_numvote* ซึ่งเป็นผลรวมของค่า *numvote_left* และ *numvote_right* ว่าคลาสใดมีค่ามากที่สุด หากมีเพียงคลาสเดียวก็จะให้ผลลัพธ์ของอัลกอริทึมเป็นคลาสนั้น แต่หากมีมากกว่าหนึ่งคลาส จะพิจารณาโดยใช้ส่วนที่สองคือ Sum weight value โดยพิจารณาจากค่า *total_w* ซึ่งเป็นผลรวมของค่า *w_left และ w_right* ว่าคลาสใดมีค่ามากที่สุดก็จะเลือกตอบเป็นคลาสนั้น

Let *i* be the identical number of class *i*, where *i* = 1, …, *n* and *n* be the number of classes

Let *numvote_left* and *numvote_right* be the number of vote on the Leftclassifier and Rightclassifier, respectively.

Let *w_left* and *w_right* be the number of summation of absolute discriminant value on the Leftclassifier and Rightclassifier, respectively.

Input: *numvote_left*, *numvote_right, w_left*, *w_right*

Ouput: *i.name* // a name of class *i*

ภาพประกอบ 3.6 อัลกอริทึม 1vs1_Voting

1: **For** each class *i* **do** *// *** **Part 1: Numvote value** ****

2: $total\_numvote_i = numvote\_left_i + numvote\_right_i$

3: **End for**

4: *total_numvote_max* = Max (all of $total\_numvote_i$)

5: **If** *total_numvote_max* have a single class **then**

6: **Return** class *i.name* where $total\_numvote_i$ == *total_numvote_max*

7: **Else** *//*** **Part 2: Sum weight value** ****

8: **For** each class *i* where $total\_numvote_i$ == *total_numvote_max* **do**

9: $total\_w_i = w\_left_i + w\_right_i$

10: **End for**

11: *total_w_max* = Max (all of $total\_w_i$)

12: **If** *total_w_max* have a single class **then**

13: **Return** class *i*.name where $total\_w_i$ == *total_w_max*

14: **Else**

15: **Return** one of class *i.name* where $total\_w_i$ == *total_w_max*

16: **End if** // *total_w_max*

17: **End if** *// total_numvote_max*

ภาพประกอบ 3.6 อัลกอริทึม 1vs1_Voting (ต่อ)

1vs1 SVM classification					
Classifier	**Output value**	**Class**			**Predicted Class**
		C1	**C2**	**C3**	
Leftclassifier	number of vote	**2**	1	0	C1
	∑|discriminant value|	0.94	0.65	0	
Rightclassifier	number of vote	1	**2**	0	C2
	∑|discriminant value|	0.84	1.02	0	

ภาพประกอบ 3.7 ตัวอย่างผลลัพธ์ที่ได้จาก 1vs1 SVM

1vs1_Voting algorithm					
Part	**Variable**	**Class**			**Voted Class**
		C1,*i*=1	**C2,*i*=2**	**C3,*i*=3**	
Part 1: Numvote value	*numvote_left*	2	1	0	
	numvote_right	1	2	0	
	total_numvote	**3**	**3**	0	C1,C2
Part 2: Sum weight value	*w_left*	0.94	0.65		
	w_right	0.84	1.02		
	total_w	**1.78**	1.67		**C1**

ภาพประกอบ 3.8 ตัวอย่างการทำงานของอัลกอริทึม 1vs1_Voting

(b) One-vs-All (1vsAll)

วิธีการนี้ทั้ง Leftfeature และ Rightfeature จะถูกใช้สำหรับการจำแนกประเภทโดยวิธีการ SVM แบบ 1vsAll ซึ่งจะถูกเรียกว่า Leftclassifier และ Rightclassifier ตามลำดับ ต่อมาผลลัพธ์ที่ได้จากการจำแนกประเภททั้งสองจะถูกนำมาทำนายผลลัพธ์สุดท้ายโดยใช้อัลกอริทึม 1vsAll_Voting ดังภาพประกอบ 3.9 ซึ่งประกอบด้วย 3 ส่วน คือ ส่วน Rank value, Weight threshold value และ Weight value ซึ่งสามารถแสดงตัวอย่างการทำงาน โดยกำหนดให้ผลลัพธ์ของการจำแนกประเภทด้วย SVM ดังภาพประกอบ 3.10 การทำงานของอัลกอริทึม 1vs1_Voting ดังภาพประกอบ 3.11 โดยส่วนแรก ส่วน Rank value จะพิจารณาค่าลำดับ (rank) ของ discriminant value โดยค่าที่มากที่สุดจะให้ค่าลำดับสูงสุดซึ่งมีค่าเท่ากับจำนวนของคลาส ค่าที่มี discriminant value รองลงมาจะมีค่าลำดับน้อยลงตามลำดับจนถึงค่า 1 เมื่อทำการเรียงลำดับแล้วจะพิจารณาค่า *total_rank* ซึ่งเป็นผลรวมของค่า *rank_left* และ *rank_right* ว่าคลาสใดมีค่ามากที่สุด หากมีเพียงคลาสเดียวก็จะให้ผลลัพธ์ของอัลกอริทึมเป็นคลาสนั้น แต่หากมีมากกว่าหนึ่งคลาส จะพิจารณาโดยใช้ส่วนที่สองคือ Weight threshold value โดยพิจารณาจากค่า *total_E* ซึ่งเป็นผลรวมของค่า *E_ left และ E _right* ซึ่งค่าทั้งสองนี้ได้จากการพิจาณาค่า discriminant value ที่มากกว่าหรือเท่ากับ 0 จะให้เป็น 1 แต่ถ้าไม่ใช่จะให้เป็น -1 ในขั้นนี้ถ้าพบว่าค่า *total_E* คลาสใดมีค่ามากที่สุดก็จะเลือกตอบเป็นคลาสนั้น แต่หากมีมากกว่าหนึ่งคลาส จะพิจารณาโดยใช้ส่วนที่สามคือ Weight value โดยพิจารณาจากค่า *total_w* ซึ่งเป็นผลรวมของค่า *w_left และ w_right* ในขั้นนี้ถ้าพบว่าค่า *total_E* คลาสใดมีค่ามากที่สุดก็จะเลือกตอบเป็นคลาสนั้น

4.2 แบบจำลองจะทำการประเมินประสิทธิภาพของการจำแนกประเภทด้วยค่า F-measure

Let *i* be the identical number of class *i*, where *i* = 1, …, *n* and *n* be the number of classes.

Let *w_left* and *w_right* be the number of discriminant value on the Leftclassifier and Rightclassifier, respectively.

Input: *w_left*, *w_right*

Ouput: *i.name* // a name of class *i*

---

*// *** **Part 1: Rank value** ****

1: For each class *i*, rank *w_left* in descending order, then assign it to $rank\_left_i$

2: For each class *i*, rank *w_right* in descending order, then assign it to $rank\_right_i$

3: **For** each class *i* **do**

4: $total\_rank_i = rank\_left_i + rank\_right_i$

5: **End for**

6: $total\_rank\_max$ = Max (all of $total\_rank_i$)

7: **If** $total\_rank\_max$ have a single class **then**

8: **Return** class *i*.name where $total\_rank_i == total\_rank\_max$

9: **Else** *// *** **Part 2: Weight threshold value** ****

10: **For** each class *i* where $total\_rank_i == total\_rank\_max$ **do**

11: **If** $w\_left_i >= 0$ **then**

12: $E\_left_i = 1$

13: **Else**

14: $E\_left_i = -1$

15: **End if**

16: **If** $w\_right_i >= 0$ **then**

17: $E\_right_i = 1$

18: **Else**

19: $E\_right_i = -1$

20: **End if**

21: $total\_E_i = E\_left_i + E\_right_i$

22: **End for**

ภาพประกอบ 3.9 อัลกอริทึม 1vsAll_Voting

```
23:    total_E_max  = Max (all of total_E_i)
24:    If total_E_max  have a single class then
25:        Return class i.name where total_E_i  == total_E_max
-------------------------------------------------------------------------
26:     Else          // *** Part 3: Weight value ***
27:        For each class i where total_E_i  == total_E_max do
28:             total_w_i =  w_left_i + w_right_i
29:        End for
30:        total_w_max = Max (all of total_ w_i)
31:        If total_w_max have a single class then
32:           Return class i.name where total_w_i == total_w_max
33:        Else
34:           Return one of class i.name where total_w_i == total_w_max
35:        End if // total_w_max
36:    End if // total_E_max
37: End if // total_rank_max
```

ภาพประกอบ 3.9 อัลกอริทึม 1vsAll_Voting (ต่อ)

1vsAll SVM classification					
Classifier	**Output value**	**Class**			**Predicted Class**
		C1	**C2**	**C3**	
Leftclassifier	discriminant value	**0.65**	-0.94	-1.90	C1
Rightclassifier	discriminant value	-1.89	**6.84**	-4.19	C2

ภาพประกอบ 3.10 ตัวอย่างผลลัพธ์ที่ได้จาก 1vsAll SVM

1vsAll_Voting algorithm					
Part	**Variable**	**Class**			**Voted Class**
		C1,*i*=1	**C2,*i*=2**	**C3,*i*=3**	
Part 1: Rank value	*rank_left*	3	2	1	
	rank_right	2	3	1	
	total_rank	**5**	**5**	2	C1,C2
Part 2: Weight threshold value	*E_ left*	1	-1		
	E _right	-1	1		
	total_E	**0**	**0**		C1,C2
Part 3: Weight value	*w_left*	0.65	-0.94		
	w_right	-1.89	6.84		
	total_w	-1.24	**5.9**		**C2**

ภาพประกอบ 3.11 ตัวอย่างการทำงานของอัลกอริทึม 1vsAll_Voting

บทที่ 4

โปรแกรมการจำแนกประเภทเว็บเพจโดยใช้การให้คะแนนเสียง และมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีน

การพัฒนาโปรแกรมการจำแนกประเภทเว็บเพจโดยการลดขนาดลักษณะเฉพาะและมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีนได้ใช้แนวคิดและลำดับขั้นตอนการทำงานตามแบบจำลองการจำแนกประเภทเว็บเพจโดยใช้การให้คะแนนเสียงและมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีน (Web Page Classification Using Voting Algorithm And Multi-Class Support Vector Machine: WPC_VAMSVM) การทำงานของโปรแกรมจะอธิบายด้วยผังการทำงานของโปรแกรม ส่วนประกอบของโปรแกรม ผลการทำงานของโปรแกรมและเครื่องมือที่ใช้ในการพัฒนาโปรแกรม

4.1 ผังการทำงานของโปรแกรม

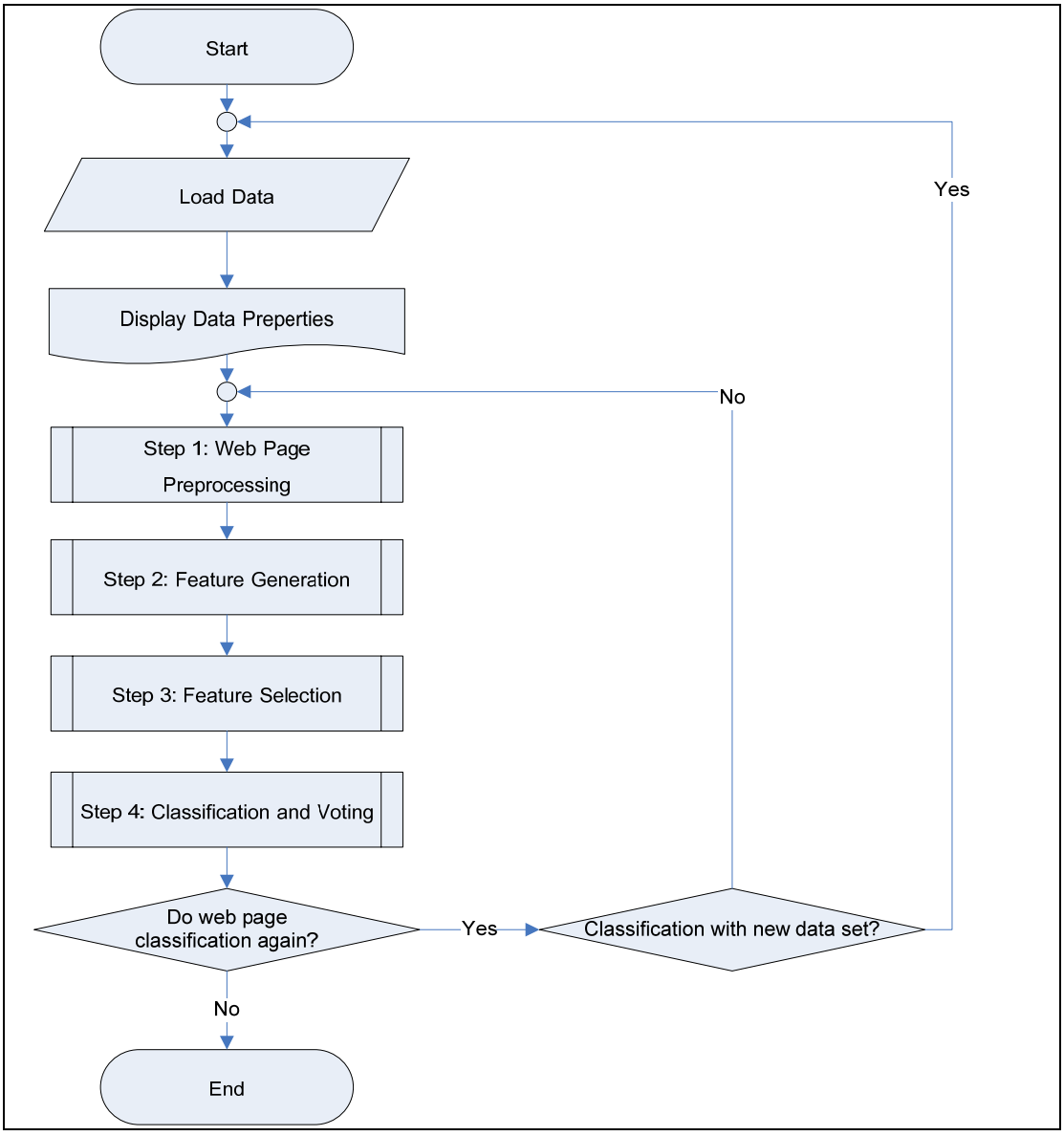
4.1.1 ผังงานโปรแกรมหลักของโปรแกรมสำหรับจำแนกประเภทดังภาพประกอบ 4.1 โดยขั้นตอนการทำงานทั้งหมดของโปรแกรมประกอบด้วย 4 ขั้นตอนคือ 1) การเตรียมข้อมูลเว็บเพจ (Web Page Preprocessing) 2) การสร้างลักษณะเฉพาะ (Feature Generation) 3) การลดขนาดลักษณะเฉพาะ (Feature Reduction) 4) การจำแนกประเภทและการให้คะแนนเสียง (Classification and Voting)

4.1.2 ผังงานโปรแกรมย่อยของ Step 1: Web Page Preprocessing แสดงดังภาพประกอบ 4.2 ซึ่งการทำงานจะสกัดข้อความและหัวเรื่องออกจากหน้าเว็บเพจ จากนั้นจะนำคำเหล่านี้มาหารากศัพท์ของคำด้วยวิธีการ Porter สุดท้ายจึงตัดคำที่เป็น Stoplist ออกไป

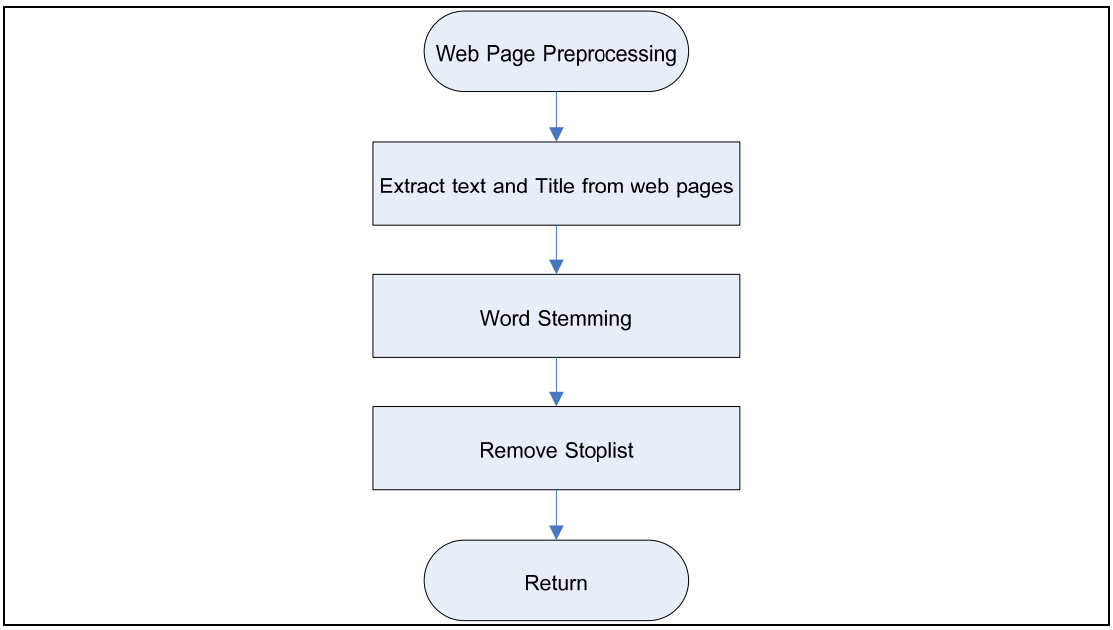
4.1.3 ผังงานโปรแกรมย่อยของ Step 2: Feature Generation แสดงดังภาพประกอบ 4.3 ซึ่งการทำงานจะสร้างลักษณะเฉพาะจากข้อความและหัวเรื่อง โดยสามารถเลือกเฉพาะข้อความหรือมีการรวมหัวเรื่องเข้าไปด้วย และในส่วนของข้อความมีส่วนของการระบุ Document Frequency Threshold ด้วย

4.1.4 ผังงานโปรแกรมย่อยของ Step 3: Feature Reduction แสดงดังภาพประกอบ 4.4 ซึ่งการทำงานจะประกอบด้วยการเลือกเทคนิคสำหรับการลดขนาดลักษณะเฉพาะและระบุจำนวนลักษณะเฉพาะที่ต้องการ

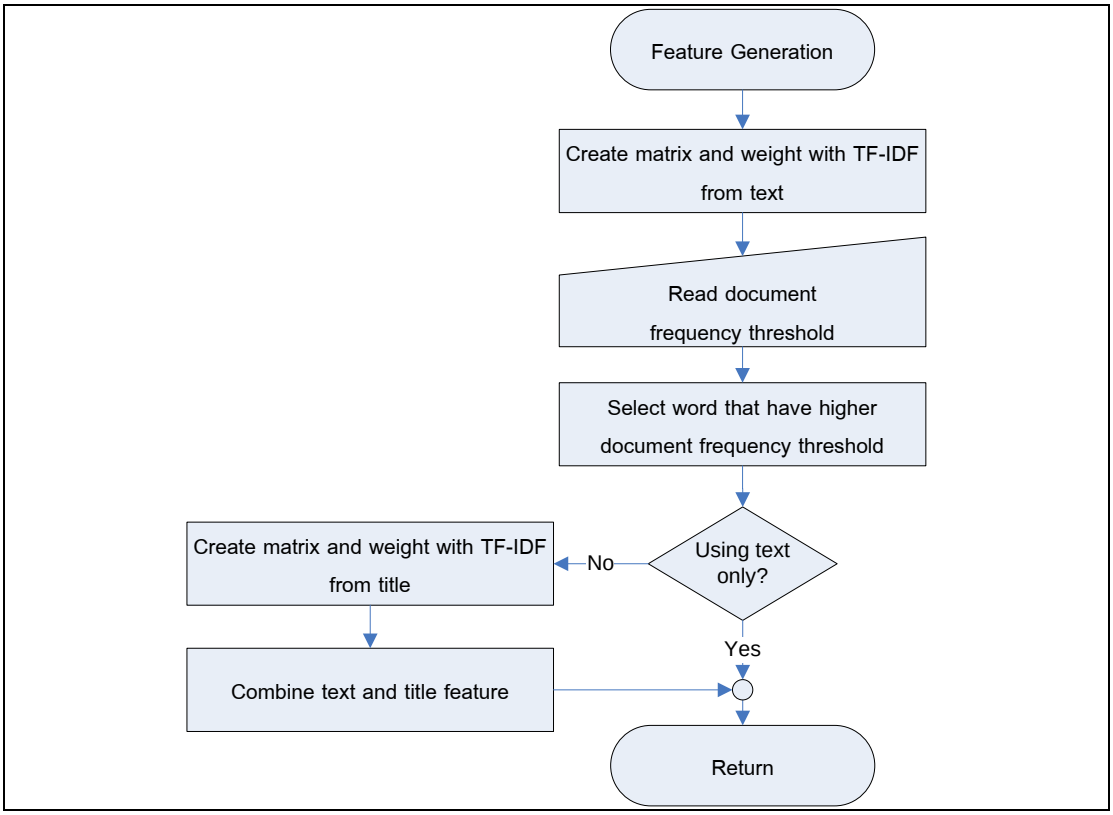
4.1.5 ผังงานโปรแกรมย่อยของ Step 4: Classification and Voting แสดงดังภาพประกอบ 4.5 ซึ่งการทำงานจะประกอบด้วยการเลือกวิธีการที่ใช้ในการจำแนกประเภทและการให้คะแนนเสียงและระบุจำนวนของการทดสอบแบบไขว้เปลี่ยน (Cross Validation) ที่ต้องการ



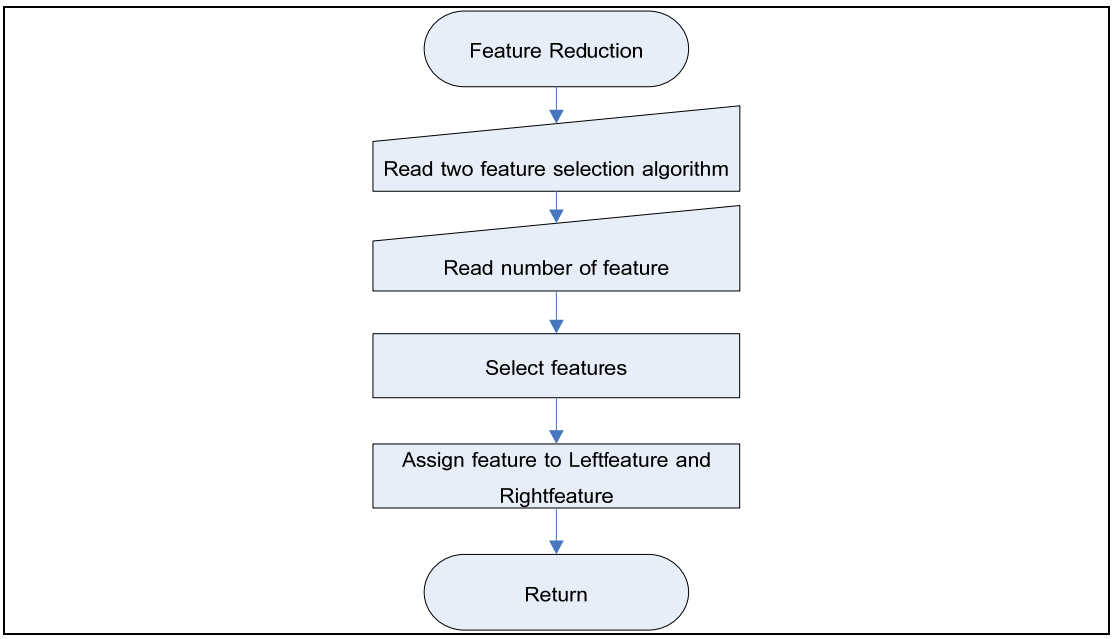
ภาพประกอบ 4.1 ผังการทำงานของ WPC_VAMSVM



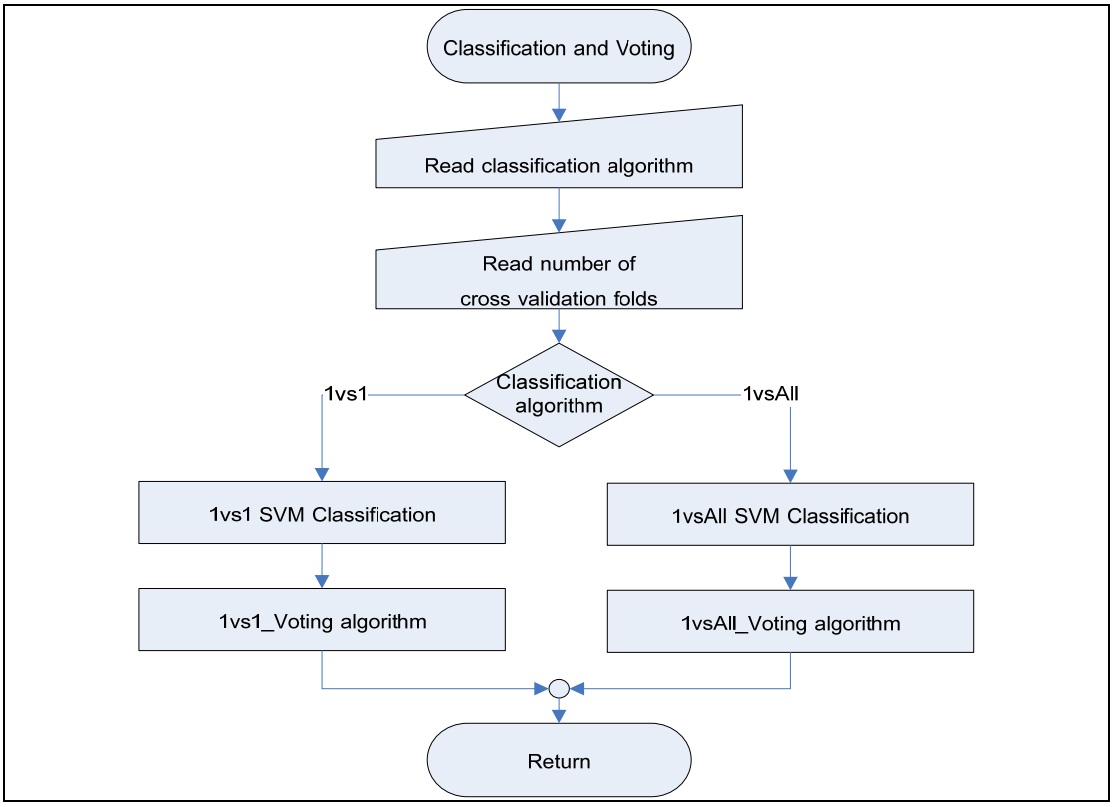
ภาพประกอบ 4.2 ผังการทำงานของการเตรียมข้อมูลเว็บเพจ (Web Page Preprocessing)



ภาพประกอบ 4.3 ผังการทำงานของการสร้างลักษณะเฉพาะ (Feature Generation)



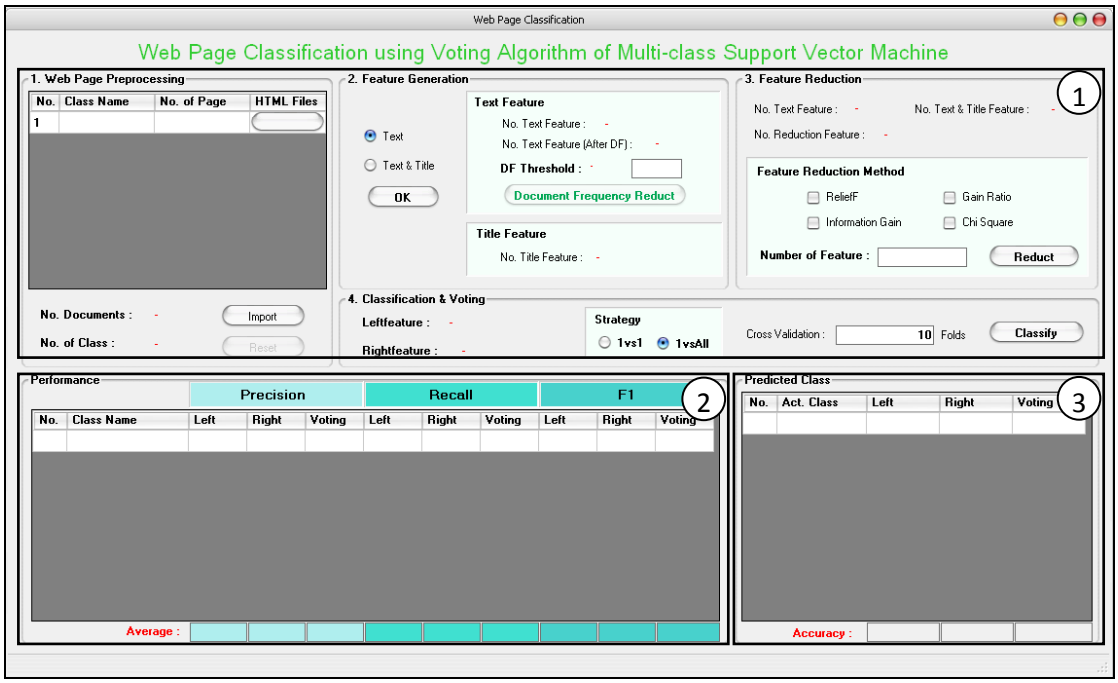
ภาพประกอบ 4.4 ผังการทำงานของการลดขนาดลักษณะเฉพาะ (Feature Selection)



ภาพประกอบ 4.5 ผังการทำงานของการจำแนกประเภทและการให้คะแนนเสียง (Classification and Voting)

4.2 ส่วนประกอบของโปรแกรม

ตัวอย่างการทำงานเมื่อเปิดโปรแกรมการจำแนกประเภทเว็บเพจโดยการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีนจะปรากฏหน้าจอให้ผู้ใช้ทำงานดังภาพประกอบ 4.6

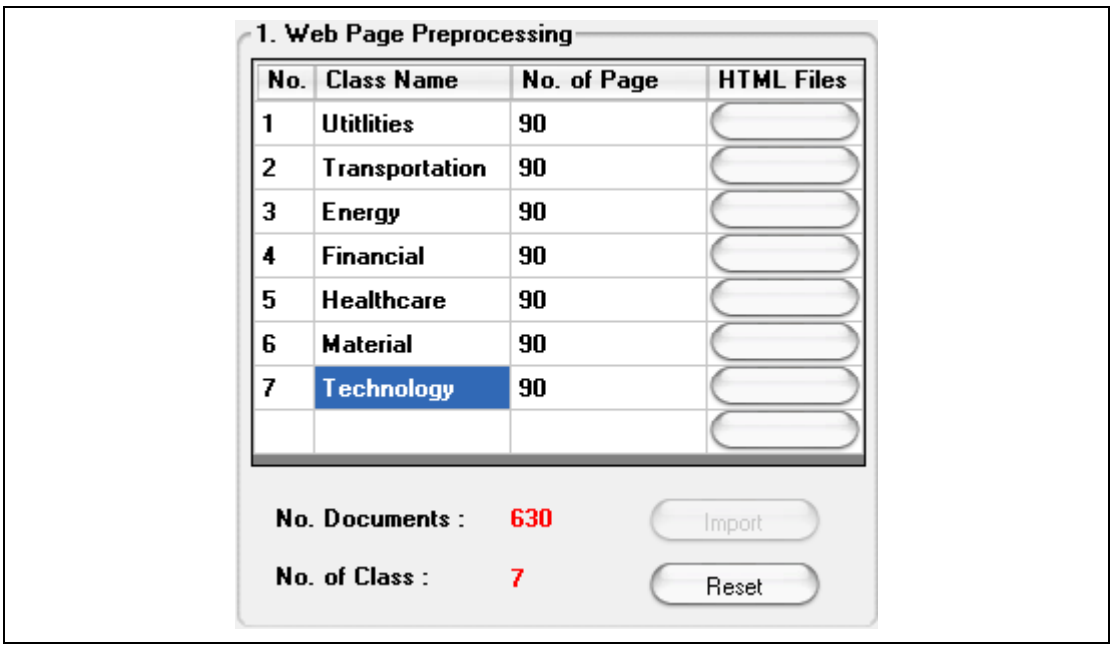


ภาพประกอบ 4.6 หน้าจอหลักของโปรแกรม

หน้าจอหลักของโปรแกรมประกอบด้วยองค์ประกอบหลัก 3 ส่วน คือ ส่วนการทำงาน ส่วนการแสดงผลค่าประสิทธิภาพ และส่วนการแสดงผลการทำนายคลาส โดยในแต่ละส่วนมีรายละเอียดดังนี้

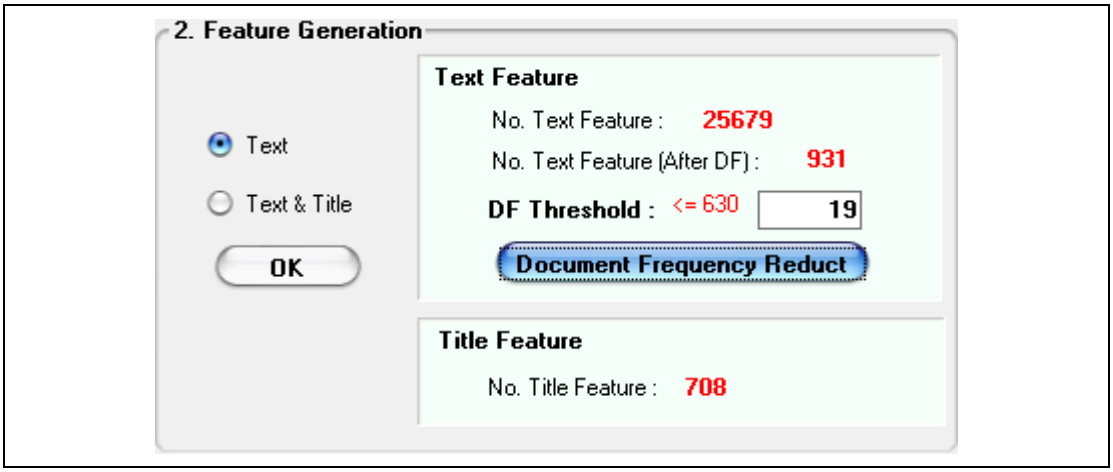
4.2.1 ส่วนการทำงานมี 4 ขั้นตอนย่อย คือ

1) Web Page Processing เป็นส่วนที่ทำหน้าที่นำเข้าข้อมูลเว็บเพจซึ่งอยู่ในรูปแบบไฟล์ htm, html โดยการกดปุ่ม ที่อยู่ในคอลัมน์ HTML Files ผู้ใช้ต้องระบุชื่อของคลาสที่นำเข้าในคอลัมน์ Class Name เมื่อป้อนข้อมูลครบทุกคลาส กดปุ่ม Import โปรแกรมจะทำการสกัดข้อความและหัวเรื่องจากหน้าเว็บเพจ หารากศัพท์และตัดคำ Stoplist โดยอัตโนมัติ พร้อมแสดงจำนวนเว็บเพจและจำนวนคลาสทั้งหมด ดังแสดงภาพประกอบ 4.7



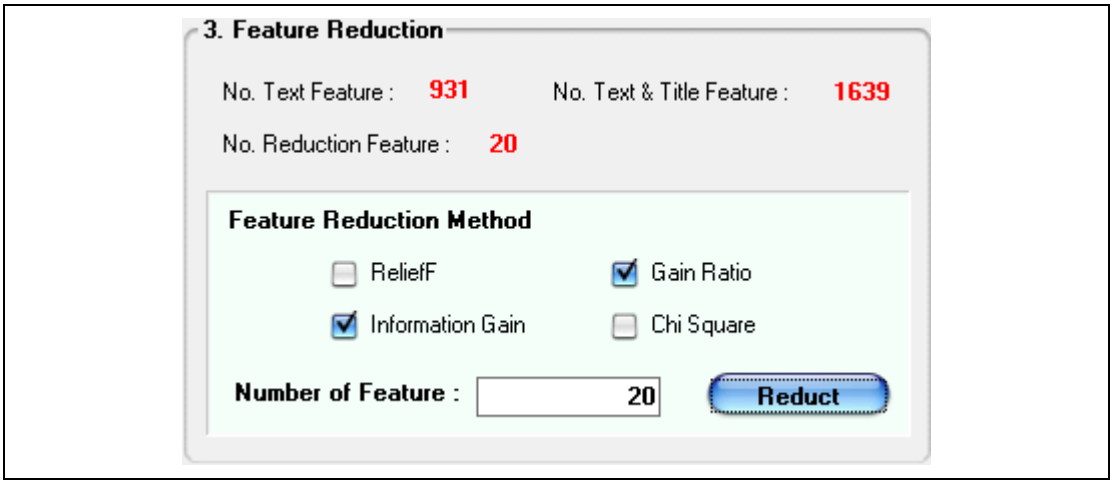
ภาพประกอบ 4.7 หน้าจอของส่วน Web Page Preprocessing

2) Feature Generation เป็นส่วนที่ทำหน้าที่แปลงข้อมูลนำเข้า มาสร้างให้อยู่ในรูปของลักษณะเฉพาะ เพื่อให้อยู่ในรูปแบบที่ระบบสามารถจำแนกประเภทได้ ซึ่งสามารถเลือกประเภทของข้อมูลได้ 2 แบบคือ ข้อมูลที่ใช้เฉพาะข้อความ (Text) และข้อมูลที่ใช้ข้อความกับหัวเรื่อง (Text & Title) โดยการกดปุ่ม OK ส่วนนี้โปรแกรมได้แสดงจำนวนของลักษณะเฉพาะจากข้อความและหัวเรื่องไว้ และสำหรับส่วนของข้อความผู้ใช้ต้องใส่ตัวเลขของ Document Frequency Threshold ในช่อง DF Threshold ซึ่งสามารถใส่เฉพาะตัวเลขเท่านั้น และค่านี้ต้องอยู่ระหว่าง 0 ถึง จำนวนของเว็บเพจทั้งหมด จากนั้นจึงกดปุ่ม Document Frequency Reduct โปรแกรมจะแสดงจำนวนของฟีเจอร์ที่เหลือ (After DF) ตัวอย่างการทำงานแสดงดังภาพประกอบ 4.8



ภาพประกอบ 4.8 หน้าจอของส่วน Feature Generation

3) Feature Reduction เป็นส่วนที่ทำหน้าที่ลดขนาดของลักษณะเฉพาะที่สร้างมาให้มีจำนวนน้อยลงโดยผู้ใช้สามารถเลือก 2 วิธีจาก 4 วิธีเพื่อเป็นลักษณะเฉพาะสำหรับการจำแนกประเภทและการให้คะแนนเสียงต่อไป โดยประกอบด้วยวิธีดังต่อไปนี้ ReliefF, Information Gain, Gain Ratio และ Chi Sqaure จากนั้นระบุจำนวนของลักษณะเฉพาะที่ต้องการในช่อง Number of feature ซึ่งต้องเป็นตัวเลขเท่านั้น และค่านี้ต้องอยู่ในช่วง 0 ถึงจำนวนลักษณะเฉพาะทั้งหมด จากนั้นจึงกดปุ่ม Reduct ตัวอย่างการทำงานแสดงดังภาพประกอบ 4.9



ภาพประกอบ 4.9 หน้าจอของส่วน Feature Reduction

4) Classification & Voting เป็นส่วนที่ทำหน้าที่จำแนกประเภทจากลักษณะเฉพาะทั้งสองที่ได้จากขั้นตอนก่อนหน้า แล้วนำผลลัพธ์ของทั้งสองลักษณะเฉพาะนี้หาผลลัพธ์สุดท้ายของการจำแนกโดยอัลกอริทึมการให้คะแนนเสียง (Voting algorithm) ส่วนนี้โปรแกรมได้แสดงข้อมูลของวิธีในการลดขนาดของลักษณะเฉพาะทั้ง Leftfeature และ Rightfeature ต่อมาผู้ใช้สามารถเลือกได้ 2 วิธี คือ 1vs1 และ 1vsAll จากนั้นผู้ใช้ป้อนตัวเลขของการทดสอบแบบไขว้เปลี่ยนในช่อง Cross Validation แล้วกดปุ่ม Classify ตัวอย่างการทำงานแสดงดังภาพประกอบ 4.10

ภาพประกอบ 4.10 หน้าจอของส่วน Classification and Voting

4.2.2 ส่วนการแสดงผลค่าประสิทธิภาพ เป็นส่วนที่นำผลลัพธ์ที่ได้จากการจำแนกทั้งสอง (Leftfeature และ Rightfeature) และผลลัพธ์ที่ได้จากอัลกอริทึมการให้คะแนนเสียง (Voting) โดยประกอบด้วย Precision, Recall และ F1 โดยโปรแกรมได้แสดงค่าประสิทธิภาพของทุกคลาสโดยค่าโดยเฉลี่ยด้วย ตัวอย่างการทำงานแสดงดังภาพประกอบ 4.11

Performance

		Precision			Recall			F1		
No.	Class Name	Left	Right	Voting	Left	Right	Voting	Left	Right	Voting
1	Utitlities	0.8889	0.8556	0.8889	0.8696	0.8750	0.9091	0.8791	0.8652	0.8989
2	Transportation	0.9111	0.9111	0.9667	0.8723	0.8723	0.8878	0.8913	0.8913	0.9255
3	Energy	0.9333	0.9222	0.9333	0.9333	0.8557	0.9130	0.9333	0.8877	0.9231
4	Financial	0.8667	0.8778	0.9000	0.9398	0.9294	0.9529	0.9017	0.9029	0.9257
5	Healthcare	0.8556	0.7889	0.8778	0.8105	0.8452	0.8876	0.8324	0.8161	0.8827
6	Material	0.8667	0.8444	0.8889	0.8571	0.7917	0.8421	0.8619	0.8172	0.8649
7	Technology	0.9111	0.8556	0.9222	0.9647	0.8953	1.0000	0.9371	0.8750	0.9595
Average :		0.8905	0.8651	0.9111	0.8925	0.8664	0.9132	0.8910	0.8650	0.9115

ภาพประกอบ 4.11 แสดงหน้าจอของส่วนของการแสดงค่าประสิทธิภาพ

4.2.3 ส่วนการแสดงผลการทำนายคลาส เป็นส่วนที่นำผลลัพธ์ที่ได้จากการจำแนกทั้งสอง (Leftfeature และ Rightfeature) และผลลัพธ์ที่ได้จากอัลกอริทึมการให้คะแนนเสียง (Voting) โดยแสดงรายละเอียดของแต่ละเว็บเพจว่าวิธีการจำแนก ทั้ง Leftfeature, Rightfeature และ Voting ได้จำแนกประเภทของเว็บเพจนั้น ๆ เป็นคลาสใด พร้อมทั้งแสดงค่าความถูกต้อง (Accuracy) ของแต่ละวิธีด้วย ตัวอย่างการทำงานแสดงดังภาพประกอบ 4.12 เช่น เว็บเพจที่ 16 ผลลัพธ์ที่ได้จาก Leftfeature และ Rightfeature มีความต่างกัน กล่าวคือ ผลลัพธ์ของ Leftfeature เป็นคลาส Utilities ผลลัพธ์ของ Rightfeature เป็นคลาส Energy อัลกอริทึมการให้คะแนนเสียงให้ผลลัพธ์ที่ถูกต้องตรงกับคลาสที่แท้จริง (Act. Class) คือคลาส Utilities

Predicted Class

No.	Act. Class	Left	Right	Voting
13	Utitlities	Utitlities	Utitlities	Utitlities
14	Utitlities	Utitlities	Utitlities	Utitlities
15	Utitlities	Utitlities	Utitlities	Utitlities
16	Utitlities	Utitlities	Energy	Utitlities
17	Utitlities	Utitlities	Utitlities	Utitlities
18	Utitlities	Utitlities	Energy	Utitlities
19	Utitlities	Financial	Utitlities	Utitlities
20	Utitlities	Utitlities	Energy	Utitlities
21	Utitlities	Utitlities	Utitlities	Utitlities
Accuracy :		86.8254	83.4921	90.1587

ภาพประกอบ 4.12 หน้าจอของส่วนของการทำนายคลาส

บทที่ 5

ผลการทดลองและวิจารณ์

บทนี้ได้นำเสนอผลลัพธ์ที่ได้จากการทดลองตามแบบจำลองการจำแนกประเภทเว็บเพจโดยวิธีการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน (Web Page Classification Model using Feature Reduction and Support Vector Machine: WPC_FRSVM) การทดลองใช้ชุดข้อมูล 2 ชุดข้อมูล คือข้อมูลจาก Yahoo และข้อมูลมาตรฐาน CMU การทดลองจะแบ่งเป็นการทดลองการลดขนาดลักษะเฉพาะโดยใช้ค่าความถี่เอกสาร และการทดลองโดยใช้วิธีการให้คะแนนเสียงจากมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีน ในการทดลองการลดขนาดลักษะเฉพาะโดยใช้ค่าความถี่เอกสารนั้นจะทำการทดลองเพื่อทดสอบประสิทธิภาพของการจำแนกประเภทเมื่อกำหนดค่าความถี่เอกสารในจำนวนที่แตกต่างกัน และการทดลองโดยใช้วิธีการให้คะแนนเสียงจากมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีนนั้น จะทำการทดลองเพื่อทดสอบประสิทธิภาพของการลดขนาดลักษณะเฉพาะด้วยเทคนิควิธีต่าง ๆ กับอัลกอริทึมการให้คะแนนเสียง

5.1 ชุดข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการทดลอง

5.1.1 ชุดข้อมูลจากเว็บไซต์ Yahoo

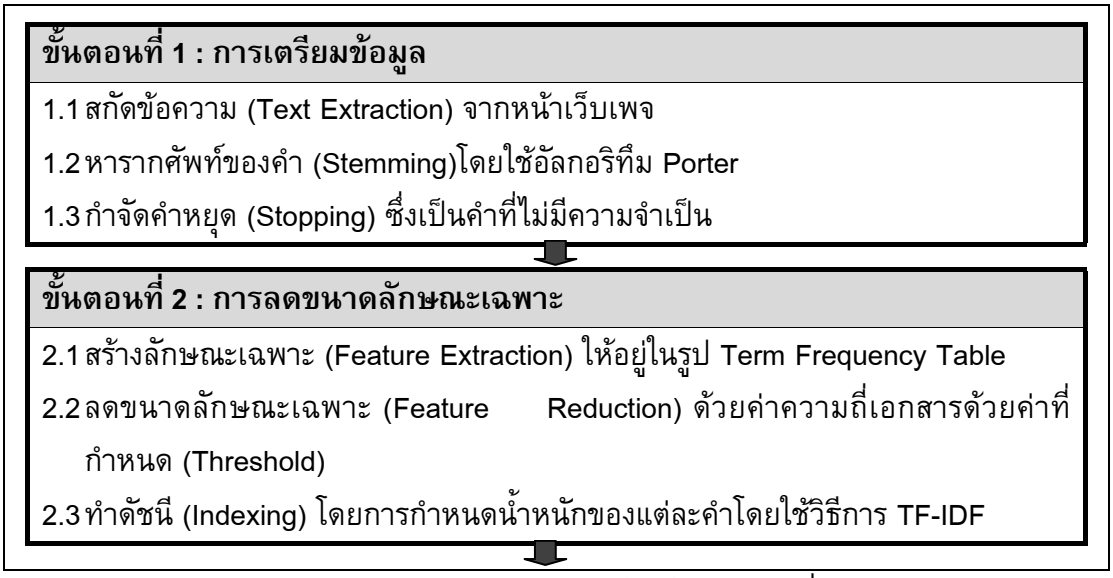
เอกสารเว็บเพจจากเว็บไซต์ Yahoo (http://www.yahoo.com: 2 มิถุนายน 2551) ซึ่งมีเนื้อหาเกี่ยวกับข่าวกีฬาจำนวน 5 ประเภท ได้แก่ 1) Boxing 2) Golf 3) Soccer 4) Tennis และ 5) Cycling โดยใช้ข้อมูลประเภทละ 50 เว็บเพจ รวมเป็น 250 เว็บเพจ

5.1.2 ชุดข้อมูลจาก CMU

เอกสารเว็บเพจ CMU ซึ่งอยู่ในกลุ่มของ WebKB (http://www.cs.cmu.edu/~WebKB/) เฉพาะส่วนที่เป็นไฟล์ html, htm ประกอบด้วย 7 ประเภทได้แก่ 1) Utilities จำนวน 239 เว็บเพจ 2) Energy จำนวน 291 เว็บเพจ 3) Healthcare จำนวน 321 เว็บเพจ 4) Transportation จำนวน 402 เว็บเพจ 5) Financial จำนวน 718 เว็บเพจ 6) Technology จำนวน 785 เว็บเพจ 7) Material จำนวน 777 เว็บเพจ รวมจำนวน 3,533 เว็บเพจ

5.2 การทดลองการลดขนาดลักษณะเฉพาะโดยใช้ค่าความถี่เอกสาร

การทดลองการจำแนกเว็บเพจโดยใช้ค่าความถี่เอกสาร ร่วมกับซัพพอร์ตเวกเตอร์แมชีน (Web Page Classification Using Document Frequency and Support Vector Machine: WPC_DF_SVM) ประกอบด้วยขั้นตอนทั้งหมด 3 ขั้นตอนดังภาพประกอบ 5.1 คือ ขั้นตอนที่ 1 การเตรียมข้อมูลเพื่อนำข้อความจากเว็บเพจมาผ่านกระบวนการ ให้อยู่ในลักษณะที่เหมาะสมในการวิเคราะห์ข้อมูลสำหรับดำเนินการในขั้นตอนต่อไป ขั้นตอนที่ 2 คือ การลดขนาดลักษณะเฉพาะ เป็นขั้นตอนที่นำคำที่ได้มาทำดัชนี ซึ่งผ่านการลดขนาดลักษณะเฉพาะด้วยค่าความถี่เอกสารตามค่า Threshold ที่ได้กำหนดไว้ ด้วยวิธีการ TF-IDF และขั้นตอนที่ 3 คือการประเมินประสิทธิผล เป็นขั้นตอนการประเมินผลด้วยค่า F-measure ของตัวจำแนก 3 ประเภทได้แก่ SVM ต้นไม้ตัดสินใจ C4.5 และ RBFNN การทดลองได้ใช้เอกสารเว็บเพจ 2 ชุดข้อมูล เพื่อทำการทดลอง โดยชุดข้อมูลแรกเป็นเอกสารเว็บเพจจาก CMU ซึ่งนิยมใช้กันอย่างแพร่หลายสำหรับการจำแนกประเภทเว็บเพจ ประกอบด้วยข้อมูล 7 ประเภท โดยการสุ่มจำนวน 271 เว็บเพจ จากข้อมูลทั้งหมด 8,282 เว็บเพจ ได้แก่ 1) Material จำนวน 49 เว็บเพจ 2) Energy จำนวน 39 เว็บเพจ 3) Financial จำนวน 21 เว็บเพจ 4) Healthcare จำนวน 43 เว็บเพจ 5) Technology จำนวน 45 เว็บเพจ 6) Transportation 36 เว็บเพจ และ 7) Utilities 38 เว็บเพจ ชุดข้อมูลที่สองเอกสารเว็บเพจจากเว็บไซต์ Yahoo [15] ซึ่งเป็นเว็บเพจที่มีเฉพาะเนื้อหาเกี่ยวกับข่าวกีฬาจำนวน 5 ประเภท ได้แก่ 1) Boxing 2) Golf 3) Soccer 4) Tennis และ 5) Cycling โดยใช้ข้อมูลประเภทละ 50 เว็บเพจ รวมเป็น 250 เว็บเพจ ทำการทดสอบแบบ 10-Fold Cross Validation โดยใช้ WEKA



ภาพประกอบ 5.1 แบบจำลองการจำแนกเว็บเพจโดยใช้ค่าความถี่เอกสารร่วมกับ SVM

ขั้นตอนที่ 3 : การประเมินประสิทธิผล
3.1 จำแนกประเภทด้วยตัวจำแนกประเภท 3 ประเภทได้แก่ Support Vector Machine ต้นไม้ตัดสินใจ C4.5 และ RBFNN
3.2 ประเมินประสิทธิผลค่า F-measure ด้วยการเปรียบเทียบตัวจำแนกทั้ง 3 ประเภท

ภาพประกอบ 5.1 แบบจำลองการจำแนกเว็บเพจโดยใช้ค่าความถี่เอกสารร่วมกับ SVM (ต่อ)

5.2.1 ขั้นตอนที่ 1 การเตรียมข้อมูล

5.2.1.1 การสกัดข้อความ (Text Extraction)

การสกัดข้อความโดยใช้โปรแกรม HTMLAsText [7] เป็นเครื่องมือในการสกัดข้อความ ซึ่งจะสกัดเฉพาะส่วนของข้อความ (Text) ที่ปรากฏอยู่บนหน้าเว็บเพจเท่านั้น หลังจากใช้โปรแกรม HTMLAsText แล้วผลลัพธ์ที่ได้จะเป็นเนื้อความของเอกสารเว็บเพจนั้นซึ่งจะไม่มีส่วน HTML tag รูปภาพ หรือสื่อมัลติมิเดีย ปรากฏอยู่

5.2.1.2 การหารากศัพท์ (Stemming)

เมื่อได้ข้อความจากขั้นตอนการสกัดคำแล้ว ต่อไปจึงนำคำเหล่านี้มาหารากศัพท์โดยใช้อัลกอริทึม Porter ซึ่งเป็นอัลกอริทึมที่ได้รับความนิยม โดยปรับเปลี่ยนคำท้าย (Suffix) ในภาษาอังกฤษ แล้วจึงกำหนดให้ตัวอักษรตัวเล็กทั้งหมด เช่นคำว่า “Run” และ “Running” จะเปลี่ยนเป็นรากศัพท์เดียวกันคือ “run” เป็นต้น

5.2.1.3 การกำจัดคำหยุด (Stopping)

การกำจัดคำหยุด เป็นกระบวนการหนึ่งที่สำคัญสำหรับการค้นคืนสารสนเทศ เนื่องจากคำบางคำไม่มีความจำเป็นสำหรับการนำมาวิเคราะห์ ดังนั้นจึงควรกำจัดคำเหล่านี้ออกไปเพื่อให้เหลือเฉพาะคำที่มีความสำคัญเท่านั้น คำเหล่านี้เรียกว่าคำ “Stoplist” เช่น คำว่า “a” “been” และ “ago” เป็นต้น

5.2.2 ขั้นตอนที่ 2 การลดขนาดลักษณะเฉพาะ

5.2.2.1 การสร้างลักษณะเฉพาะ (Feature Extraction)

โดยนำคำที่ได้จากขั้นตอนการเตรียมข้อมูลมาสร้างให้อยู่ในรูปของ Term Frequency Table ตัวอย่างเช่น มีเอกสารเว็บเพจข่าวกีฬาจำนวน 3 เว็บเพจ และมีคำอยู่ 4 คำ คือ play, tennis, win และ knockout สามารถนำมาสร้างให้อยู่ในรูปของ Term Frequency และ Document Frequency ดังตารางที่ 5.1 โดยที่กำหนดให้ *TF* (Term Frequency) หมายถึงค่าความถี่ของคำแต่ละคำที่ปรากฏอยู่ในเอกสาร ตัวอย่างเช่น คำว่า “play” ปรากฏอยู่ใน

เอกสารฉบับที่ 1 (Doc_1) จำนวน 3 ครั้ง ปรากฏอยู่ในเอกสารฉบับที่ 2 (Doc_2) จำนวน 4 ครั้ง และปรากฏอยู่ในเอกสารฉบับที่ 3 (Doc_3) จำนวน 0 ครั้งหรือไม่ปรากฏอยู่ในเอกสารฉบับนี้ เป็นต้น โดย *DF* (Document Frequency) หมายถึง ความถี่เอกสารของคำที่พิจารณา ปรากฏในเอกสารทั้งหมดกี่ฉบับ เช่น คำว่า “play” ปรากฏอยู่ในเอกสาร 2 ฉบับ ได้แก่ เอกสารฉบับที่ 1 (Doc_1) และเอกสารฉบับที่ 2 (Doc_2) ดังนั้น จึงมีค่าค่าความถี่เอกสารเป็น 2 และคำว่า “tennis” ปรากฏอยู่ในเอกสารทั้งหมด 3 ฉบับ ได้แก่ เอกสารฉบับที่ 1 เอกสารฉบับที่ 2 และเอกสารฉบับที่ 3 จึงมีค่าความถี่เอกสารเป็น 3 เป็นต้น

ตารางที่ 5.1 ตัวอย่าง Term Frequency Table

เอกสาร	**ความถี่ของคำ *TF* (Term Frequency) ในแต่ละเอกสาร (Doc)**			
	play	**tennis**	**win**	**knockout**
Doc_1	3	5	0	1
Doc_2	4	2	0	1
Doc_3	0	1	3	1
ความถี่เอกสาร *DF* (Document Frequency)	**2**	**3**	**1**	**3**

5.2.2.2 การลดขนาดลักษณะเฉพาะ (Feature Reduction)

เนื่องจากขนาดลักษณะเฉพาะของการจำแนกประเภทเว็บเพจมีจำนวนมาก การลดจำนวนลักษณะเฉพาะจึงเป็นสิ่งจำเป็นอย่างยิ่ง การทดลองนี้ได้ใช้ค่าความถี่ของเอกสาร (Document Frequency) ที่ลักษณะเฉพาะนั้นปรากฏอยู่ โดยจะเลือกลักษณะเฉพาะที่มากกว่าหรือเท่ากับค่า Threshold ที่กำหนดไว้ ถ้ากำหนดค่า Threshold (λ) ให้มีค่ามากกว่าหรือเท่ากับ 3 จากตารางที่ 3 ค่าความถี่เอกสาร (Document Frequency) ของคำว่า (“play” = 2, “tennis” = 3, “win” = 1, “knockout” = 3) เราจะได้คำที่ผ่านค่า λ เพียง 2 คำ คือคำว่า “tennis” และ “knockout” เพื่อใช้ในการทำงานในขั้นต่อไป และตัดคำว่า “play” และ “win” ทิ้งไป

5.2.2.3 การทำดัชนี (Indexing)

เมื่อผ่านการลดขนาดลักษณะเฉพาะแล้วจะนำคำที่ได้เหล่านั้นมาให้ค่าน้ำหนักโดยใช้วิธีการ TF-IDF โดย *TF* (Term Frequency) คือความถี่ของแต่ละคำที่ปรากฏในเอกสาร และ *IDF* (Inverse Document Frequency) คือ ส่วนกลับของความถี่เอกสารที่ปรากฏคำนั้นอยู่ ดังภาพประกอบ 3.5 โดยที่ W_{jk} คือ ค่า TF-IDF ของคำที่ k ในเอกสารที่ j โดย j = 1,

…, *n* เมื่อ *n* คือ จำนวนของเอกสารทั้งหมด จากนั้นเมื่อคำนวณ ค่า TF-IDF ของทุกคำแล้ว นำค่าที่ได้จากการคำนวณนี้มาสร้างให้อยู่ในรูปแบบ Document Word Matrix

5.2.3 ขั้นตอนที่ 3 การประเมินประสิทธิผล

5.2.3.1 การจำแนกประเภทเว็บเพจ (Web Page Classification)

การทดลองนี้ใช้ตัวจำแนกประเภท (Classifier) 3 ประเภท คือ Support Vector Machine (SVM) ต้นไม้ตัดสินใจ C4.5 และ RBF Neural Networks

5.2.3.2 ประเมินประสิทธิผลค่า F-measure

จากผลการทดลองทั้ง 2 ชุดข้อมูล โดยทดสอบกับค่าความถี่เอกสารต่าง ๆ เปรียบเทียบกับตัวจำแนกประเภททั้ง 3 ประเภทตามขั้นตอนการจำแนกเว็บเพจ WPC_DF_SVM

จากนั้นจึงนำค่า F-measure ที่ได้ของทุกประเภท (Class) มาหาค่าเฉลี่ย โดยสามารถพิจารณาประสิทธิภาพของขั้นตอนวิธีดังประเด็นต่อไปนี้

1) ประสิทธิภาพค่าความถูกต้องการลดขนาดลักษณะเฉพาะ

จากตารางที่ 5.3 ของข้อมูลชุดที่ 1 เมื่อลดขนาดลักษณะเฉพาะโดยใช้ความถี่เอกสาร (Document Frequency) ที่มีค่า Threshold (λ) ต่างๆ คือ $\geq 0, \geq 5, \ldots, \geq 50$ ตามลำดับ สามารถลดจำนวณลักษณะเฉพาะจาก 12,722 จำนวนเหลือเพียง 88 จำนวน และจากตารางที่ 5.4 ของข้อมูลชุดที่ 2 สามารถลดจำนวณลักษณะเฉพาะจาก 9,608 จำนวนเหลือเพียง 492 จำนวนเท่านั้น โดยพบว่าจากข้อมูลชุดที่ 1 SVM ยังคงให้ค่าความถูกต้องที่ดี โดยมีค่าความต่างระหว่างค่าเฉลี่ย F-measure มากที่สุด 93.19% และค่าเฉลี่ย F-measure น้อยที่สุด 88.44% มีค่าประมาณ 5% ผลการทดลองข้อมูลชุดที่ 2 SVM มีค่าเฉลี่ย F-measure 100% ทั้งในกรณีก่อนการลดขนาดลักษณะเฉพาะและหลังการลดขนาดลักษณะเฉพาะ

2) ประสิทธิภาพของตัวจำแนกประเภท

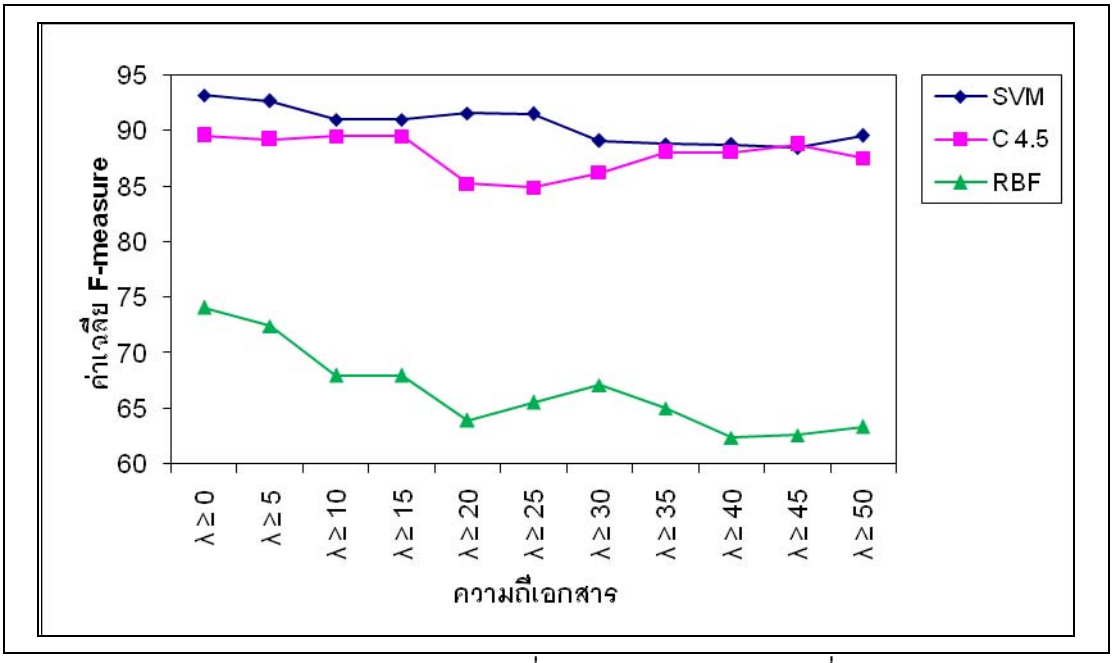
เมื่อพิจาณาตัวจำแนกทั้ง 3 ประเภท จากข้อมูลชุดที่ 1 WebKB ภาพประกอบ 5.2 แสดงให้เห็นว่าเมื่อเลือก λ มากกว่าหรือเท่ากับ 0 จนถึงค่า λ มากกว่าหรือเท่ากับ 50 SVM ให้ค่า F-measure สูงที่สุด รองลงมาคือ C4.5 และ RBFNN ให้ค่า F-measure น้อยที่สุด จากชุดข้อมูลที่ 2 เว็บเพจข่าวกีฬา Yahoo ภาพประกอบ 5.3 ที่ขนาดลักษณะเฉพาะมากที่สุด ($\lambda \geq 0$) SVM ให้ค่าดีที่สุด คือ 100% ทุกกรณี C4.5 ให้ค่า 99.6% ทุกกรณี และ RBFNN ให้ค่า 92.95% โดยมีค่าเพิ่มขึ้นเป็น 100% เมื่อค่า $\lambda \geq 20$ ซึ่งพบว่าเมื่อลดจำนวนลักษณะเฉพาะลงจะทำให้ตัวจำแนกประเภทสามารถทำงานได้ดียิ่งขึ้น

ตารางที่ 5.2 ค่าเฉลี่ย F-measure ชุดข้อมูลที่ 1 เว็บเพจ WebKB

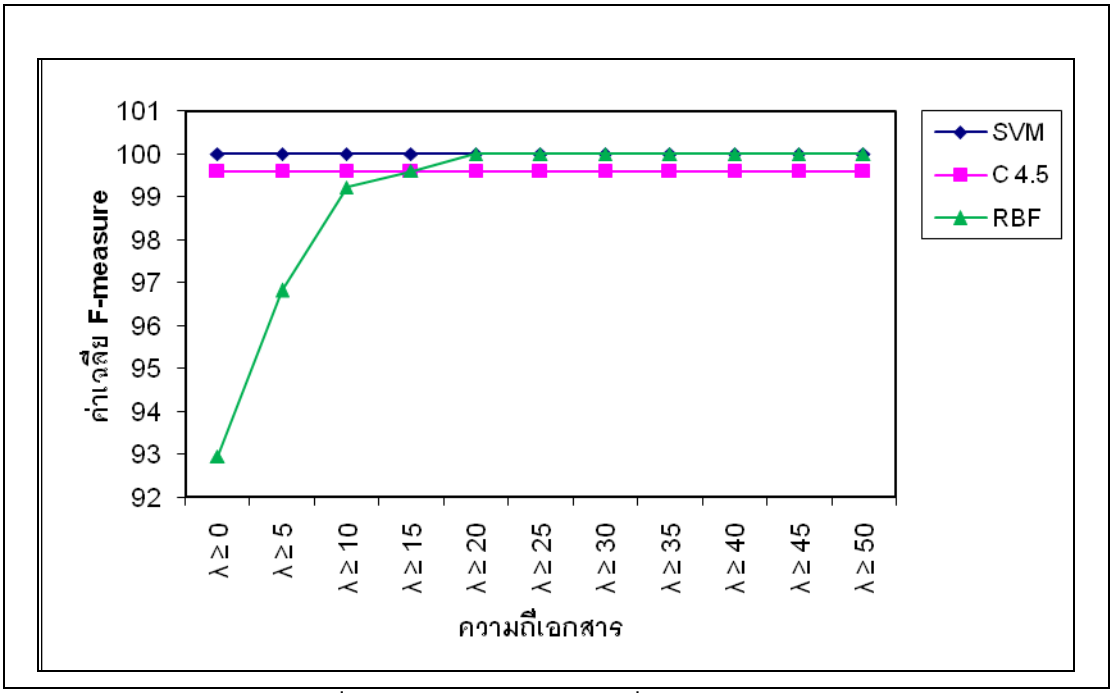
ความถี่เอกสาร (Document Frequency)	จำนวนลักษณะเฉพาะ	ค่าเฉลี่ย F-measure (%)		
		SVM	C4.5	RBF
$\lambda \geq 0$	12,722	**93.19**	**89.57**	**74.06**
$\lambda \geq 5$	1,666	92.68	89.22	72.42
$\lambda \geq 10$	812	90.98	89.50	67.96
$\lambda \geq 15$	511	90.98	89.50	67.96
$\lambda \geq 20$	357	91.57	85.22	63.91
$\lambda \geq 25$	262	91.53	84.90	65.55
$\lambda \geq 30$	187	89.10	86.23	67.08
$\lambda \geq 35$	144	88.78	88.06	65.04
$\lambda \geq 40$	120	88.75	88.08	62.39
$\lambda \geq 45$	102	88.44	88.74	62.62
$\lambda \geq 50$	88	89.56	87.56	63.33

ตารางที่ 5.3 ค่าเฉลี่ย F-measure ชุดข้อมูลที่ 2 เว็บเพจข่าวเว็บไซต์ Yahoo

ความถี่เอกสาร (Document Frequency)	จำนวนลักษณะเฉพาะ	ค่าเฉลี่ย F-measure (%)		
		SVM	C4.5	RBF
$\lambda \geq 0$	9,608	100.00	99.60	92.95
$\lambda \geq 5$	2,285	100.00	99.60	96.82
$\lambda \geq 10$	1,364	100.00	99.60	99.22
$\lambda \geq 15$	1,019	100.00	99.60	99.60
$\lambda \geq 20$	803	100.00	99.60	100.00
$\lambda \geq 25$	691	100.00	99.60	100.00
$\lambda \geq 30$	617	100.00	99.60	100.00
$\lambda \geq 35$	565	100.00	99.60	100.00
$\lambda \geq 40$	533	100.00	99.60	100.00
$\lambda \geq 45$	511	100.00	99.60	100.00
$\lambda \geq 50$	492	100.00	99.60	100.00



ภาพประกอบ 5.2 ค่าเฉลี่ย F-measure ชุดข้อมูลที่ 1 WebKB



ภาพประกอบ 5.3 ค่าเฉลี่ย F-measure ชุดข้อมูลที่ 2 เว็บเพจข่าวกีฬา เว็บไซต์ Yahoo

5.3 การทดลองโดยใช้วิธีการให้คะแนนเสียงจากมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีน

การทดลองการจำแนกเว็บเพจโดยใช้วิธีการให้คะแนนเสียงจากมัลติคลาสซัพพอร์ตเวกเตอร์แมชชีน (Web Page Classification using A novel Voting Algorithm of Multi-class SVM: WPC_VAMSVM) ประกอบด้วยขั้นตอนทั้งหมด 4 ขั้นตอนดังภาพประกอบ 3.1 (บทที่ 3) ขั้นตอนที่ 1 คือ การเตรียมข้อมูลเว็บเพจ ขั้นตอนที่ 2 คือการสร้างลักษณะเฉพาะ ขั้นตอนที่ 3 การลดขนาดลักษณะเฉพาะ ขั้นตอนที่ 4 การจำแนกประเภทและการให้คะแนนเสียง การทดลองได้ใช้เอกสารเว็บเพจจากชุดข้อมูลมาตรฐาน CMU ประกอบด้วยข้อมูล 7 ประเภท ได้แก่ 1) Material 2) Energy 3) Financial 4) Healthcare 5) Technology 6) Transportation 36 และ 7) Utilities จำนวนคลาสละ 90 เว็บเพจ รวมจำนวน 630 เว็บเพจจาก 3,533 (เฉพาะ html) เว็บเพจ จำนวนของลักษณะเฉพาะจากข้อความมี 22,938 จำนวน และจำนวนของลักษณะเฉพาะจากข้อความมี 716 จำนวน หลังจากเลือกลักษณะเฉพาะด้วย document frequency threshold จำนวนของลักษณะเฉพาะจากข้อความมี 1,033 จำนวน และ จำนวนของลักษณะเฉพาะจากข้อความและหัวเรื่องมี 1,749 จำนวน การทดลองนี้ได้ออกแบบรูปแบบการทดลองที่ต่างกัน ได้แก่ A, B, C และ D เพื่อทดสอบกับวิธี 10-Fold Cross Validation ดังตารางที่ 5.4

ตารางที่ 5.4 รูปแบบการทดลองของ WPC_VAMSVM

Exp.	Experiment Name	Feature		SVM and Voting	
		Text Only	Text&Title	1vs1	1vsAll
A	TO_1vs1	✓	×	✓	×
B	TO_1vsAll	✓	×	×	✓
C	TT_1vs1	×	✓	✓	×
D	TT_1vsAll	×	✓	×	✓

ตารางที่ 5.5 ค่า F-measure (%) ของการทดลองรูปแบบ A: TO_1vs1

No. Features	1vs1 SVM Classification				1vs1_Voting					
	RF	*IG*	*CS*	*GR*	*RF&IG*	*RF&CS*	*RF&GR*	*IG&GR*	*IG&CS*	*CS&GR*
100	89.10	86.50	87.79	79.61	91.15	92.73	91.91	87.36	89.05	87.65
150	89.41	84.19	82.19	83.50	90.30	90.35	90.50	87.69	84.57	87.56
200	88.92	85.75	84.81	82.41	89.22	89.08	87.69	85.57	85.73	84.02
250	87.31	85.13	85.14	85.13	87.81	87.63	87.81	85.11	85.14	85.43
300	87.49	84.67	84.67	84.67	87.15	87.15	87.15	84.67	84.67	84.67

ตารางที่ 5.5 ค่า F-measure (%) ของการทดลองรูปแบบ A: TO_1vs1 (ต่อ)

No. Features	1vs1 SVM Classification				1vs1_Voting					
	RF	***IG***	***CS***	***GR***	***RF&IG***	***RF&CS***	***RF&GR***	***IG&GR***	***IG&CS***	***CS&GR***
350	86.16	83.41	83.41	83.41	85.83	85.83	85.83	83.41	83.41	83.41
400	86.80	81.59	81.59	81.59	84.61	84.61	84.61	81.59	81.59	81.59
all	80.16	80.16	80.16	80.16	80.16	80.16	80.16	80.16	80.16	80.16

ตารางที่ 5.6 ค่า F-measure (%) ของการทดลองรูปแบบ B: TO_1vsAll

No. Features	1vsAll SVM Classification				1vsAll_Voting					
	RF	***IG***	***CS***	***GR***	***RF&IG***	***RF&CS***	***RF&GR***	***IG&GR***	***IG&CS***	***CS&GR***
100	91.44	86.50	87.50	80.14	91.44	92.37	89.34	84.78	87.01	86.49
150	91.28	85.43	83.49	83.13	91.28	91.89	90.48	87.41	86.82	87.07
200	91.13	87.18	88.45	84.81	92.72	92.72	92.87	87.97	87.65	88.26
250	90.82	88.29	87.02	87.81	91.46	91.63	91.61	87.97	87.98	87.97
300	91.28	88.15	88.15	88.15	91.80	91.80	91.80	88.15	88.15	88.15
350	90.67	86.56	86.56	86.56	91.78	91.78	91.78	86.56	86.56	86.56
400	89.89	84.15	84.15	84.15	89.91	89.91	89.91	84.15	84.15	84.15
all	86.23	86.23	86.23	86.23	86.23	86.23	86.23	86.23	86.23	86.23

ตารางที่ 5.7 ค่า F-measure (%) ของการทดลองรูปแบบ C: TT_1vs1

No. Features	1vs1 SVM Classification				1vs1_Voting					
	RF	***IG***	***CS***	***GR***	***RF&IG***	***RF&CS***	***RF&GR***	***IG&GR***	***IG&CS***	***CS&GR***
100	90.35	87.00	87.46	77.09	91.63	92.73	90.52	88.10	88.57	88.43
150	89.70	84.35	87.27	81.32	90.22	91.31	91.30	89.22	86.65	89.10
200	89.21	84.76	84.76	81.07	90.82	91.16	90.62	86.67	85.23	86.64
250	89.06	86.24	85.45	83.58	90.36	90.23	90.05	86.60	86.36	86.75
300	88.29	85.29	85.12	83.60	88.13	87.66	88.99	85.17	85.44	85.01
350	87.99	84.97	85.72	84.78	88.47	88.31	88.47	85.12	86.08	85.42
400	88.46	84.05	83.53	84.47	86.74	87.96	87.35	84.68	85.43	85.13
all	81.57	81.57	81.57	81.57	81.57	81.57	81.57	81.57	81.57	81.57

ตารางที่ 5.8 ค่า F-measure (%) ของการทดลองรูปแบบ D: TT_1vsAll

No. Features	1vsAll SVM Classification				1vsAll_Voting					
	RF	***IG***	***CS***	***GR***	***RF&IG***	***RF&CS***	***RF&GR***	***IG&GR***	***IG&CS***	***CS&GR***
100	93.04	86.85	86.87	77.32	93.35	93.01	91.13	87.95	87.80	88.40
150	91.77	86.06	86.04	80.61	92.56	92.68	90.01	87.79	88.12	87.46
200	92.07	86.71	87.15	81.84	92.87	93.17	92.37	88.20	87.00	89.00
250	91.30	89.87	88.43	85.91	93.98	93.18	92.23	89.72	89.57	88.30
300	90.78	88.47	87.51	87.06	93.18	93.18	93.19	88.48	88.63	88.64
350	90.81	88.12	87.97	87.97	93.03	92.41	92.40	88.77	88.76	87.81
400	90.79	88.77	87.84	87.51	92.09	91.78	92.25	88.46	89.24	87.97
all	87.22	87.22	87.22	87.22	87.22	87.22	87.22	87.22	87.22	87.22

สามารถสรุปประเด็นการทดลองได้ 4 ประเด็นดังต่อไปนี้

1) ประเด็นของการเลือกวิธีในการลดขนาดลักษณะเฉพาะ

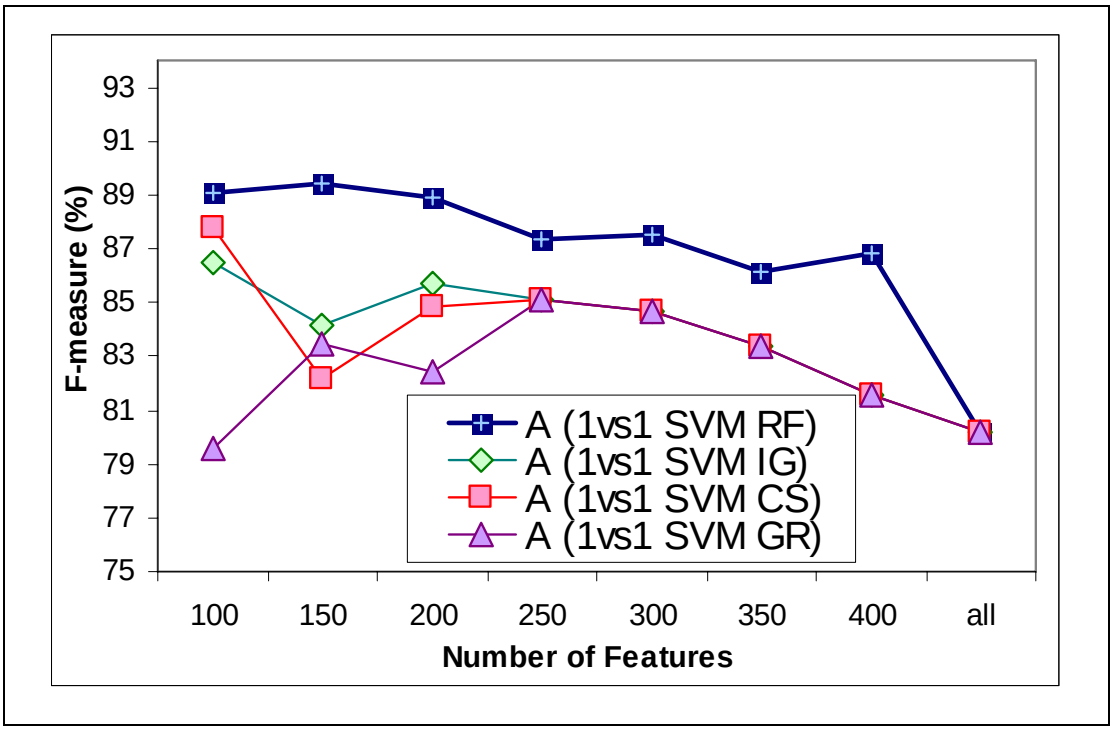
การลดขนาดลักษณะด้วยวิธี RF ให้ค่า F-measure ที่สูงที่สุดเมื่อเทียบกับวิธี IG, CS และ GR. ผลการทดลองที่ลักษณะเฉพาะจำนวน 100

สำหรับผลการทดลอง A ในตารางที่ 5.5 และภาพประกอบ 5.4 พบว่า RF ให้ค่า F-measure คือ 89.10% ขณะที่ IG ให้ค่า F-measure คือ 86.50%, CS ให้ค่า F-measure คือ 87.79% และ GR ให้ค่า F-measure คือ 79.61%

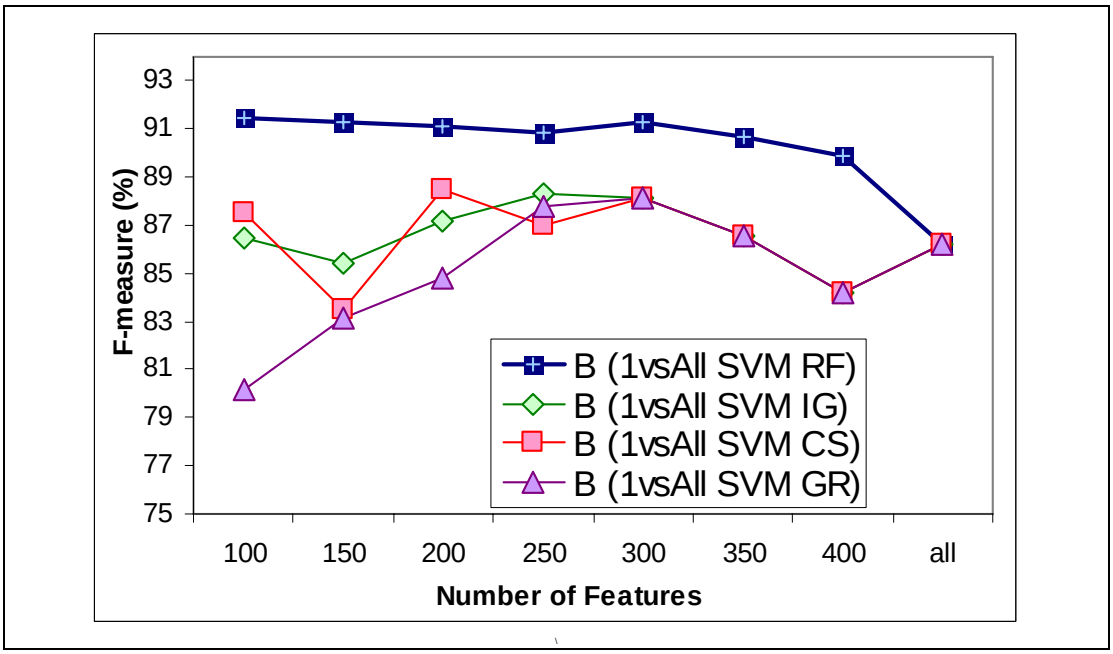
สำหรับผลการทดลอง B ในตารางที่ 5.6 และภาพประกอบ 5.5 พบว่า RF ให้ค่า F-measure สูงที่สุด คือ 91.44%

สำหรับผลการทดลอง C ในตารางที่ 5.7 และภาพประกอบ 5.6 พบว่า RF ให้ค่า F-measure สูงที่สุด คือ 90.35%

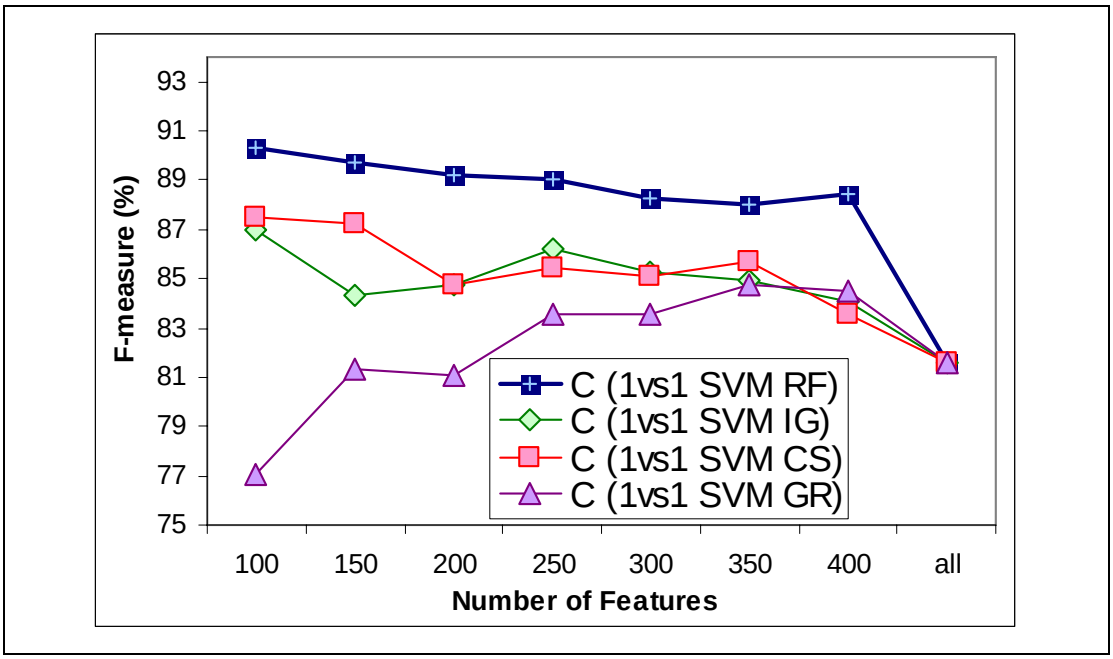
สำหรับผลการทดลอง D ในตารางที่ 5.8 และภาพประกอบ 5.7 พบว่า RF ให้ค่า F-measure สูงที่สุด คือ 93.04%.



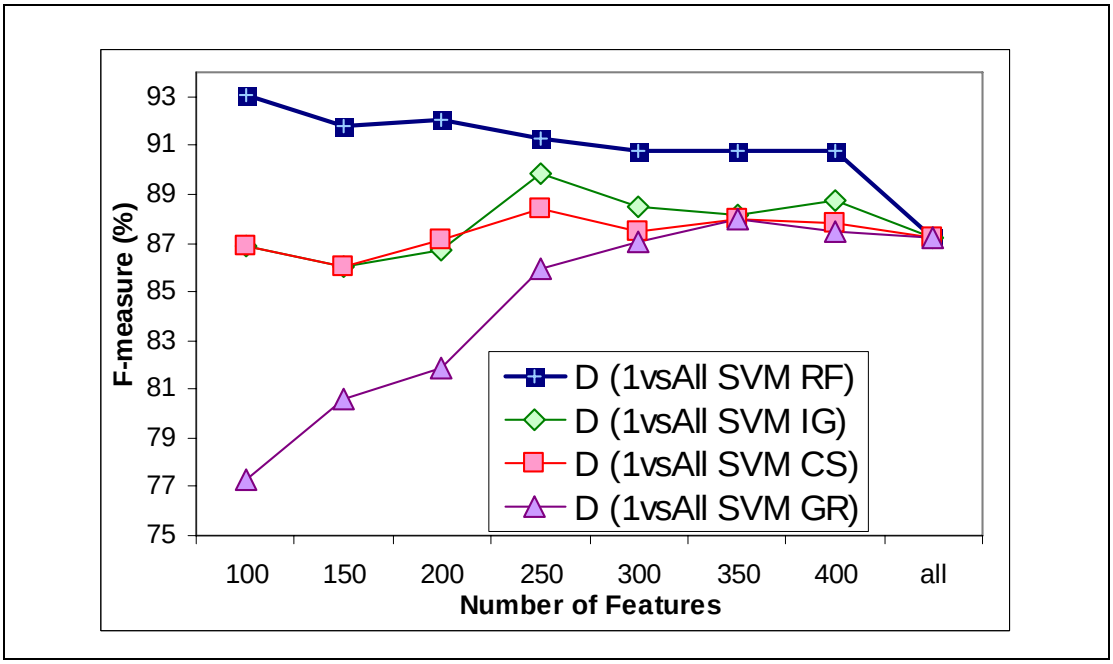
ภาพประกอบ 5.4 ค่า F-measure วิธีการเลือกลักษณะแบบต่าง ๆ ของการทดลอง A



ภาพประกอบ 5.5 ค่า F-measure วิธีการเลือกลักษณะแบบต่าง ๆ ของการทดลอง B



ภาพประกอบ 5.6 ค่า F-measure วิธีการเลือกลักษณะแบบต่าง ๆ ของการทดลอง C



ภาพประกอบ 5.7 ค่า F-measure วิธีการเลือกลักษณะแบบต่าง ๆ ของการทดลอง D

2) ประเด็นของการเลือกวิธีมัลติคลาสของซัพพอร์ตเวกเตอร์แมชชีน

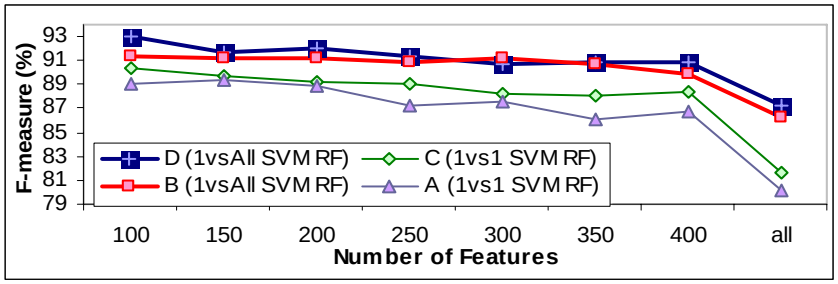
การเลือกวิธีมัลติคลาสแบบ 1vsAll ให้ค่า F-measure ที่สูงกว่าแบบ 1vs1

สำหรับผลการทดลองโดยการเลือกลักษณะเฉพาะแบบ RF ที่จำนวนลักษณะเฉพาะ 100 แสดงดังภาพ ประกอบ 5.8 เมื่อพิจารณาลักษณะเฉพาะจากข้อความเท่านั้น พบว่าค่า F-measure ของ 1vsAll (การทดลอง B ตารางที่ 5.6) คือ 91.44% ขณะที่ 1vs1 (การทดลอง A ตารางที่ 5.5) มีค่า F-measure คือ 89.10% เท่านั้น และเมื่อพิจารณาลักษณะเฉพาะจากข้อความและหัวเรื่อง พบว่าค่า F-measure ของ 1vsAll (การทดลอง D ตารางที่ 5.8) ให้ค่า F-measure ที่สูงกว่าคือ 93.04% ขณะที่ 1vs1 (การทดลอง C ตารางที่ 5.7) มีค่า F-measure คือ 90.35% เท่านั้น

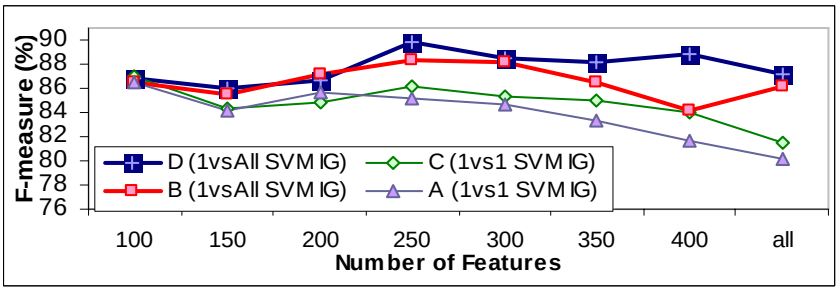
สำหรับผลการทดลองโดยการเลือกลักษณะเฉพาะแบบ IG ที่จำนวนลักษณะเฉพาะ 150 แสดงดังภาพประกอบ 5.9 เมื่อพิจารณาลักษณะเฉพาะจากข้อความเท่านั้น พบว่าค่า F-measure ของ 1vsAll (การทดลอง B ตารางที่ 5.6) คือ 85.43% ขณะที่ 1vs1 (การทดลอง A ตารางที่ 5.5) มีค่า F-measure คือ 84.19% เท่านั้น และเมื่อพิจารณาลักษณะเฉพาะจากข้อความและหัวเรื่อง พบว่าค่า F-measure ของ 1vsAll (การทดลอง D ตารางที่ 5.8) ให้ค่าที่สูงกว่าคือ 86.06% ขณะที่ 1vs1 (การทดลอง C ตารางที่ 5.7) มีค่า F-measure คือ 84.35% เท่านั้น

สำหรับผลการทดลองโดยการเลือกลักษณะเฉพาะแบบ CS ที่จำนวนลักษณะเฉพาะ 200 แสดงดังภาพประกอบ 5.10 เมื่อพิจารณาลักษณะเฉพาะจากข้อความเท่านั้น พบว่า ค่า F-measure ของ 1vsAll (การทดลอง B ตารางที่ 5.6) คือ 88.45% ขณะที่ 1vs1 (การทดลอง A ตารางที่ 5.5) มีค่า F-measure คือ 84.81% เท่านั้น และเมื่อพิจารณาลักษณะเฉพาะจากข้อความและหัวเรื่อง พบว่าค่า F-measure ของ 1vsAll (การทดลอง D ตารางที่ 5.8) ให้ค่าที่สูงกว่าคือ 87.15% ขณะที่ 1vs1 (การทดลอง C ตารางที่ 5.7) มีค่า F-measure คือ 84.76% เท่านั้น

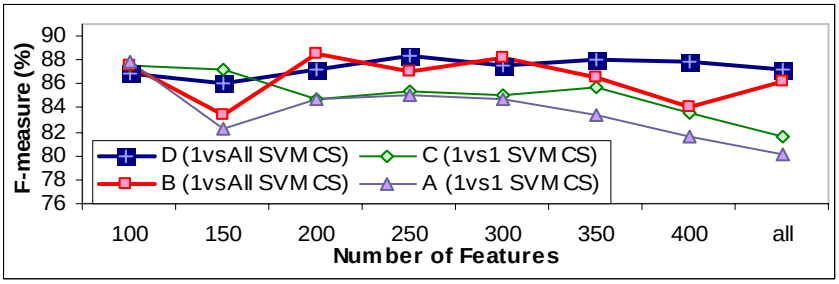
สำหรับผลการทดลองโดยการเลือกลักษณะเฉพาะแบบ GR ที่จำนวนลักษณะเฉพาะ 200 แสดงดังภาพประกอบ 5.11 เมื่อพิจารณาลักษณะเฉพาะจากข้อความเท่านั้น พบว่า ค่า F-measure ของ 1vsAll (การทดลอง B ตารางที่ 5.6) คือ 84.81% ขณะที่ 1vs1 (การทดลอง A ตารางที่ 5.5) มีค่า F-measure คือ 82.41% เท่านั้น และเมื่อพิจารณาลักษณะเฉพาะจากข้อความและหัวเรื่อง พบว่าค่า F-measure ของ 1vsAll (การทดลอง D ตารางที่ 5.8) ให้ค่าที่สูงกว่าคือ 81.84% ขณะที่ 1vs1 (การทดลอง C ตารางที่ 5.7) มีค่า F-measure คือ 81.07% เท่านั้น



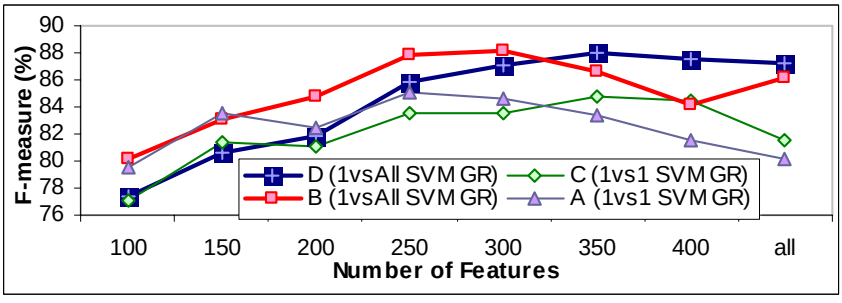
ภาพประกอบ 5.8 ค่า F-measure ของการทดลอง เมื่อใช้การลดขนาดลักษณะเฉพาะแบบ RF



ภาพประกอบ 5.9 ค่า F-measure ของการทดลอง เมื่อใช้การลดขนาดลักษณะเฉพาะแบบ IG



ภาพประกอบ 5.10 ค่า F-measure ของการทดลอง เมื่อใช้การลดขนาดลักษณะเฉพาะแบบ CS



ภาพประกอบ 5.11 ค่า F-measure ของการทดลอง เมื่อใช้การลดขนาดลักษณะเฉพาะแบบ GR

3) ประเด็นของอัลกอริทึมการให้คะแนนเสียง

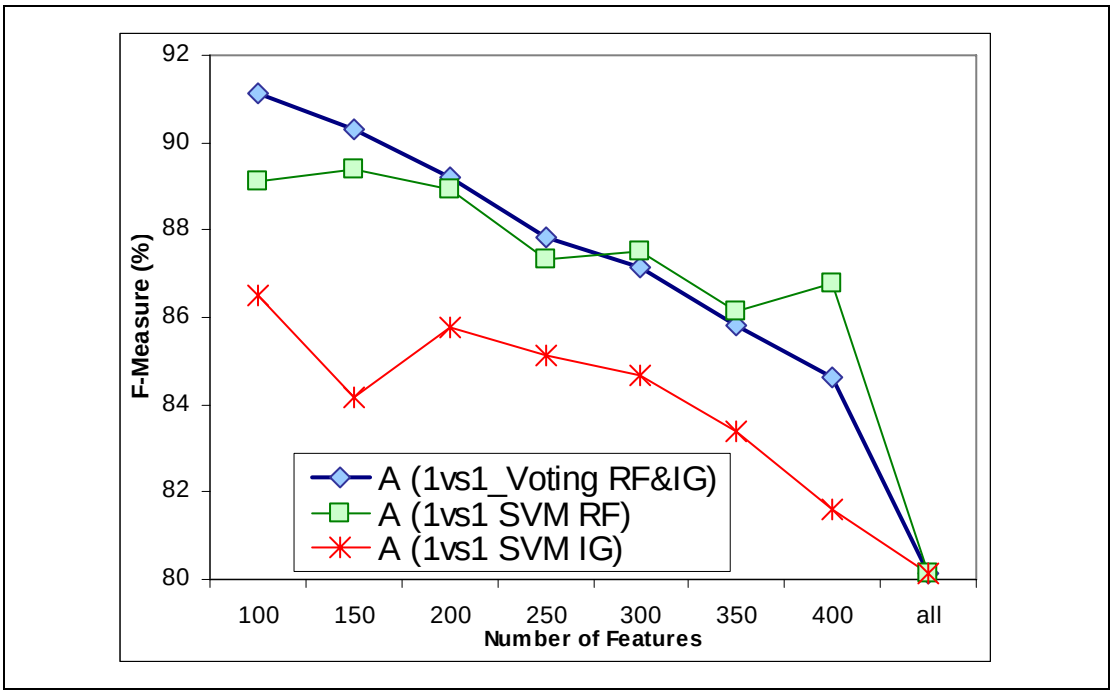
3.1) อัลกอริทึมการให้คะแนนเสียงระหว่าง RF และ IG

สำหรับประสิทธิภาพของอัลกอริทึม 1vs1_Voting พบว่า 1vs1_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vs1 SVM classification) คือ

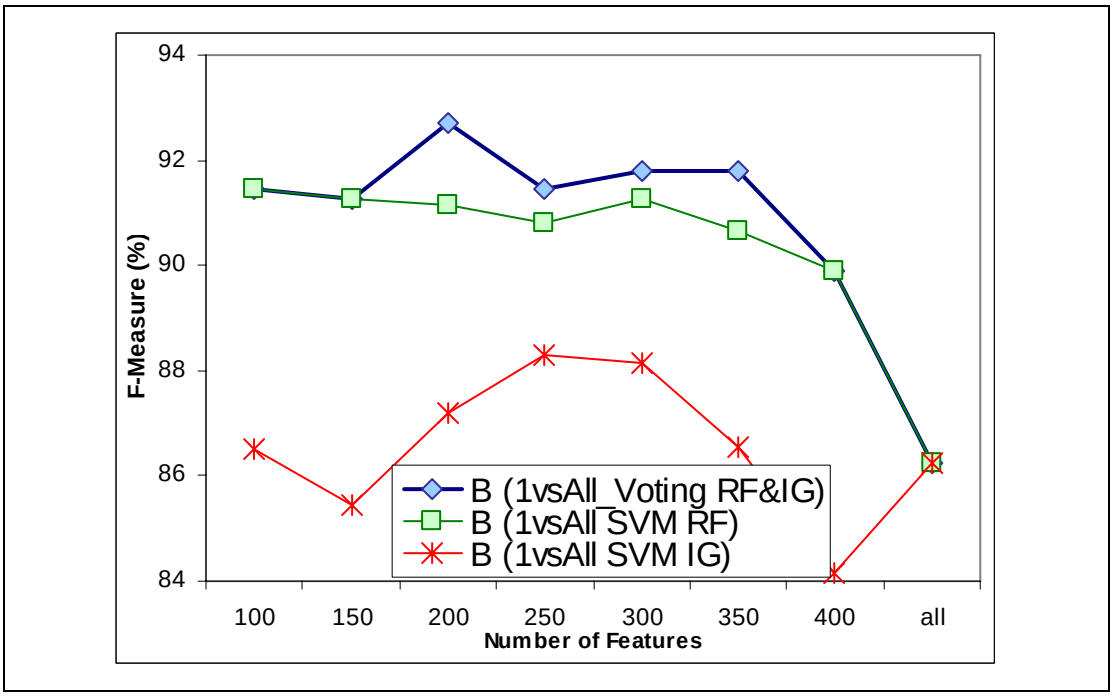
- สำหรับผลการทดลอง A ตารางที่ 5.5 แสดงดังภาพประกอบ 5.12 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vs1_Voting ระหว่าง RF และ IG มีค่า F-measure ที่สูงกว่าคือ 87.81% ขณะที่ 1vs1 SVM ของ RF มีค่า F-measure คือ 87.31% และ IG มีค่า F-measure คือ 85.13%
- สำหรับผลการทดลอง C ตารางที่ 5.7 แสดงดังภาพประกอบ 5.14 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vs1_Voting ระหว่าง RF และ IG ให้ค่า F-measure ที่สูงกว่าคือ 90.36% ขณะที่ 1vs1 SVM ของ RF มีค่า F-measure คือ 89.06% และ IG มีค่า F-measure คือ 86.24%

สำหรับประสิทธิภาพของอัลกอริทึม 1vsAll_Voting พบว่า 1vsAll_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vsAll SVM classification) คือ

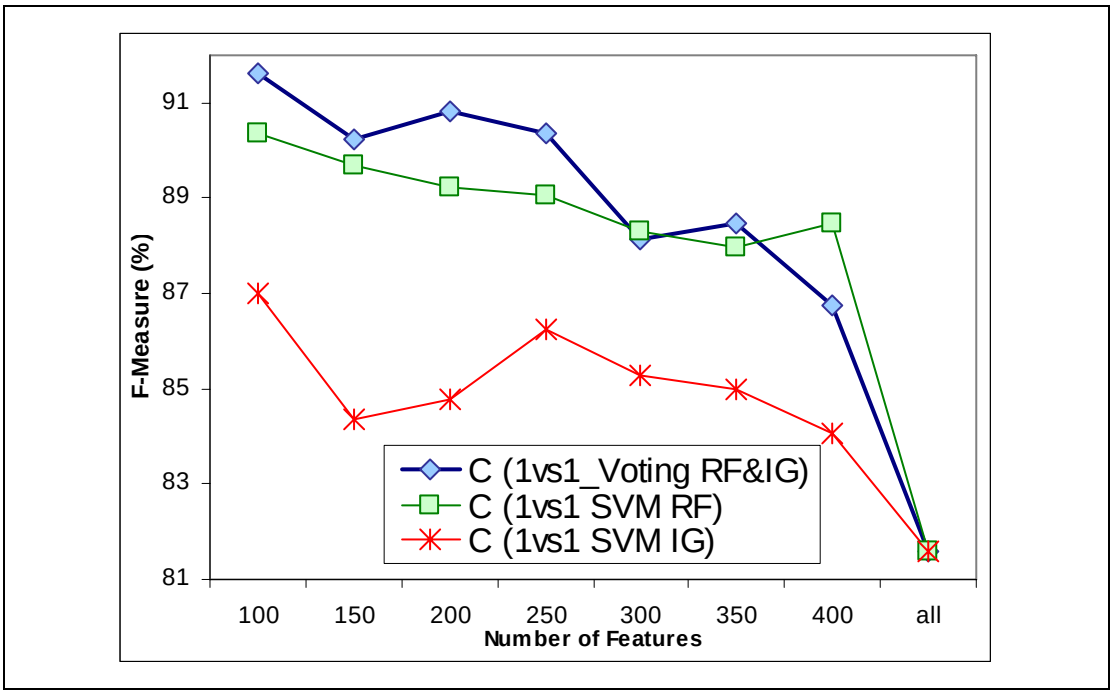
- สำหรับผลการทดลอง B ตารางที่ 5.6 แสดงดังภาพประกอบ 5.13 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง RF และ IG มีค่า F-measure ที่สูงกว่าคือ 91.46% ขณะที่ 1vsAll SVM ของ RF มีค่า F-measure คือ 90.82% และ IG มีค่า F-measure คือ 88.29%
- สำหรับผลการทดลอง D ตารางที่ 5.8 แสดงดังภาพประกอบ 5.15 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง RF และ IG มีค่า F-measure ที่สูงกว่าคือ 93.98% ขณะที่ 1vsAll SVM ของ RF มีค่า F-measure คือ 91.30% และ IG มีค่า F-measure คือ 89.87%



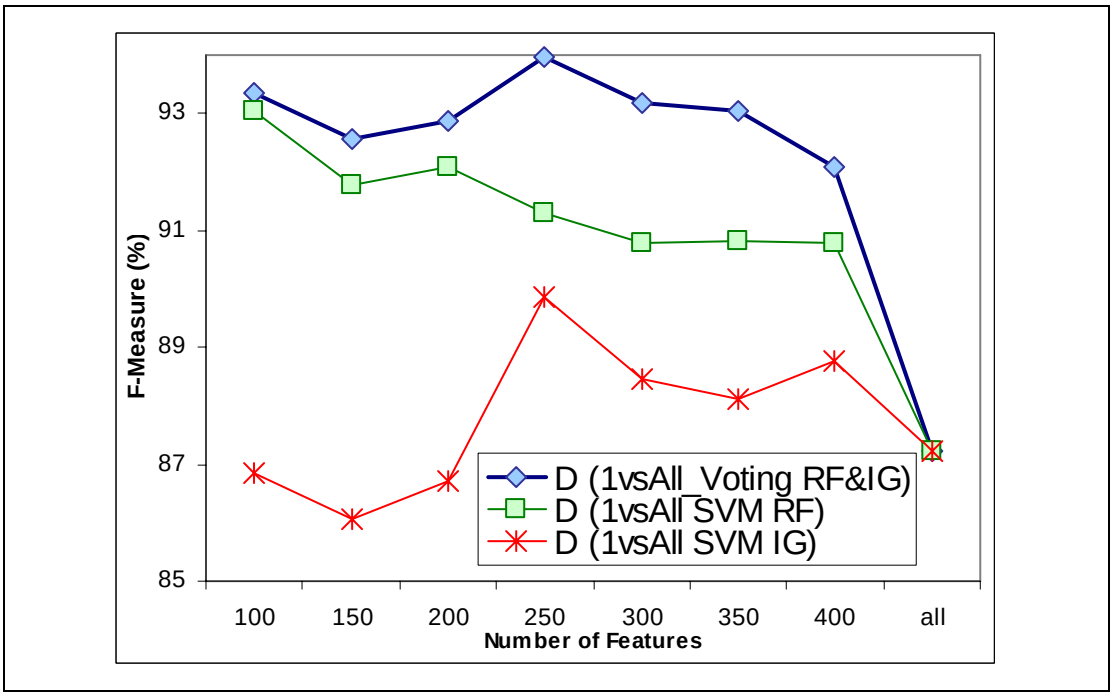
ภาพประกอบ 5.12 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ IG การทดลอง A



ภาพประกอบ 5.13 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ IG การทดลอง B



ภาพประกอบ 5.14 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ IG การทดลอง C



ภาพประกอบ 5.15 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ IG การทดลอง D

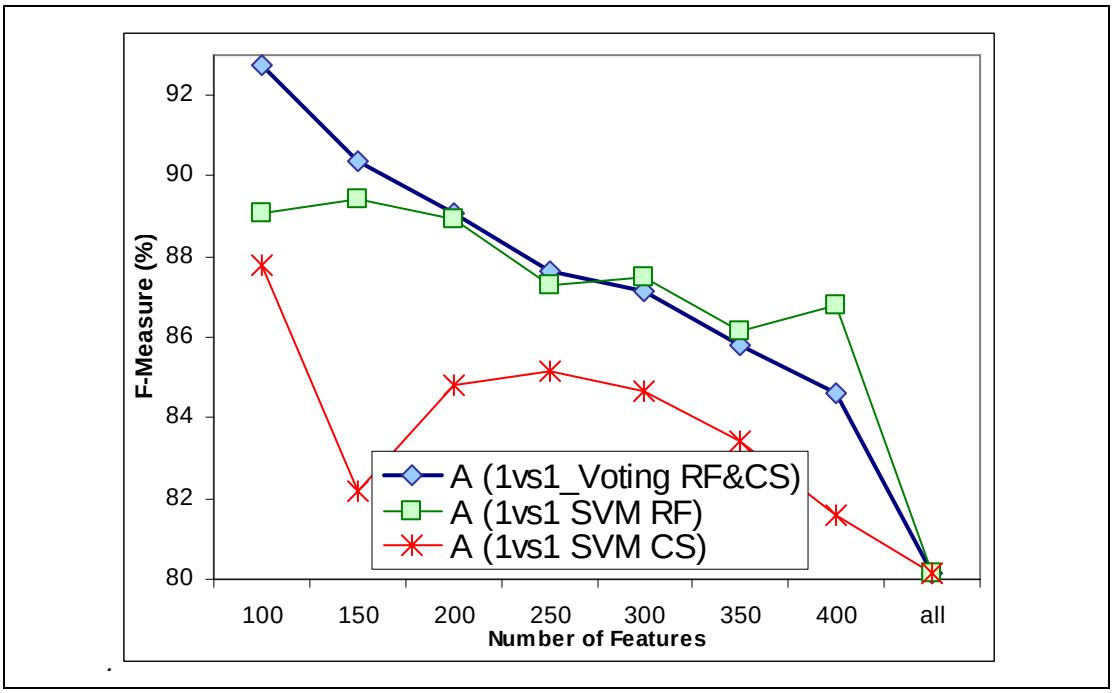
3.2) อัลกอริทึมการให้คะแนนเสียงระหว่าง RF และ CS

สำหรับประสิทธิภาพของอัลกอริทึม 1vs1_Voting พบว่า 1vs1_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vs1 SVM classification) คือ

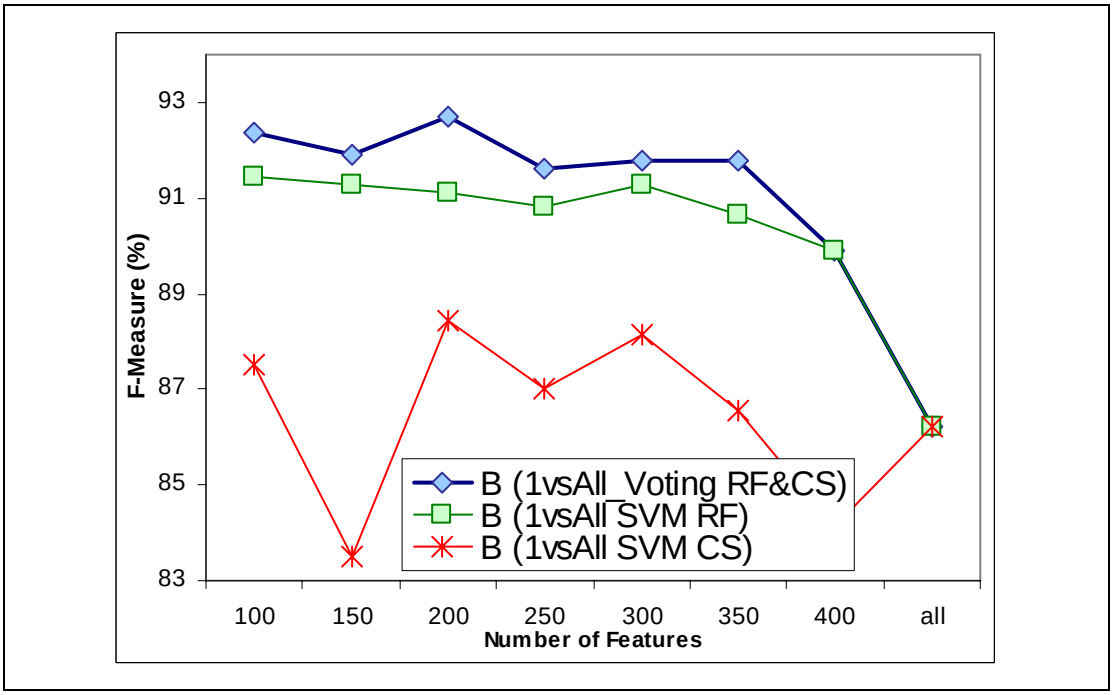
- สำหรับผลการทดลอง A ตารางที่ 5.5 แสดงดังภาพประกอบ 5.16 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vs1_Voting ระหว่าง RF และ CS มีค่า F-measure ที่สูงกว่าคือ 87.63% ขณะที่ 1vs1 SVM ของ RF มีค่า F-measure คือ 87.31% และ CS มีค่า F-measure คือ 85.14%
- สำหรับผลการทดลอง C ตารางที่ 5.7 แสดงดังภาพประกอบ 5.18 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vs1_Voting ระหว่าง RF และ CS มีค่า F-measure ที่สูงกว่าคือ 90.23% ขณะที่ 1vs1 SVM ของ RF มีค่า F-measure คือ 89.06% และ CS มีค่า F-measure คือ 85.45%

สำหรับประสิทธิภาพของอัลกอริทึม 1vsAll_Voting พบว่า 1vsAll_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vsAll SVM classification) คือ

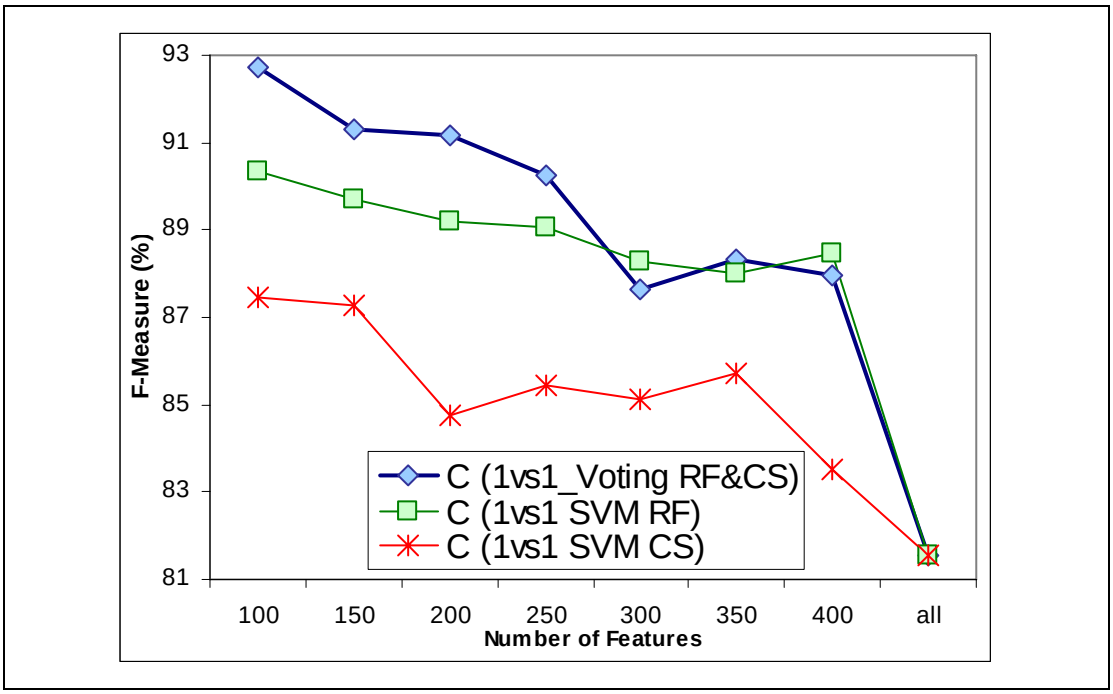
- สำหรับผลการทดลอง B ตารางที่ 5.6 แสดงดังภาพประกอบ 5.17 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง RF และ CS ให้ค่า F-measure ที่สูงกว่าคือ 91.63% ขณะที่ 1vsAll SVM ของ RF มีค่า F-measure คือ 90.82% และ CS มีค่า F-measure คือ 87.02%
- สำหรับผลการทดลอง D ตารางที่ 5.8 แสดงดังภาพประกอบ 5.19 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง RF และ CS ให้ค่า F-measure ที่สูงกว่าคือ 93.18% ขณะที่ 1vsAll SVM ของ RF มีค่า F-measure คือ 91.30% และ CS มีค่า F-measure คือ 88.43%



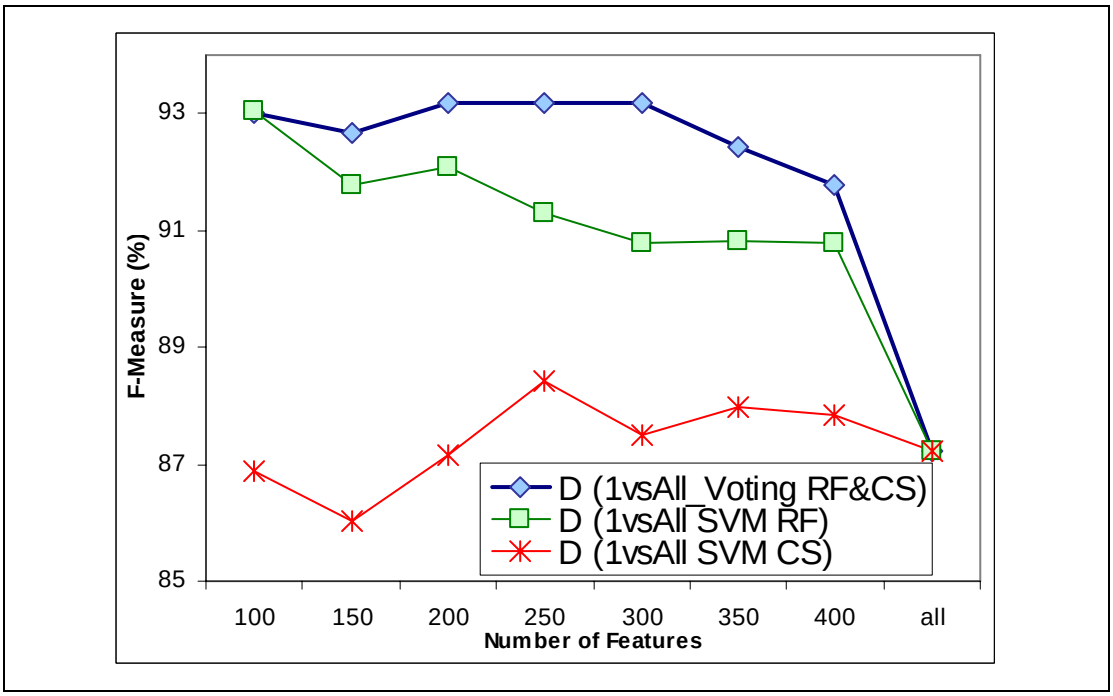
ภาพประกอบ 5.16 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ CS การทดลอง A



ภาพประกอบ 5.17 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ CS การทดลอง B



ภาพประกอบ 5.18 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ CS การทดลอง C



ภาพประกอบ 5.19 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ CS การทดลอง D

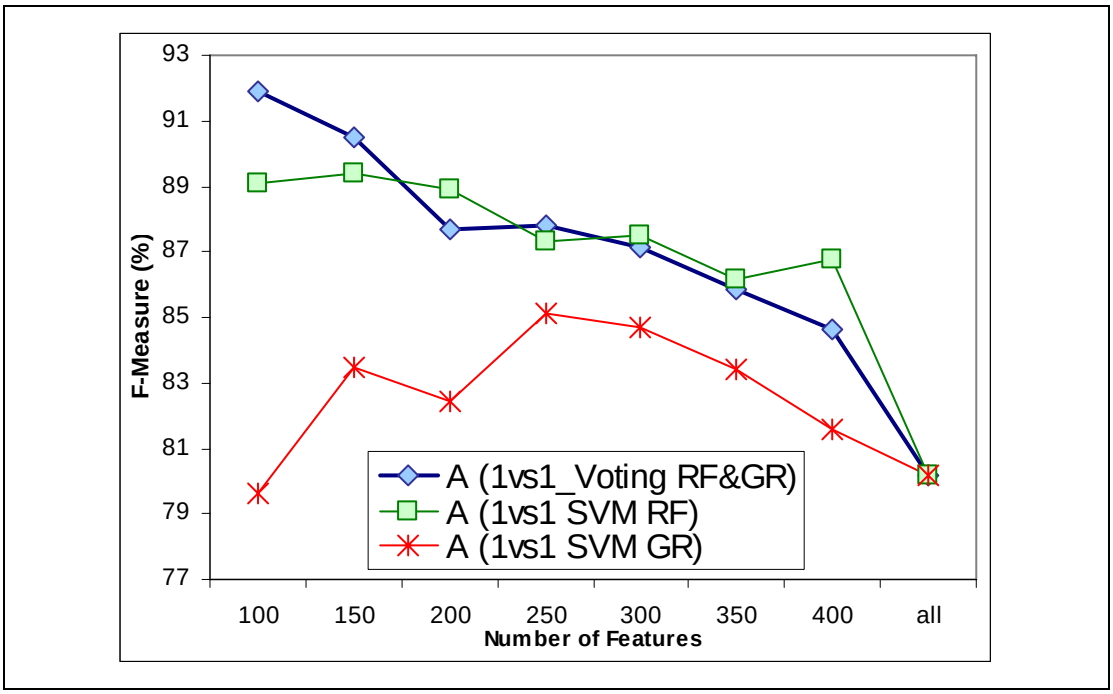
3.3) อัลกอริทึมการให้คะแนนเสียงระหว่าง RF และ GR

สำหรับประสิทธิภาพของอัลกอริทึม 1vs1_Voting พบว่า 1vs1_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vs1 SVM classification) คือ

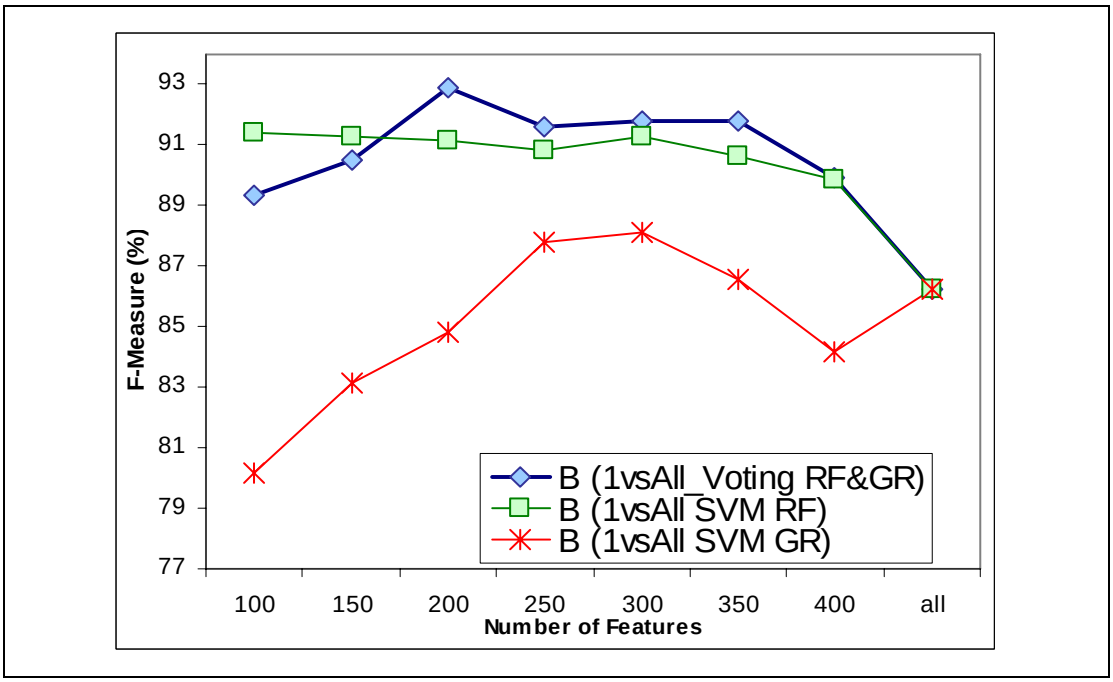
- สำหรับผลการทดลอง A ตารางที่ 5.5 แสดงดังภาพประกอบ 5.20 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vs1_Voting ระหว่าง RF และ GR มีค่า F-measure ที่สูงกว่าคือ 87.81% ขณะที่ 1vs1 SVM ของ RF มีค่า F-measure คือ 87.31% และ GR มีค่า F-measure คือ 85.13%
- สำหรับผลการทดลอง C ตารางที่ 5.7 แสดงดังภาพประกอบ 5.22 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vs1_Voting ระหว่าง RF และ GR มีค่า F-measure ที่สูงกว่าคือ 90.05% ขณะที่ 1vs1 SVM ของ RF มีค่า F-measure คือ 89.06% และ GR มีค่า F-measure คือ 83.58%

สำหรับประสิทธิภาพของอัลกอริทึม 1vsAll_Voting พบว่า 1vsAll_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vsAll SVM classification) คือ

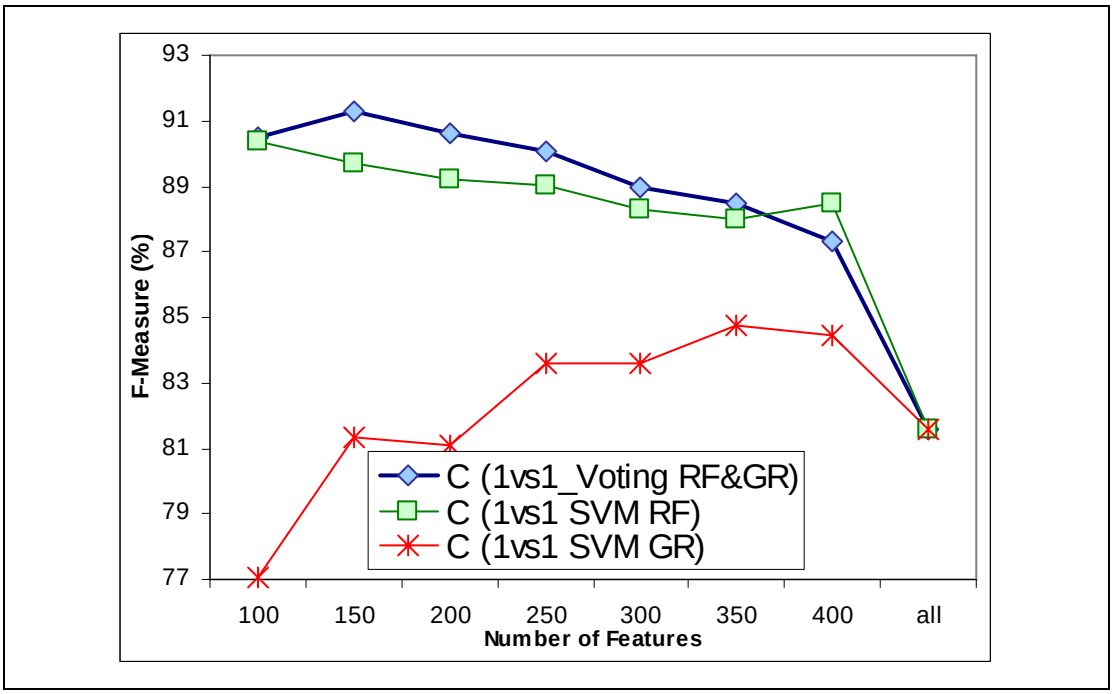
- สำหรับผลการทดลอง B ตารางที่ 5.6 แสดงดังภาพประกอบ 5.21 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง RF และ GR ให้ค่า F-measure ที่สูงกว่าคือ 91.61% ขณะที่ 1vsAll SVM ของ RF มีค่า F-measure คือ 90.82% และ GR มีค่า F-measure คือ 87.81%
- สำหรับผลการทดลอง D ตารางที่ 5.8 แสดงดังภาพประกอบ 5.23 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง RF และ GR ให้ค่า F-measure ที่สูงกว่าคือ 92.23% ขณะที่ 1vsAll SVM ของ RF มีค่า F-measure คือ 91.30% และ GR มีค่า F-measure คือ 85.91%



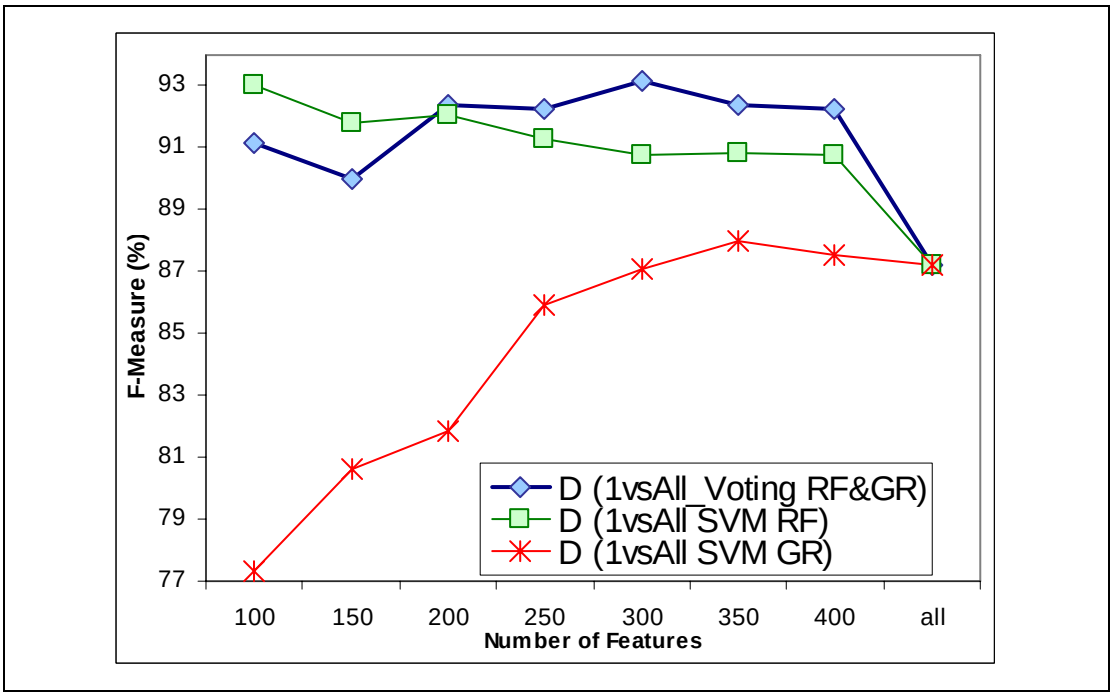
ภาพประกอบ 5.20 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ GR การทดลอง A



ภาพประกอบ 5.21 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ GR การทดลอง B



ภาพประกอบ 5.22 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ GR การทดลอง C



ภาพประกอบ 5.23 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง RF และ GR การทดลอง D

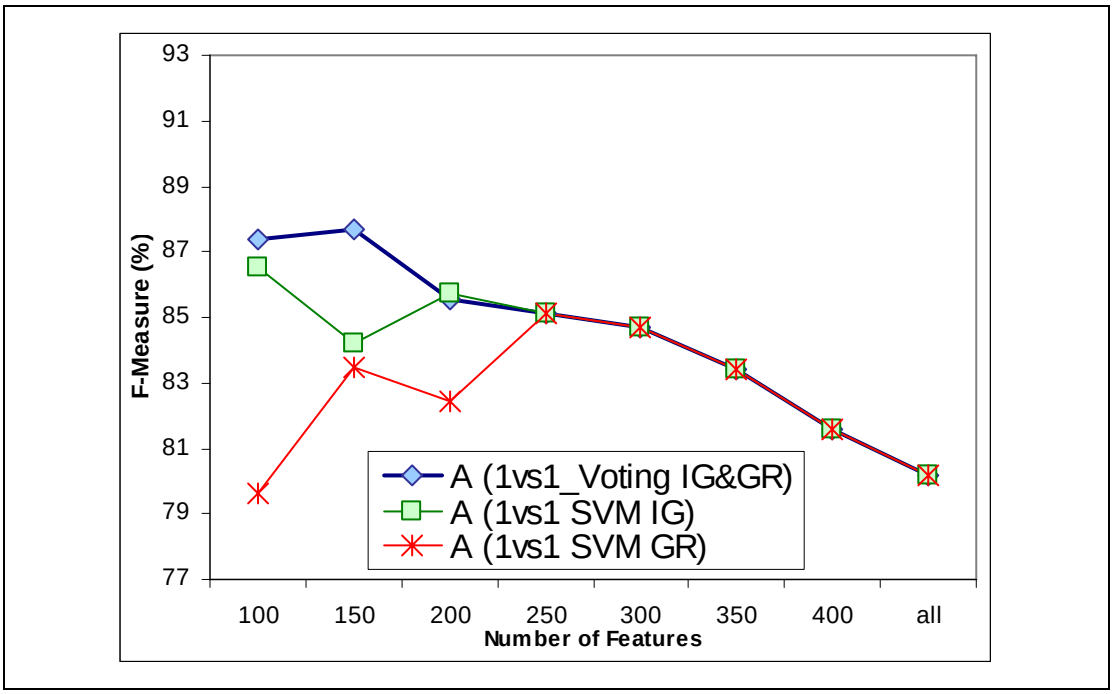
3.4) อัลกอริทึมการให้คะแนนเสียงระหว่าง IG และ GR

สำหรับประสิทธิภาพของอัลกอริทึม 1vs1_Voting พบว่า 1vs1_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vs1 SVM classification) คือ

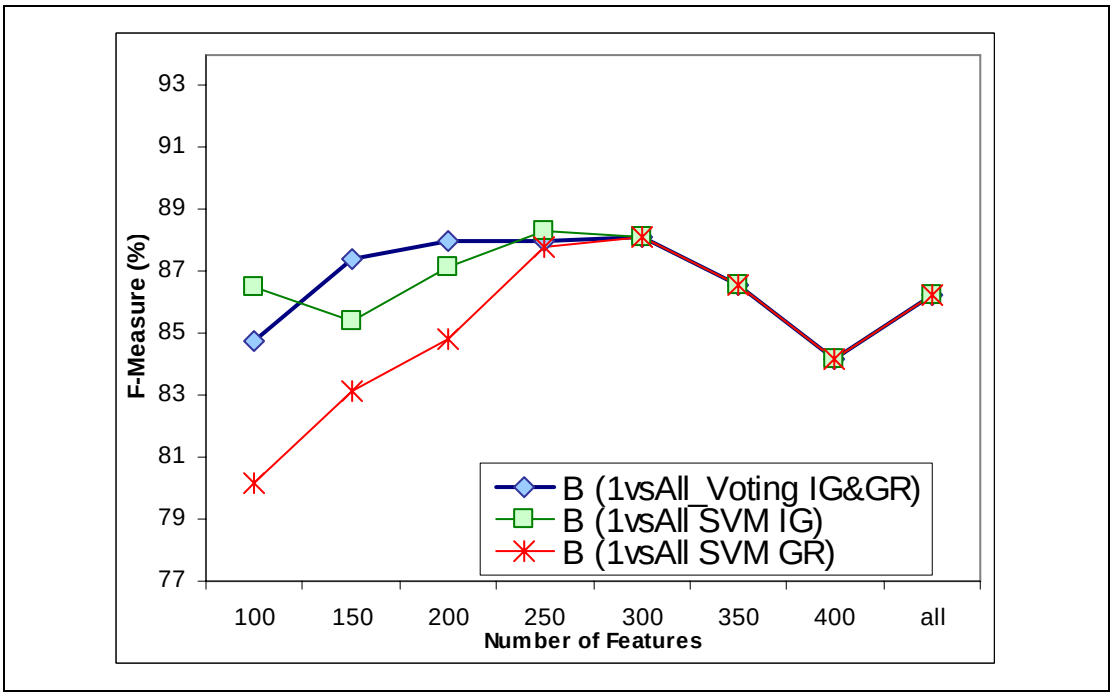
- สำหรับผลการทดลอง A ตารางที่ 5.5 แสดงดังภาพประกอบ 5.24 ที่จำนวนลักษณะเฉพาะ 150 พบว่า 1vs1_Voting ระหว่าง IG และ GR มีค่า F-measure ที่สูงกว่าคือ 87.69% ขณะที่ 1vs1 SVM ของ IG มีค่า F-measure คือ 84.19% และ GR มีค่า F-measure คือ 83.50%
- สำหรับผลการทดลอง C ตารางที่ 5.7 แสดงดังภาพประกอบ 5.26 ที่จำนวนลักษณะเฉพาะ 150 พบว่า 1vs1_Voting ระหว่าง IG และ GR มีค่า F-measure ที่สูงกว่าคือ 89.22% ขณะที่ 1vs1 SVM ของ IG มีค่า F-measure คือ 84.35% และ GR มีค่า F-measure คือ 81.32%

สำหรับประสิทธิภาพของอัลกอริทึม 1vsAll_Voting พบว่า 1vsAll_Voting ให้ค่า F-measure สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vsAll SVM classification) คือ

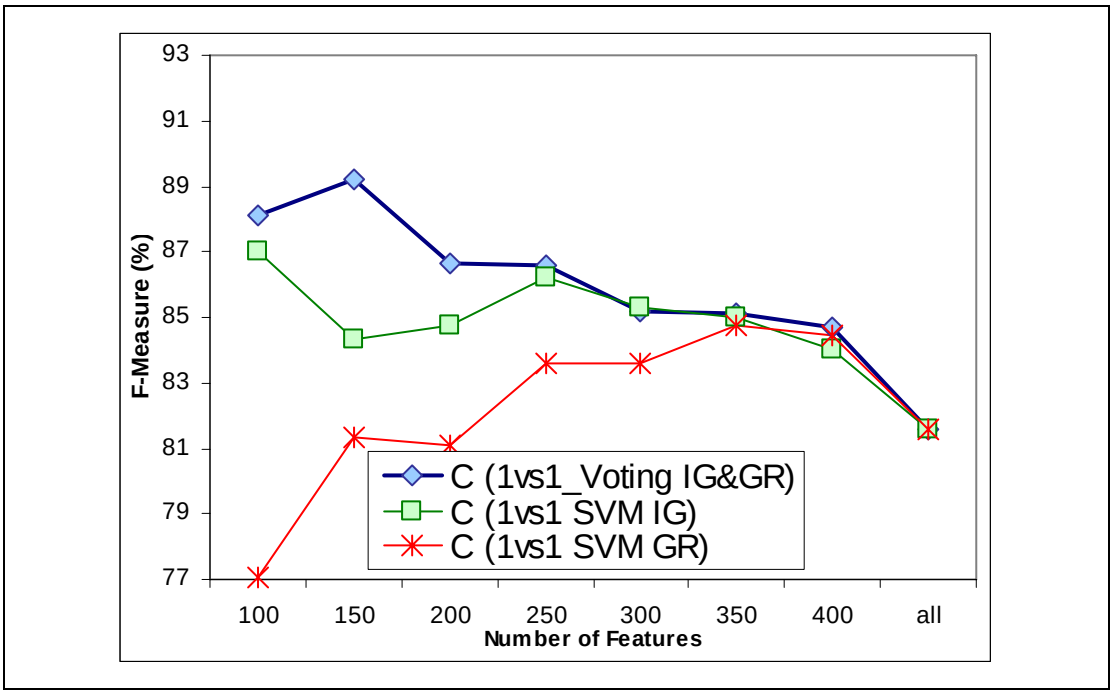
- สำหรับผลการทดลอง B ตารางที่ 5.6 แสดงดังภาพประกอบ 5.25 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง IG และ GR มีค่า F-measure ที่สูงกว่าคือ 87.41% ขณะที่ 1vsAll SVM ของ IG มีค่า F-measure คือ 85.43% และ GR มีค่า F-measure คือ 83.13%
- สำหรับผลการทดลอง D ตารางที่ 5.8 แสดงดังภาพประกอบ 5.27 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง IG และ GR มีค่า F-measure ที่สูงกว่าคือ 87.79% ขณะที่ 1vsAll SVM ของ IG มีค่า F-measure คือ 86.06% และ GR มีค่า F-measure คือ 80.61%



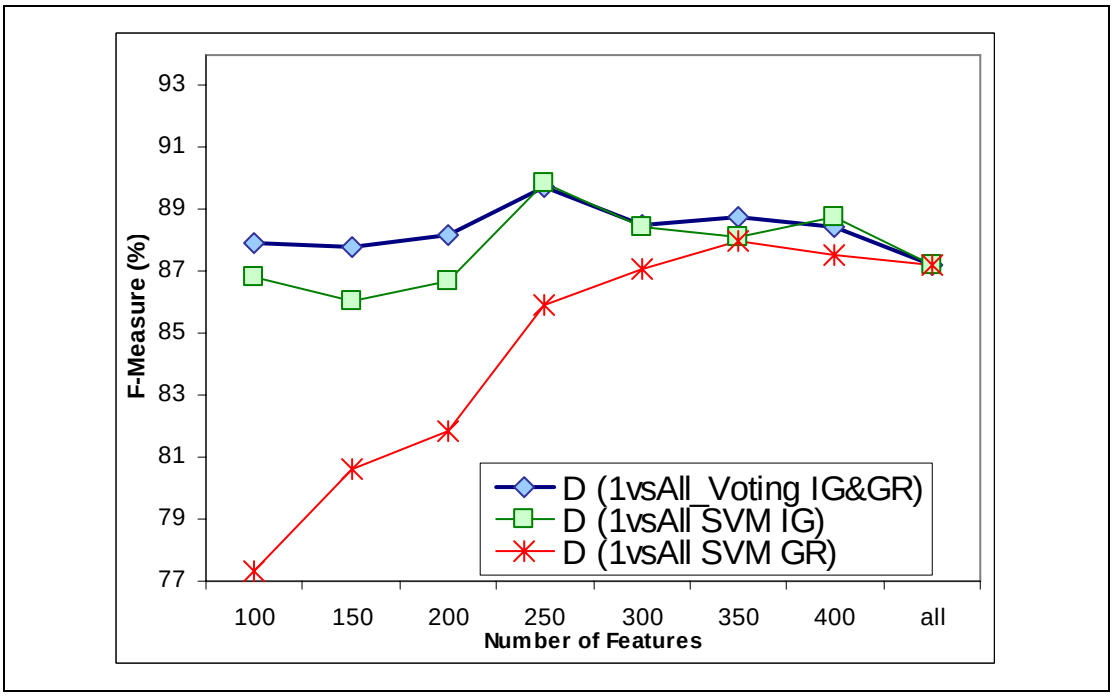
ภาพประกอบ 5.24 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง IG และ GR การทดลอง A



ภาพประกอบ 5.25 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง IG และ GR การทดลอง B



ภาพประกอบ 5.26 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง IG และ GR การทดลอง C



ภาพประกอบ 5.27 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง IG และ GR การทดลอง D

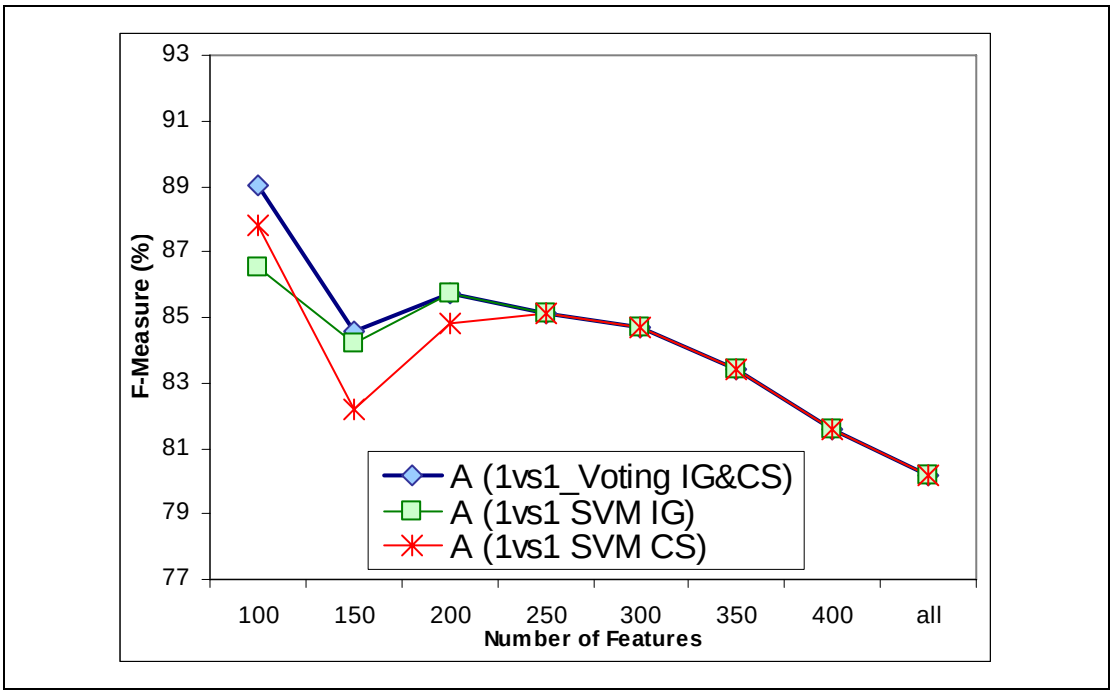
3.5) อัลกอริทึมการให้คะแนนเสียงระหว่าง IG และ CS

สำหรับประสิทธิภาพของอัลกอริทึม 1vs1_Voting พบว่า 1vs1_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vs1 SVM classification) คือ

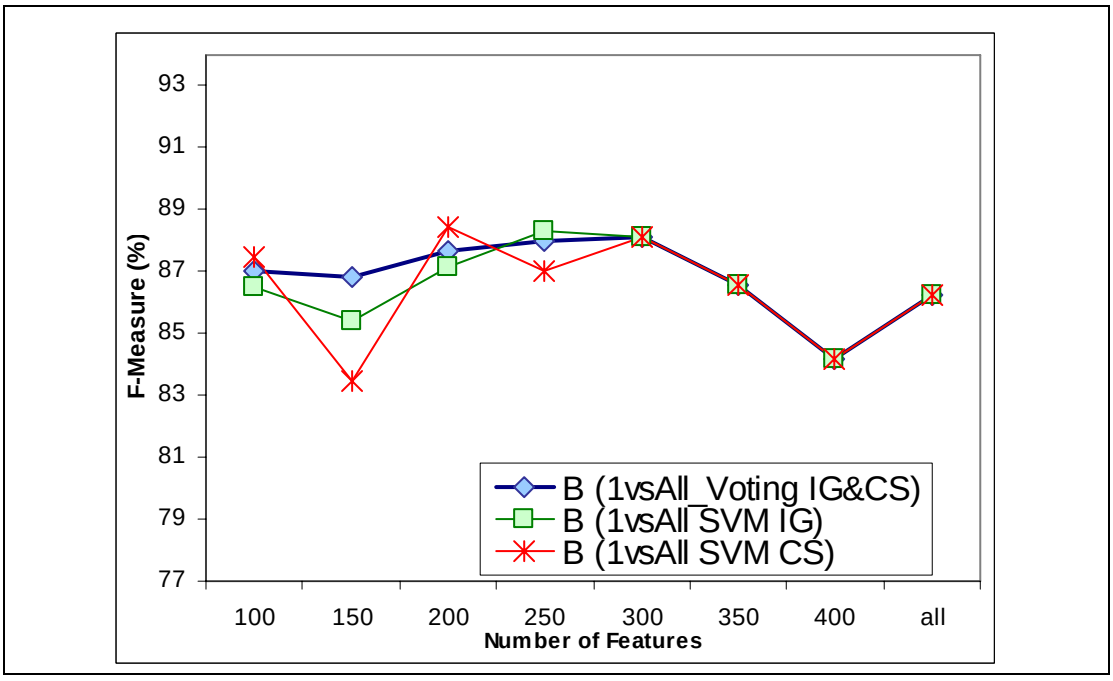
- สำหรับผลการทดลอง A ตารางที่ 5.5 แสดงดังภาพประกอบ 5.28 ที่จำนวนลักษณะเฉพาะ 150 พบว่า 1vs1_Voting ระหว่าง IG และ CS มีค่า F-measure ที่สูงกว่าคือ 84.57% ขณะที่ 1vs1 SVM ของ IG มีค่า F-measure คือ 84.19% และ CS มีค่า F-measure คือ 82.19%
- สำหรับผลการทดลอง C ตารางที่ 5.7 แสดงดังภาพประกอบ 5.30 ที่จำนวนลักษณะเฉพาะ 150 พบว่า 1vs1_Voting ระหว่าง IG และ CS มีค่า F-measure ที่สูงกว่าคือ 86.65% ขณะที่ 1vs1 SVM ของ IG มีค่า F-measure คือ 84.35% และ CS มีค่า F-measure คือ 86.27%

สำหรับประสิทธิภาพของอัลกอริทึม 1vsAll_Voting พบว่า 1vsAll_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vsAll SVM classification) คือ

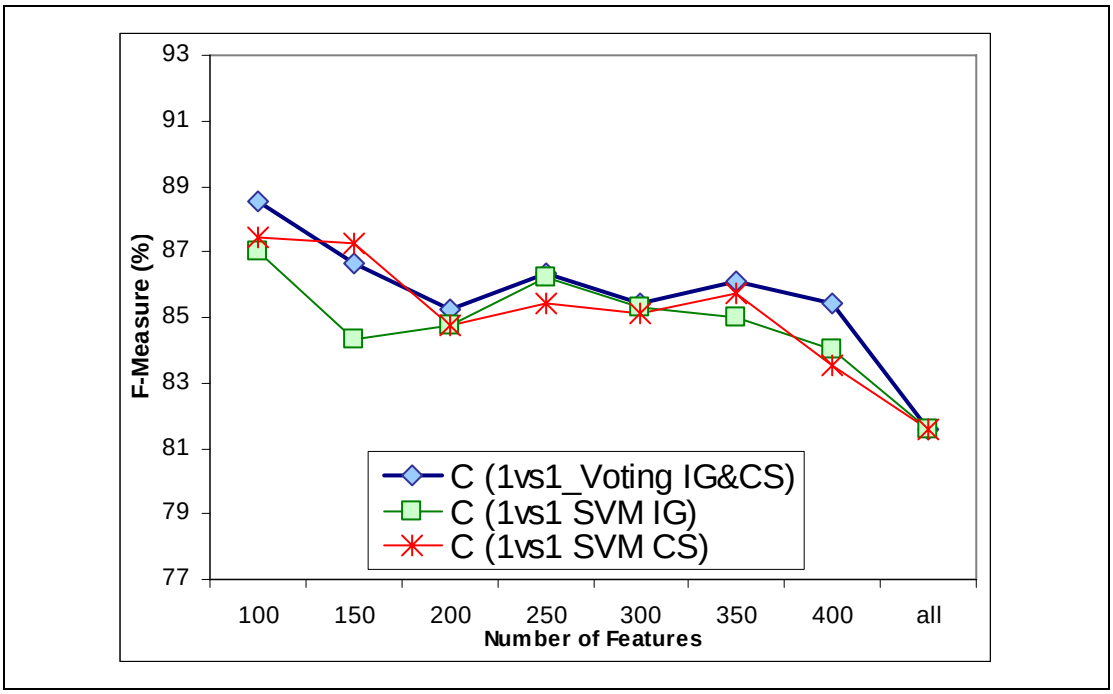
- สำหรับผลการทดลอง B ตารางที่ 5.6 แสดงดังภาพประกอบ 5.29 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง IG และ CS ให้ค่า F-measure ที่สูงกว่าคือ 86.82% ขณะที่ 1vsAll SVM ของ IG มีค่า F-measure คือ 85.43% และ CS มีค่า F-measure คือ 83.49%
- สำหรับผลการทดลอง D ตารางที่ 5.8 แสดงดังภาพประกอบ 5.31 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง IG และ CS ให้ค่า F-measure ที่สูงกว่าคือ 88.12% ขณะที่ 1vsAll SVM ของ IG มีค่า F-measure คือ 86.06% และ CS มีค่า F-measure คือ 86.04%



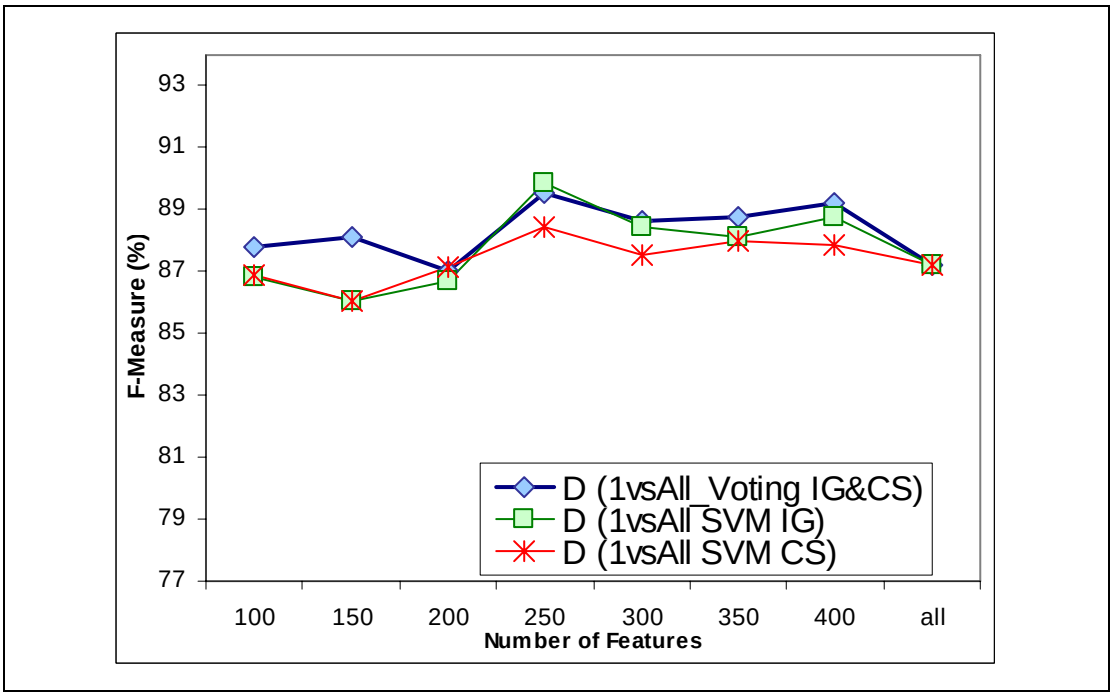
ภาพประกอบ 5.28 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง IG และ CS การทดลอง A



ภาพประกอบ 5.29 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง IG และ CS การทดลอง B



ภาพประกอบ 5.30 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง IG และ CS การทดลอง C



ภาพประกอบ 5.31 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง IG และ CS การทดลอง D

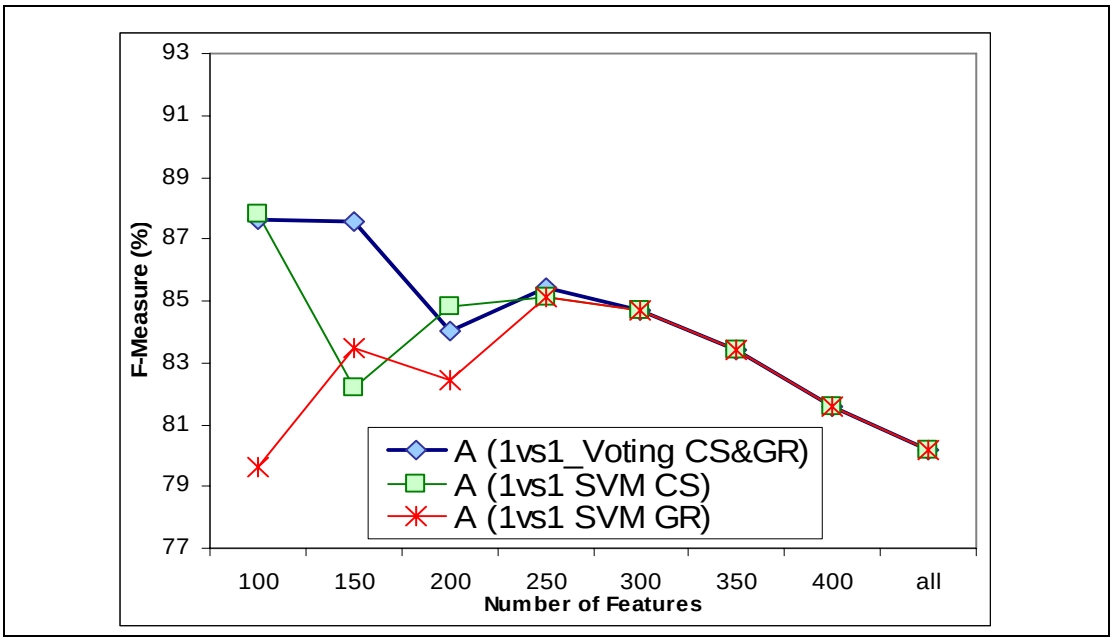
3.6) อัลกอริทึมการให้คะแนนเสียงระหว่าง CS และ GR

สำหรับประสิทธิภาพของอัลกอริทึม 1vs1_Voting พบว่า 1vs1_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vs1 SVM classification) คือ

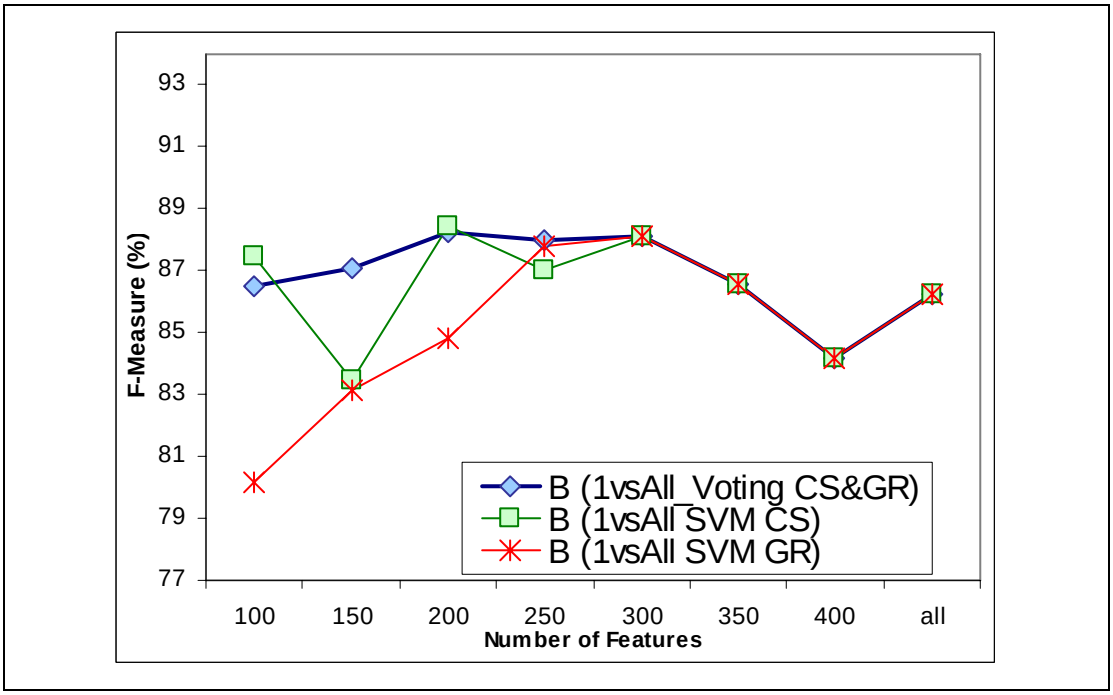
- สำหรับผลการทดลอง A ตารางที่ 5.5 แสดงดังภาพประกอบ 5.32 ที่จำนวนลักษณะเฉพาะ 150 พบว่า 1vs1_Voting ระหว่าง CS และ GR มีค่า F-measure ที่สูงกว่าคือ 87.56% ขณะที่ 1vs1 SVM ของ CS มีค่า F-measure คือ 82.19% และ GR มีค่า F-measure คือ 83.50%
- สำหรับผลการทดลอง C ตารางที่ 5.7 แสดงดังภาพประกอบ 5.34 ที่จำนวนลักษณะเฉพาะ 150 พบว่า 1vs1_Voting ระหว่าง CS และ GR มีค่า F-measure ที่สูงกว่าคือ 89.10% ขณะที่ 1vs1 SVM ของ CS มีค่า F-measure คือ 87.27% และ GR มีค่า F-measure คือ 81.32%

สำหรับประสิทธิภาพของอัลกอริทึม 1vsAll_Voting พบว่า 1vsAll_Voting ให้ค่า F-measure ที่สูงกว่าใช้การจำแนกประเภทด้วย SVM เพียงวิธีเดียว (1vsAll SVM classification) คือ

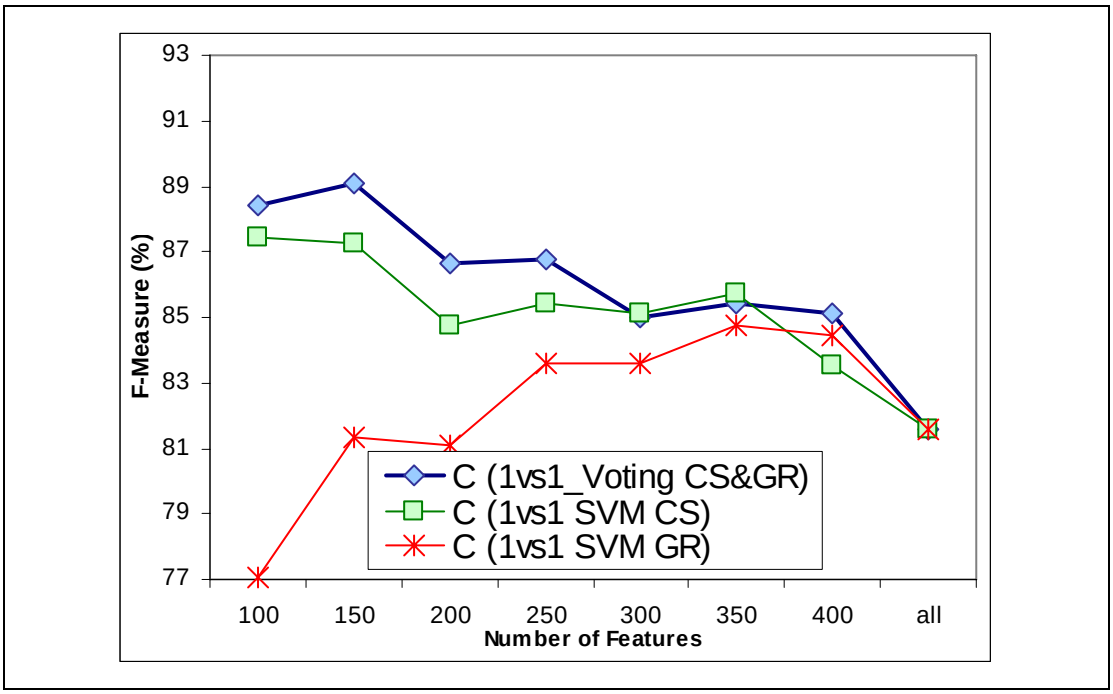
- สำหรับผลการทดลอง B ตารางที่ 5.6 แสดงดังภาพประกอบ 5.33 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง CS และ GR มีค่า F-measure ที่สูงกว่าคือ 87.07% ขณะที่ 1vsAll SVM ของ CS มีค่า F-measure คือ 83.49% และ GR มีค่า F-measure คือ 83.13%
- สำหรับผลการทดลอง D ตารางที่ 5.8 แสดงดังภาพประกอบ 5.35 ที่จำนวนลักษณะเฉพาะ 250 พบว่า 1vsAll_Voting ระหว่าง CS และ GR มีค่า F-measure ที่สูงกว่าคือ 87.46% ขณะที่ 1vsAll SVM ของ CS มีค่า F-measure คือ 86.04% และ GR มีค่า F-measure คือ 80.61%



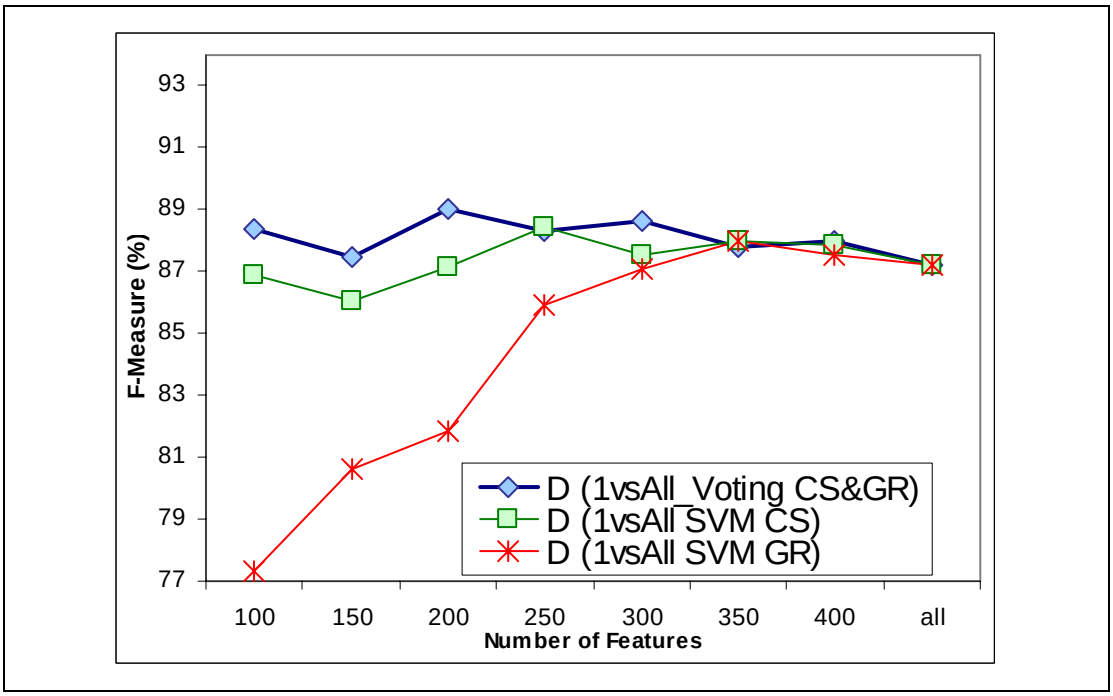
ภาพประกอบ 5.32 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง CS และ GR การทดลอง A



ภาพประกอบ 5.33 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง CS และ GR การทดลอง B



ภาพประกอบ 5.34 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง CS และ GR การทดลอง C



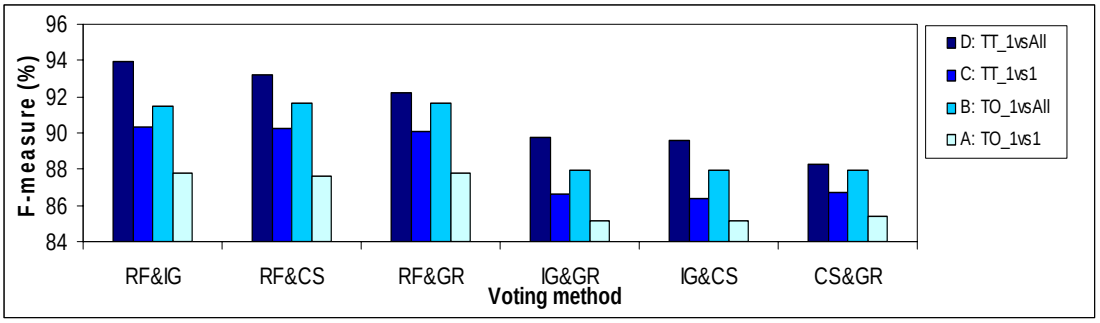
ภาพประกอบ 5.35 ค่า F-measure อัลกอริทึมการให้คะแนนเสียง CS และ GR การทดลอง D

4) ประเด็นการเปรียบเทียบผลการทดลองของ WPC_VAMSVM

การทดลอง D ซึ่งใช้ลักษณะเฉพาะจากข้อความและหัวเรื่องกับอัลกอริทึม 1vsAll_Voting ให้ค่า F-measure สูงที่สุด เมื่อเทียบกับการทดลองอื่น ๆ ซึ่งได้แก่ การทดลอง A ใช้ลักษณะเฉพาะจากข้อความเท่านั้นกับอัลกอริทึม 1vs1_Voting การทดลอง B ใช้ลักษณะเฉพาะจากข้อความเท่านั้นกับอัลกอริทึม 1vsAll_Voting การทดลอง C ใช้ลักษณะเฉพาะจากข้อความและหัวเรื่องกับอัลกอริทึม 1vs1_Voting ดังภาพประกอบ 5.36 ซึ่งมีรายละเอียดดังนี้

- ผลการทดลองสำหรับอัลกอริทึมการให้คะแนนเสียงระหว่าง RF และ IG พบว่าค่า F-measure ของการทดลอง D ตารางที่ 5.8 ที่ลักษณะเฉพาะจำนวน 250 คือ 93.98% ขณะที่การทดลอง A ตารางที่ 5.5 ที่ลักษณะเฉพาะจำนวน 250 มีค่า F-measure คือ 87.81% การทดลอง B ตารางที่ 5.6 ที่ลักษณะเฉพาะจำนวน 250 มีค่า F-measure คือ 91.46% และการทดลอง C ตารางที่ 5.7 ที่ลักษณะเฉพาะจำนวน 250 มีค่า F-measure คือ 90.36%
- ผลการทดลองสำหรับอัลกอริทึมการให้คะแนนเสียงระหว่าง RF และ CS พบว่าค่า F-measure ของการทดลอง D ตารางที่ 5.8 ที่ลักษณะเฉพาะจำนวน 250 คือ 93.18% ขณะที่การทดลอง A ตารางที่ 5.5 ที่ลักษณะเฉพาะจำนวน 250 มีค่า F-measure คือ 87.63% การทดลอง B ตารางที่ 5.6 ที่ลักษณะเฉพาะจำนวน 250 มีค่า F-measure คือ 91.63% และการทดลอง C ตารางที่ 5.7 ที่ลักษณะเฉพาะจำนวน 250 มีค่า F-measure คือ 90.23%
- ผลการทดลองสำหรับอัลกอริทึมการให้คะแนนเสียงระหว่าง RF และ GR พบว่าค่า F-measure ของการทดลอง D ตารางที่ 5.8 ที่ลักษณะเฉพาะจำนวน 250 คือ 92.23% ขณะที่ การทดลอง A ตารางที่ 5.5 ที่ลักษณะเฉพาะจำนวน 250 มีค่า F-measure คือ 87.81% การทดลอง B ตารางที่ 5.6 ที่ลักษณะเฉพาะจำนวน 250 มีค่า F-measure คือ 91.61% และการทดลอง C ตารางที่ 5.7 ที่ลักษณะเฉพาะจำนวน 250 มีค่า F-measure คือ 90.05%
- ผลการทดลองสำหรับอัลกอริทึมการให้คะแนนเสียงระหว่าง IG และ GR พบว่าค่า F-measure ของการทดลอง D คือ 89.72% ขณะที่ การทดลอง A มีค่า F-measure คือ 85.11% การทดลอง B มีค่า F-measure คือ 87.97% และการทดลอง C มีค่า F-measure คือ 86.60%
- ผลการทดลองสำหรับอัลกอริทึมการให้คะแนนเสียงระหว่าง IG และ CS พบว่าค่า F-measure ของการทดลอง D คือ 89.57% ขณะที่ การทดลอง A มีค่า F-measure คือ 85.14% การทดลอง B มีค่า F-measure คือ 87.98% และการทดลอง C มีค่า F-measure คือ 86.36%
- ผลการทดลองสำหรับอัลกอริทึมการให้คะแนนเสียงระหว่าง CS และ GR พบว่าค่า F-measure ของการทดลอง D มีค่า F-measure คือ 88.30% ขณะที่ การทดลอง

A มีค่า F-measure คือ 85.43% การทดลอง B มีค่า F-measure คือ 87.97% และการทดลอง C มีค่า F-measure คือ 86.75%



ภาพประกอบ 5.36 ค่า F-measure อัลกอริทึมการให้คะแนนเสียงที่ลักษณะเฉพาะจำนวน 250

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอแนวคิดใหม่ในการจำแนกประเภทเว็บเพจโดยการลดขนาดลักษณะและซัพพอร์ตเวกเตอร์แมชชีน โดยสร้างแบบจำลองการจำแนกประเภทเว็บเพจโดยการให้คะแนนเสียงและซัพพอร์ตเวกเตอร์แมชชีน (Web Page Classification using Voting Algorithm and Multi-Class Support Vector Machine: WPC_VAMSVM) สำหรับการจำแนกประเภทเว็บเพจได้ใช้ซัพพอร์ตเวกเตอร์แมชชีนแบบมัลติคลาส 2 วิธีคือ 1vs1 และ 1vsAll จากนั้นจึงนำผลลัพธ์ที่ได้จากการจำแนกดังกล่าวเข้าสู่อัลกอริมการให้คะแนนเสียงเพื่อระบุคลาสของเว็บเพจนั้น

ผู้วิจัยได้พัฒนาโปรแกรมจากแบบจำลองที่นำเสนอเพื่อจำแนกประเภทเว็บเพจดังกล่าว ซึ่งผู้ใช้สามารถใช้งานได้ง่าย Graphic User Interface (GUI) โปรแกรมการจำแนกประเภทเว็บเพจนี้ได้พัฒนาโดยใช้ Visual C#.Net สำหรับเตรียมข้อมูลเว็บเพจ และสร้างลักษณะเฉพาะ ทำงานร่วมกับโปรแกรม WEKA แบบ Command Line Interface สำหรับลดขนาดลักษณะเฉพาะ และโปรแกรม MATLAB สำหรับจำแนกประเภทและให้คะแนนเสียงเว็บเพจ

ผลการทดลองของงานวิจัยนี้ในการลดขนาดลักษณะเฉพาะด้วยค่าความถี่เอกสาร เรื่อง “การจำแนกประเภทเว็บเพจโดยใช้ค่าความถี่เอกสารและซัพพอร์ตเวกเตอร์แมชชีน” ได้รับการตีพิมพ์ใน The 12th National Computer Science and Engineering Conference (NCSEC 2008) วันที่ 20-21 พฤศจิกายน 2551 หน้าที่ 498-504 ดังภาคผนวก ข และผลการทดลองเกี่ยวกับอัลกอริทึมการให้คะแนนเสียงเรื่อง “A Novel Voting Algorithm of Multi-Class SVM for Web Page Classification” ได้รับการตีพิมพ์ใน The 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009) วันที่ 8-11 สิงหาคม พ.ศ. 2552 ดังภาคผนวก ค

ผลลัพธ์ของการจำแนกประเภทเว็บเพจโดยการลดขนาดลักษณะเฉพาะและซัพพอร์ตเวกเตอร์แมชชีน สามารถสรุปได้ 5 ประเด็นดังต่อไปนี้

1) การจำแนกประเภทโดยใช้ค่าความถี่เอกสารร่วมกับตัวจำแนกประเภท SVM ให้ผลการทดลองที่ดีที่สุด เมื่อเทียบกับ ต้นไม้ตัดสินใจ C4.5 และ RBFNN โดยสามารถลดขนาดลักษณะเฉพาะจำนวนมาก ซึ่งยังได้ผลการทดลองค่า F-measure ที่สูง

2) การลดขนาดลักษณะเฉพาะโดยการใช้วิธี ReliefF Feature Reduction ให้ค่า F-measure สูงที่สุดเมื่อเทียบกับ Information Gain, Chi Square, and Gain Ratio

3) สำหรับการจำแนกประเภทโดยใช้ซัพพอร์ตเวกเตอร์แบบซีนแบบมัลติคลาส เปรียบเทียบ 2 ชนิดระหว่าง 1vs1 และ 1vsAll พบว่า 1vsAll ให้ค่า F-measure ที่สูงกว่า 1vs1 SVM ทั้งการใช้ลักษณะเฉพาะเพียงข้อความ และการใช้ลักษณะเฉพาะของข้อความและหัวเรื่อง

4) อัลกอริทึมการให้คะแนนเสียงทั้ง 2 แบบ คือ 1vs1_Voting และ 1vsAll_Voting ให้ค่า F-measure ที่สูงกว่าการใช้ซัพพอร์ตเวกเตอร์แบบซีน โดย 1vs1_Voting ให้ค่า F-measure ที่สูงกว่าการใช้ซัพพอร์ตเวกเตอร์แบบซีนแบบ 1vs1 และ 1vsAll_Voting ให้ค่า F-measure ที่สูงกว่าการใช้ซัพพอร์ตเวกเตอร์แบบซีนแบบ 1vsAll

5) เมื่อทำการเปรียบเทียบการลองพบว่า การทดลองซึ่งใช้ลักษณะเฉพาะจากข้อความและหัวเรื่องกับอัลกอริทึม 1vsAll_Voting ให้ค่า F-measure สูงที่สุด เมื่อเทียบกับการทดลองอื่น ๆ ซึ่งได้แก่ การทดลองที่ใช้ลักษณะเฉพาะจากข้อความเท่านั้นกับอัลกอริทึม 1vs1_Voting การทดลองที่ใช้ลักษณะเฉพาะจากข้อความเท่านั้นกับอัลกอริทึม 1vsAll_Voting การทดลองที่ใช้ลักษณะเฉพาะจากข้อความและหัวเรื่องกับอัลกอริทึม 1vs1_Voting

6.2 ปัญหาและอุปสรรค

6.2.1 สำหรับการทำงานกับข้อมูลที่มีจำนวนของลักษณะเฉพาะมากด้วยโปรแกรม WEKA ต้องใช้หน่วยจำสูง 1 กิกะไบท์ จึงจะสามารถทำงานได้อย่างมีประสิทธิภาพ หากหน่วยความจำของเครื่องคอมพิวเตอร์มีน้อยเกินไปจะทำให้ทำงานล่าช้าหรือไม่สามารถทำงานได้

6.2.2 ข้อมูลเว็บเพจเป็นข้อมูลที่ไม่มีรูปแบบของลักษณะเฉพาะที่ชัดเจน การเลือกลักษณะเฉพาะสำหรับนำมาใช้ในข้อมูลเว็บเพจชุดหนึ่ง อาจไม่สามารถนำมาใช้กับข้อมูลเว็บเพจอีกชุดหนึ่งได้ เนื่องจากข้อมูลเว็บเพจชุดนั้นอาจไม่มีลักษณะเฉพาะดังกล่าว หรือมีแต่ไม่สมบูรณ์

6.2.3 ข้อมูลเว็บเพจที่นำมาทดสอบอยู่ในรูปแบบที่ไม่สามารถนำมาใช้งานเพื่อการจำแนกประเภทได้ทันทีต้องใช้เวลาและกระบวนการที่ซับซ้อนในการแปลงให้อยู่ในรูปแบบที่สามารถนำมาใช้งานได้ ซึ่งต่างจากบางข้อมูลอื่นในการทำดาต้าไมนิ่งที่มีข้อมูลและจำนวนลักษณะเฉพาะที่คงที่สามารถนำมาใช้งานได้ทันที

6.3 ข้อเสนอแนะ

6.3.1 แบบจำลองการจำแนกประเภทเว็บเพจสำหรับงานวิจัยนี้ไปปรับปรุงกับภาษาอื่น เช่น ภาษาไทย ภาษาจีน ภาษาญี่ปุ่น เป็นต้น ซึ่งอาจได้ผลลัพธ์ที่แตกต่างกันขึ้นอยู่กับลักษณะของภาษานั้น ๆ

6.3.2 เนื่องจากแบบจำลองการจำแนกประเภทเว็บเพจสำหรับงานวิจัยนี้ได้ใช้การลดขนาดลักษณะเฉพาะที่ต้องระบุจำนวน หากมีการพัฒนาโดยใช้การลดขนาดลักษณะเฉพาะโดยผู้ใช้ไม่ต้องระบุจำนวน แล้วผลการทดลองยังคงให้ประสิทธิภาพที่ดี จะเป็นประโยชน์ต่อการจำแนกประเภทเว็บเพจสำหรับงานวิจัยนี้อย่างมาก

บรรณานุกรม

วัลลภ อินทร์ฉ่ำ, 2548. ระบบการจัดหมวดหมู่เอกสารภาษาไทยอัตโนมัติโดยใช้ SVM ร่วมกับประมวลผลภาษา. วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.

Chen, J., and Huang R. 2004. Research of Web Classification Mining Based on RBF Neural Network. Control Automation Robotics and Vision Conference 2004 (ICARCV), pp. 1365- 1367.

Data Mining Software in Java, The University of Waikato. http://www.cs.waikato.ac.nz/ml/weka/.

Frakes, W. B., and Yates, R.B. 1992. Information retrieval data structure & algorithm New Jersey: Prentice Hall.

Ganchev, T., Zervas, P., Fakotakis, N., and Kokkinakis, G. 2006. Benchmarking feature selection techniques on the speaker verification task. Proceedings of the 5th International. Symposium on Communication systems, networks and digital signal processing, pp 314-318.

Gunn, S. R. 1998. Support Vector Machines for Classification and Regression. University of Southampton.

Huang, Y., McCullagh, PJ., and Black, ND. 2004. Feature selection via supervised model construction. Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), pp 411-414.

Joachims T. 1997. A Probabilistic Analysis of The Rocchio Algorithm with TFIDF for Text Categorization. Proceedings of ICML-97 International Conference On Machine Learning.

Jujie, C., and Rongbing, J. 2004. Research of Web Classification Mining Based on RBF Neural Network. Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision Kunming: China, pp. 1365-1367.

Knerr, S., Personnaz, L., and Dreyfus G. 1990. Nurocosingle-layer learning revisited: A stepwise procedure for building and training a neural network, Springer.

Koh, C.H., and Wong, L. 2007. Recognition of polyadenylation sites from Arabidopsis genomic sequences. Proceedings of the 18th International Conference on Genome Informatics (GIW), pp 73—82.

Lan, W.H., and Frank, E. 2005. Data mining: Practical machine learning tools and technique, 2nd Edition Morgan Kaufman: San Francisco.

Li, G. L., and Leong T. Y. 2005. Feature Selection for the Prediction of Translation Initiation Sites. Genomic, Proteomic and Bioinformatics. 3(2): 73-83.

Porter, M. F., 1997. Algorithm for Suffix Stripping. Morgan Kaufmann Multimedia Information and Systems Series, pp. 313–316.

Quinla, J. R. 1993. C.45: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann.

Robnik-Sikonja, M., and Kononenko I. 2003. Theoretical and Empirical Analysis of ReliefF and RReliefF, Machine Learning 53(1-2):23.69.

Rui, F., Alexander, M., and Babis, T., 2006. A Voting Method for the classification of Web Pages. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, pp. 610-613.

Rung-Ching, Chen, Chung-Hsun, Hsieh, 2006. Web Page Classification Based On A Support Vector Machine Using A Weighted Vote Schema. Expert Systems with Applications, vol. 31, pp. 427-435.

Symeonidis, A.L., Nikolaidou, V., and Mitkas, P.A. 2007. Exploiting data mining techniques for Improving the efficiency of a supply chain management agent. Proceedings of the 2006 IEEE/WIC/ACM International conference on Web Intelligence and Intelligent Agent Technology, pp 23-26.

Text Extractor from Web page, Freeware, http://www.nirsoft.net/utils/htmlastext.html.

Vapnik, V. 1998. Statistical learning theory, Wiley: NewYork.

Vidulin, V. 2007. Lustrek M., Gams M., Training the Genre Classifier for Automatic Classification of Web Pages, ITI 2007 29th Internation Conference on Information Technology Interfaces, Cavtat, Croatia, pp. 93-98.

WebKB,1999. Available: http://www.cs.cmu.edu/~WebKB/ (accessed 26/June/2008).

Xin, J., Rongyan L., Xian S., and Rongfang B. 2007. Automatic Web Pages Categorization with ReliefF and Hidden Naive Bayes, Proceedings of the 2007 ACM symposium on Applied computing: Seoul, Korea, pp. 617-621.

Xue, W., Bao, H., W. Huang, and Y. Lu, 2006, Web Page Classification Based on SVM. Proceedings of the 6th World Congress on Intelligent Control and Automation: Dalian, China, pp. 6111-6114.

Xuhong, W., Jinhua, X. 2004. RBF Neural Network Based Predictive Control of Active Power Filter, TENCON 2004. 2004 IEEE Region 10 Conference, vol 4, pp.109-112.

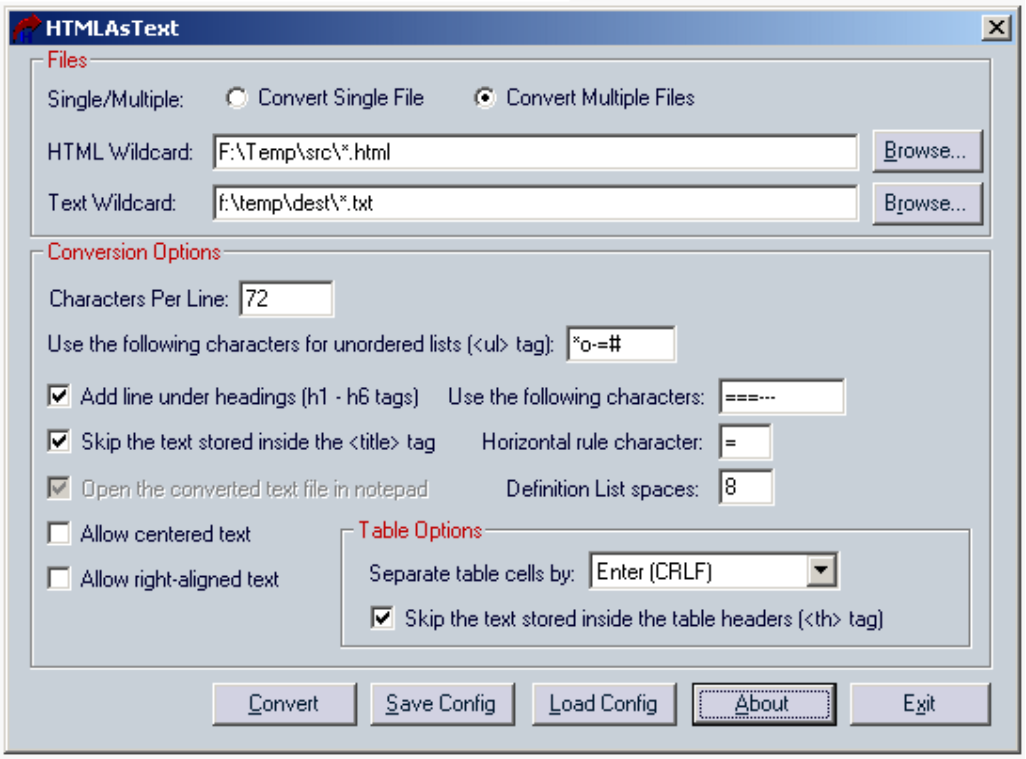
Yahoo Web Pages, 2008. Available: http://www.yahoo.com (accessed 2/July/2008).

ภาคผนวก

ภาคผนวก ก คู่มือการใช้งาน

ก.1 การใช้งานโปรแกรม HTMLAsText

โปรแกรม HTMLAsText (http://www.nirsoft.net/utils/htmlastext.html) เป็นฟรีซอฟต์แวร์ที่ใช้สำหรับการสกัดข้อความจากหน้าเว็บเพจโดยสามารถเลือกชนิดของไฟล์นำเข้าได้ 2 ชนิด คือ 1) สกัดเพียงเว็บเพจเดียว (Convert Single File) 2) สกัดหลายเว็บเพจ (Convert Multiple Files) สำหรับการทำงานจะต้องระบุไฟล์เว็บเพจต้นทางที่ต้องการสกัดข้อความในช่อง HTML Wildcard ซึ่งสามารถระบุชนิดของไฟล์ได้ เช่น *.html หมายถึง สกัดไฟล์เว็บเพจที่มีชนิดเป็น html เท่านั้น และต้องระบุไฟล์ปลายทางข้อความที่สกัดได้ในช่อง Text Wildcard ซึ่งสามารถระบุชนิดของไฟล์ได้ เช่น *.txtหมายถึง ไฟล์ข้อความที่สกัดได้จะมีชนิดเป็น txt (text) จากนั้นจึงกดปุ่ม Convert เพื่อให้โปรแกรมทำงาน



ภาพประกอบ ก.1 โปรแกรม HTMLAsText

ก.2 การใช้งาน Command Line Interface ใน WEKA-3-6

โปรแกรม WEKA (Waikato Environment for Knowledge Analysis) (http://www.cs.waikato.ac.nz/ml/weka) เป็นโปรแกรมที่ใช้ในงานด้านการเรียนรู้ของเครื่อง งานวิจัยนี้ได้ใช้แบบ Command Line User Interface สำหรับการลดขนาดลักษณะเฉพาะ

Package weka.attributeSelection

Interface Summary

AttributeTransformer	Abstract attribute transformer.
ErrorBasedMeritEvaluator	Interface for evaluators that calculate the "merit" of attributes/subsets as the error of a learning scheme
RankedOutputSearch	Interface for search methods capable of producing a ranked list of attributes.
StartSetHandler	Interface for search methods capable of doing something sensible given a starting set of attributes.

Class Summary

ASEvaluation	Abstract attribute selection evaluation class
ASSearch	Abstract attribute selection search class.
AttributeEvaluator	Abstract attribute evaluator.
AttributeSelection	Attribute selection class.
BestFirst	Class for performing a best first search.
CfsSubsetEval	CFS attribute subset evaluator.
ChiSquaredAttributeEval	Class for Evaluating attributes individually by measuring the chi-squared statistic with respect to the class.
ClassifierSubsetEval	Classifier subset evaluator.
ConsistencySubsetEval	Consistency attribute subset evaluator.
ExhaustiveSearch	Class for performing an exhaustive search.
GainRatioAttributeEval	Class for Evaluating attributes individually by measuring gain ratio with respect to the class.
GeneticSearch	Class for performing a genetic based search.
GreedyStepwise	Class for performing a hill climbing search (either forwards or backwards).
HoldOutSubsetEvaluator	Abstract attribute subset evaluator capable of evaluating subsets with respect to a data set that is distinct from that used to initialize/ train the subset evaluator.
InfoGainAttributeEval	Class for Evaluating attributes individually by measuring information gain with respect to the class.
OneRAttributeEval	Class for Evaluating attributes individually by using the OneR classifier.
PrincipalComponents	Class for performing principal components analysis/transformation.
RaceSearch	Class for performing a racing search.
RandomSearch	Class for performing a random search.
Ranker	Class for ranking the attributes evaluated by a AttributeEvaluator Valid options are:
RankSearch	Class for evaluating a attribute ranking (given by a specified evaluator) using a specified subset evaluator.
ReliefFAttributeEval	Class for Evaluating attributes individually using ReliefF.
SubsetEvaluator	Abstract attribute subset evaluator.
SVMAttributeEval	Class for Evaluating attributes individually by using the SVM classifier.
SymmetricalUncertAttributeEval	Class for Evaluating attributes individually by measuring symmetrical uncertainty with respect to the class.
UnsupervisedAttributeEvaluator	Abstract unsupervised attribute evaluator.
UnsupervisedSubsetEvaluator	Abstract unsupervised attribute subset evaluator.
WrapperSubsetEval	Wrapper attribute subset evaluator.

ภาพประกอบ ก.2 แพคเกจ AttributeSelection

ในงานวิจัยนี้ได้ใช้เลือกใช้แพคเกจ Attribute Selection ในการทำงานซึ่งแสดงตัวอย่างการเรียกใช้ Attribute Selection ได้แก่ ReliefFAttributeEval, InfoGainAttributeEval, GainRatioAttributeEval และ ChiSquareAttributeEval แสดงดังภาพประกอบ ก.2 โดยมีตัวอย่างการใช้งานดังนี้

การใช้งานแบบคำสั่ง (Command Line) ใน WEKA มีวิธีการใช้แตกต่างจากการทำงานจาก User Interface โดยจะประกอบด้วยพารามิเตอร์ที่ดังนี้

-S คือ เรียกใช้ class การจัดลำดับของลักษณะเฉพาะแบบ Ranker จาก weka.attributeSelection.Ranker

-N คือ จำนวนลักษณะเฉพาะที่ต้องการ

-E คือ เรียกใช้ class การลดแบบ GainRatioAttributeEval จาก weka.attributeSelection.GainRatioAttributeEval

-i คือ ไฟล์ Input

-o คือ ไฟล์ Output

ตัวอย่างการใช้คำสั่งสำหรับการกรองแอททริบิวมีดังนี้

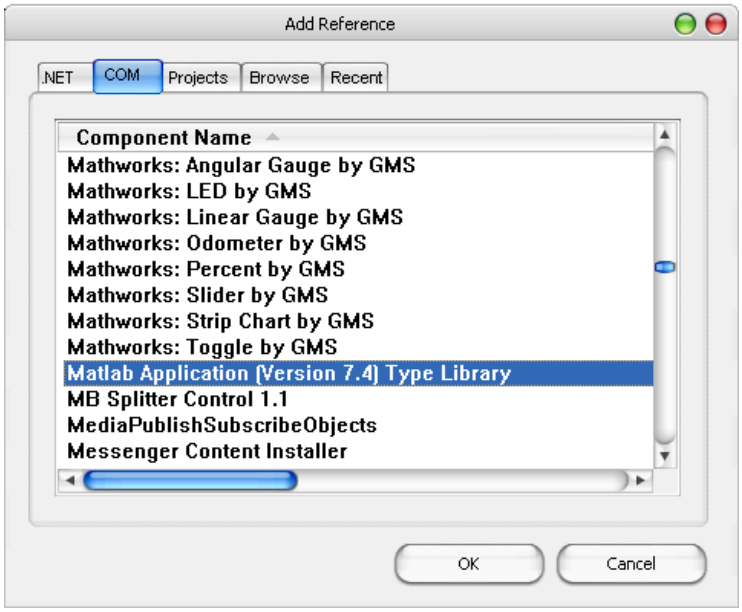
```
java weka.filters.supervised.attribute.AttributeSelection -S "weka.attributeSelection.Ranker -N 50"
-E "weka.attributeSelection.GainRatioAttributeEval" -i web.arff -o webFeature50.arff
```

ในที่นี้จะลดขนาดลักษณะเฉพาะโดยใช้เทคนิค GainRatioAttributeEval ให้มีจำนวนลักษณะเฉพาะที่ต้องการจำนวน 50 โดยมีไฟล์ input คือ web.arff และไฟล์ output คือ webFeature50.arff

ก.3 การสร้างส่วนติดต่อระหว่าง C#.Net กับ MATLAB

สำหรับงานวิจัยนี้ได้ใช้ MATLAB สำหรับสร้างโปรแกรมในการของการจำแนกประเภทโดยต้องเรียกการทำงานจากหน้า user interface ของ C#.net เพื่อให้ส่วนของ MATLAB ทำงานตามโปรแกรมที่ได้เขียนไว้ ซึ่งมีวิธีการสร้างส่วนติดต่อระหว่างสองโปรแกรมข้างต้นดังนี้

ก.3.1 เพิ่มส่วน Library ของ MATLAB เข้าไปใน C#.net โดยเลือก Add Reference จากนั้นไปที่แท็บ COM แล้วเลือก Matlab Application (Version 7.4) Type Library



ภาพประกอบ ก.3 แพคเกจ AttributeSelection

ก.3.2 เขียนคำสั่งสำหรับการทำงานดังตัวอย่างภาพประกอบ ก.4 บรรทัดที่ 1 สร้าง object ขึ้นมาใหม่สำหรับติดต่อกับ MATLAB บรรทัดที่ 2 เขียนคำสั่งที่ใช้ในการทำงานของ MATLAB บรรทัดที่ 3 หยุดการเชื่อมต่อระหว่าง C#.Net กับ MATLAB

```
1: MLApp.MLAppClass matlab = new MLApp.MLAppClass();
2: result = matlab.Execute(command); //Command คือ คำสั่งในการทำงานกับ MATLAB
3: matlab.Quit();
```

ภาพประกอบ ก.4 คำสั่งสำหรับเชื่อมต่อการทำงานระหว่าง C#.Net กับ MATLAB

ภาคผนวก ข

ผลงานตีพิมพ์

เรื่อง การจำแนกประเภทเว็บเพจโดยใช้ค่าความถี่เอกสารและซัพพอร์ตเวกเตอร์แมชชีน

Conference The 12th National Computer Science and Engineering Conference (NCSEC 2008)

สถานที่ พัทยา ประเทศไทย

วันที่ 20-21 พฤศจิกายน 2551

การจำแนกประเภทเว็บเพจโดยใช้ค่าความถี่เอกสารและซัพพอร์ตเวกเตอร์แมชชีน

Web Page Classification Using Document Frequency and Support Vector Machine

พรพล ธรรมรงค์รัตน์[1] ลัดดา ปรีชาวีรกุล[2] และ วิภาดา เวทย์ประสิทธิ์[1]

ห้องปฏิบัติการวิจัยปัญญาประดิษฐ์[1] ห้องปฏิบัติการวิจัยเทคโนโลยีระบบสารสนเทศและการประยุกต์[2] ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ จังหวัดสงขลา 90112 โทรศัพท์: 0-7428-8581 โทรสาร: 0-7444-6917

E-mail: pthamrongrat@hotmail.com, ladda.p@psu.ac.th, wwettayaprasit@yahoo.com

บทคัดย่อ

จำนวนเว็บเพจที่มากขึ้นบนโลกอินเทอร์เน็ตส่งผลให้การสืบค้นเอกสารที่ตรงกับความต้องการของผู้ใช้มีประสิทธิภาพลดลง การจำแนกเว็บเพจเป็นแนวทางหนึ่งในการแก้ปัญหานี้ บทความนี้ได้นำเสนอการจำแนกประเภทเว็บเพจโดยใช้ความถี่เอกสาร (Document Frequency) ของวิธี Term Frequency-Inverse Document Frequency (TF-IDF) เพื่อลดขนาดคุณลักษณะโดยใช้ค่า Threshold (λ) ของความถี่เอกสารที่แตกต่างกัน ทำการทดลองกับเอกสารเว็บเพจจาก WebKB และเว็บเพจข่าวกีฬาจาก Yahoo ผลการทดลองเปรียบเทียบประสิทธิผลระหว่าง Support Vector Machine (SVM) ต้นไม้ตัดสินใจ C4.5 และ Radial Basis Function Neural Networks (RBFNN) แสดงให้เห็นว่าเทคนิคที่นำเสนอด้วยวิธี SVM ให้ค่า F-measure ที่ดีที่สุด

คำสำคัญ: การจำแนกประเภทเว็บเพจ, การลดขนาดคุณลักษณะ, ค่าความถี่เอกสาร, ซัพพอร์ตเวกเตอร์แมชชีน, C4.5, RBFNN

Abstract

The increasing number of web pages on the cyber world results to the less effectiveness of document retrieval that matches the need of users. The classification of web pages is one of the solutions to solve this problem. This paper presents the classification of web pages by using Document Frequency from Term Frequency-Inverse Document Frequency (TF-IDF) to reduce the size of input features by using the document frequency values that have different threshold values. The study examines web page documents from WebKB and sport news web page from Yahoo website. The experimental results show the comparisons on the effectiveness among Support Vector Machine (SVM), decision tree C4.5 and Radial Basic Function Neural Networks (RBFNN) which indicate that the proposed technique using SVM gives the best values of F-measure.

Keywords: Web Page Classification, Feature Reduction, Document Frequency, Support Vector Machine, C4.5, RBFNN

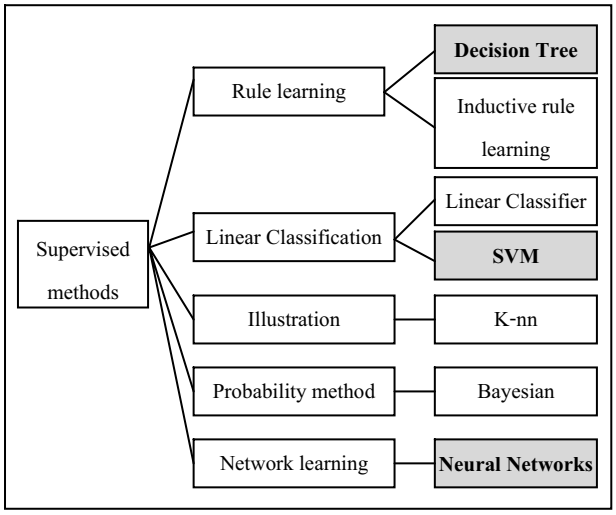
1. บทนำ

การพัฒนาอย่างรวดเร็วของเทคโนโลยีสารสนเทศ ส่งผลให้ข้อมูลข่าวสารรวมถึงความรู้ต่าง ๆ ถูกพัฒนาอยู่ในรูปเว็บเพจ โดยมีประโยชน์เพื่ออำนวยความสะดวกและเพิ่มความรวดเร็วในการสืบค้นให้กับผู้ใช้ อย่างไรก็ตามเมื่อจำนวนของเว็บเพจมีขนาดมากขึ้น ทำให้การสืบค้นช้าลง และได้ประสิทธิผลไม่ตรงกับความต้องการของผู้ใช้ แนวทางหนึ่งที่สามารถแก้ปัญหานี้ได้ คือการจำแนกประเภทของเว็บเพจ แต่เนื่องจากเว็บเพจประกอบด้วยตัวอักษรซึ่งมีความยาวที่แตกต่างกัน ส่งผลให้ขนาดของคุณลักษณะของข้อมูลเข้า (Input Feature) มีขนาดใหญ่ตามไปด้วย สิ่งนี้เป็นอุปสรรคในการจำแนกประเภท ดังนั้นงานวิจัยนี้จึงนำเสนอวิธีในการลดขนาดคุณลักษณะของข้อมูลเข้าโดยใช้ค่าความถี่เอกสารและทำการเปรียบเทียบกับตัวจำแนกประเภทที่แตกต่างกัน 3 ประเภท ได้แก่ Support Vector Machine (SVM) ต้นไม้ตัดสินใจ C4.5 และ Radial Basis Function Neural Networks (RBFNN) เพื่อพิจารณาประสิทธิผล

บทความนี้ประกอบด้วยส่วนต่างๆ ดังต่อไปนี้ ส่วนที่ 2 จะกล่าวถึงงานวิจัยที่เกี่ยวข้อง ส่วนที่ 3 คือขั้นตอนการจำแนกประเภทเว็บเพจ ส่วนที่ 4 วิธีการทดลองและผลการทดลอง ส่วนที่ 5 คือบทสรุป และส่วนที่ 6 คืองานวิจัยที่ดำเนินการต่อไป

2. งานวิจัยที่เกี่ยวข้อง

จากการศึกษาพบว่ามีงานวิจัยจำนวนมาก ได้ถูกพัฒนาขึ้นเพื่อจำแนกประเภทเว็บเพจโดยใช้ตัวจำแนกประเภทที่ต่างกัน ตัวอย่างเช่น SVM เป็นวิธีการเรียนรู้แบบมีผู้สอน Gunn [1] ได้ใช้ส่วนของข้อมูลเพื่อสอนระบบซึ่งสามารถทำนายประเภทของเอกสารได้ Xue และคณะ [2] สกัดคุณลักษณะจากเว็บเพจ โดยใช้ SVM เพื่อประเมินประสิทธิผลของการจำแนกประเภทเว็บเพจ ด้วยเคอร์แนลฟังก์ชัน (Kernel Function) ที่ต่างกัน และใช้ Radial Basis Function Neural Networks (RBFNN) เพื่อเปรียบเทียบประสิทธิผลของการทดลอง Jujie และ Rongbing [3] ออกแบบการจำแนกประเภทเว็บเพจโดยใช้ RBFNN เปรียบเทียบกับ Back Propogation (BP) ผลการทดลองพบว่า RBFNN ให้ประสิทธิผลที่ดีกว่า Vidulin และคณะ [4] ได้ใช้ URL HTML และข้อความ (Text) เป็นคุณลักษณะของข้อมูลเข้า สำหรับจำแนก



รูปที่ 1 วิธีการจำแนกประเภทโดยมีผู้สอน [5]

ประเภทเว็บเพจซึ่งได้ประเมินประสิทธิผลของการทดลอง โดยใช้ต้นไม้ตัดสินใจ C4.5 และตัวจำแนกประเภท Bagging Ensembles จากงานวิจัยข้างต้นพบว่ามีตัวจำแนกประเภทที่หลากหลายซึ่งให้ประสิทธิผลที่แตกต่างกัน โดยสามารถแสดงหมวดหมู่ของการจำแนกประเภทแบบมีผู้สอน [5] ดังรูปที่ 1 อย่างไรก็ตาม Sebastian [6] แสดงให้เห็นว่าไม่มีวิธีการใด ที่สามารถให้ประสิทธิผลที่ดีที่สุดกับทุกข้อมูลโดยจะขึ้นอยู่กับข้อมูลนั้น ๆ ด้วยว่าเหมาะสมกับวิธีใด

3. ขั้นตอนการจำแนกประเภทเว็บเพจ

บทความนี้นำเสนอขั้นตอนการจำแนกเว็บเพจโดยใช้ค่าความถี่เอกสาร ร่วมกับ SVM (Web Page Classification Using Document Frequency and Support Vector Machine: WPC_DF_SVM) ซึ่งประกอบด้วยขั้นตอนทั้งหมด 3 ขั้นตอนดังรูปที่ 2 คือ ขั้นตอนที่ 1 การเตรียมข้อมูลเพื่อนำข้อความจากเว็บเพจมาผ่านกระบวนการ ให้อยู่ในลักษณะที่เหมาะสมในการวิเคราะห์ข้อมูลสำหรับดำเนินการในขั้นตอนต่อไป ขั้นตอนที่ 2 คือ การลดขนาดคุณลักษณะ เป็นขั้นตอนที่นำคำที่ได้มาทำดัชนี ซึ่งผ่านการลดขนาดคุณลักษณะด้วยค่าความถี่เอกสารตามค่า Threshold ที่ได้กำหนดไว้ ด้วยวิธีการ TF-IDF และขั้นตอนที่ 3 คือ การประเมินประสิทธิผล เป็นขั้นตอนการประเมินผลด้วยค่า F-measure ของตัวจำแนกประเภท 3 ประเภทได้แก่ SVM ต้นไม้ตัดสินใจ C4.5 และ RBFNN

3.1 ขั้นตอนที่ 1 การเตรียมข้อมูล (Preprocessing)

ขั้นตอนนี้ประกอบด้วย 3 ขั้นตอนย่อยดังนี้

3.1.1 การสกัดข้อความ (Text Extraction)

การสกัดข้อความโดยใช้โปรแกรม HTMLAsText [7] เป็นเครื่องมือในการสกัดข้อความ ซึ่งจะสกัดเฉพาะส่วนของข้อความ (Text) ที่ปรากฏอยู่บนหน้าเว็บเพจเท่านั้น ตัวอย่างเช่น ข่าวกีฬาเทนนิสจากเว็บไซต์ Yahoo ในรูปที่ 3 หลังจากใช้โปรแกรม HTMLAsText แล้วผลลัพธ์ที่ได้จะเป็นเนื้อความของเอกสารเว็บเพจนั้นซึ่งจะไม่มีส่วน HTML tag รูปภาพ หรือสื่อมัลติมิเดีย ปรากฏอยู่ ดังแสดงในรูปที่ 4

ขั้นตอนที่ 1 : การเตรียมข้อมูล

1.1 สกัดข้อความ (Text Extraction) จากหน้าเว็บเพจ

1.2 หารากศัพท์ของคำ (Stemming) โดยใช้อัลกอริทึม Porter

1.3 กำจัดคำหยุด (Stopping) ซึ่งเป็นคำที่ไม่มีความจำเป็น

↓

ขั้นตอนที่ 2 : การลดขนาดคุณลักษณะ

2.1 สร้างคุณลักษณะ (Feature Extraction) ให้อยู่ในรูป Term Frequency Table

2.2 ลดขนาดคุณลักษณะ (Feature Reduction) ด้วยค่าความถี่เอกสารด้วยค่าที่กำหนด (Threshold)

2.3 ทำดัชนี (Indexing) โดยการกำหนดน้ำหนักของแต่ละคำโดยใช้วิธีการ TF-IDF

↓

ขั้นตอนที่ 3 : การประเมินประสิทธิผล

3.1 จำแนกประเภทด้วยตัวจำแนกประเภท 3 ประเภทได้แก่ Support Vector Machine ต้นไม้ตัดสินใจ C4.5 และ RBFNN

3.2 ประเมินประสิทธิผลค่า F-measure ด้วยการเปรียบเทียบตัวจำแนกประเภททั้ง 3 ประเภท

รูปที่ 2 ขั้นตอนการจำแนกประเภทเว็บเพจ WPC_DF_SVM

รูปที่ 3 ตัวอย่างเว็บเพจข่าวกีฬาจากเว็บไซต์ Yahoo ข่าวกีฬา

Rain delays start of men's quarters - Tennis - Yahoo! Sports Web Search * New User? * Sign Up * Sign In - Rain delayed five-times champion Roger Federer from starting his quarter-final against Mario Ancic at Wimbledon on Wednesday and the match got underway 58 minutes late just before 1300 GMT (9:00 a.m. EDT) on Centre Court.

รูปที่ 4 ตัวอย่างข้อความที่สกัดได้จากหน้าเว็บไซต์ Yahoo ข่าวกีฬา ด้วยโปรแกรม HTMLAsText

3.1.2 การหารากศัพท์ (Stemming)

เมื่อได้ข้อความจากขั้นตอนการสกัดคำแล้ว ต่อไปจึงนำคำเหล่านี้มาหารากศัพท์โดยใช้อัลกอริทึม Porter [8] ซึ่งเป็นอัลกอริทึมที่ได้รับความนิยม โดยปรับเปลี่ยนคำท้าย (Suffix) ในภาษาอังกฤษ แล้วจึงกำหนดให้ตัวอักษรตัวเล็กทั้งหมด ดังตารางที่ 1 จะเห็นได้ว่า “Run” และ “Running” จะเปลี่ยนเป็นรากศัพท์เดียวกันคือ “run” เป็นต้น

ตารางที่ 1 ตัวอย่างคำที่ผ่านการหารากศัพท์ด้วยอัลกอริทึม Porter

คำต้นฉบับ	คำที่ผ่านการหารากศัพท์
andy	andi
murray	murrai
Run	run
Running	run

3.1.3 การกำจัดคำหยุด (Stopping)

การกำจัดคำหยุด เป็นกระบวนการหนึ่งที่สำคัญสำหรับการค้นคืนสารสนเทศ เนื่องจากคำบางคำไม่มีความจำเป็นสำหรับการนำมาวิเคราะห์ ดังนั้นจึงควรกำจัดคำเหล่านี้ออกไปเพื่อให้เหลือเฉพาะคำที่มีความสำคัญเท่านั้น คำเหล่านี้เรียกว่าคำ “Stoplist” เช่น คำว่า “a” “been” และ “ago” เป็นต้น ตัวอย่างของคำที่เป็นคำหยุดแสดงดังตารางที่ 2

ตารางที่ 2 ตัวอย่างคำหยุด (Stoplist Word)

a	been	get	least	our
about	before	getting	left	ourselves
after	being	go	less	out
again	between	goes	let	over
ago	but	going	like	per
all	by	gone	make	put
also	can	gotten	may	same

3.2 ขั้นตอนที่ 2 การลดขนาดคุณลักษณะ

ขั้นตอนนี้ประกอบด้วย 3 ขั้นตอนย่อยดังนี้

3.2.1 การสร้างคุณลักษณะ (Feature Extraction)

โดยนำคำที่ได้จากขั้นตอนการเตรียมข้อมูลมาสร้างให้อยู่ในรูปของ Term Frequency Table ตัวอย่างเช่น มีเอกสารเว็บเพจข่าวกีฬาจำนวน 3 เว็บเพจ และมีคำอยู่ 4 คำคือ play, tennis, win และ knockout สามารถนำมาสร้างให้อยู่ในรูปของ Term Frequency และ Document Frequency ดังตารางที่ 3 โดยที่กำหนดให้ *TF* (Term Frequency) หมายถึงค่าความถี่ของคำแต่ละคำที่ปรากฏอยู่ในเอกสาร ตัวอย่างเช่น คำว่า “play” ปรากฏอยู่ในเอกสารฉบับที่ 1 (Doc_1) จำนวน 3 ครั้ง ปรากฏอยู่ในเอกสารฉบับที่ 2 (Doc_2) จำนวน 4 ครั้ง และปรากฏอยู่ในเอกสารฉบับที่ 3 (Doc_3) จำนวน 0 ครั้งหรือไม่ปรากฏอยู่ในเอกสารฉบับนี้ เป็นต้น โดย *DF* (Document Frequency) หมายถึง ความถี่เอกสารของคำที่พิจารณาปรากฏในเอกสารทั้งหมดกี่ฉบับ เช่น คำว่า “play” ปรากฏอยู่ในเอกสาร 2 ฉบับ ได้แก่ เอกสารฉบับที่ 1 (Doc_1) และเอกสารฉบับที่ 2 (Doc_2) ดังนั้นจึงมีค่าค่าความถี่เอกสารเป็น 2 และคำว่า “tennis” ปรากฏอยู่ในเอกสารทั้งหมด 3 ฉบับ ได้แก่ เอกสารฉบับที่ 1 เอกสารฉบับที่ 2 และเอกสารฉบับที่ 3 จึงมีค่าความถี่เอกสารเป็น 3 เป็นต้น

ตารางที่ 3 ตัวอย่าง Term Frequency Table

เอกสาร	ความถี่ของคำ *TF* (Term Frequency) ในแต่ละเอกสาร (Doc)			
	play	**tennis**	**win**	**knockout**
Doc_1	3	5	0	1
Doc_2	4	2	0	1
Doc_3	0	1	3	1
ความถี่เอกสาร *DF* (Document Frequency)	**2**	**3**	**1**	**3**

3.2.2 การลดขนาดคุณลักษณะ (Feature Reduction)

เนื่องจากขนาดคุณลักษณะของการจำแนกประเภทเว็บเพจมีจำนวนมาก การลดจำนวนคุณลักษณะจึงเป็นสิ่งจำเป็นอย่างยิ่ง ในงานวิจัยนี้ได้ใช้ค่าความถี่ของเอกสาร (Document Frequency) ที่คุณลักษณะนั้นปรากฏอยู่ โดยจะเลือกคุณลักษณะที่มากกว่าหรือเท่ากับค่า Threshold ที่กำหนดไว้ ถ้ากำหนดค่า Threshold (λ) ให้มีค่ามากกว่าหรือเท่ากับ 3 จากตารางที่ 3 ค่าความถี่เอกสาร (Document Frequency) ของคำว่า (“play” = 2, “tennis” = 3, “win” = 1, “knockout” = 3) เราจะได้คำที่ผ่านค่า λ เพียง 2 คำ คือคำว่า “tennis” และ “knockout” เพื่อใช้ในการทำงานในขั้นต่อไป และตัดคำว่า “play” และ “win” ทิ้งไป

3.2.3 การทำดัชนี (Indexing)

เมื่อผ่านการลดขนาดคุณลักษณะแล้วจะนำคำที่ได้เหล่านั้นมาให้ค่าน้ำหนักโดยใช้วิธีการ TF-IDF [9] โดย *TF* (Term Frequency) คือความถี่ของแต่ละคำที่ปรากฏในเอกสาร และ *IDF* (Inverse Document Frequency) คือ ส่วนกลับของความถี่เอกสารที่ปรากฏคำนั้นอยู่ สามารถคำนวณได้ดังสมการที่ (1)

$$IDF_k = \log_2(n / DF_k) \quad (1)$$

โดยที่ n คือ จำนวนเอกสาร (เว็บเพจ) และ DF_k คือ ความถี่ของเอกสารที่ปรากฏอยู่ในเอกสารทั้งหมดของคำที่ k โดยที่ $k = 1, \ldots, m$ เมื่อ m คือ จำนวนคำทั้งหมดจากทุกเอกสาร ตัวอย่างเช่น จากตารางที่ 3 ค่า $n = 3$ และ $m = 4$ จากนั้นคำนวณค่า TF-IDF ได้ดังสมการที่ (2)

$$W_{jk} = TF_{jk} \times IDF_k \quad (2)$$

โดยที่ W_{jk} คือ ค่า TF-IDF ของคำที่ k ในเอกสารที่ j โดย $j = 1, \ldots, n$ เมื่อ n คือ จำนวนของเอกสารทั้งหมด จากนั้นเมื่อคำนวณ

ค่า TF-IDF ของทุกคำแล้ว นำค่าที่ได้จากการคำนวณนี้มาสร้างให้อยู่ในรูปแบบ Document Word Matrix ดังตารางที่ 4

ตารางที่ 4 Document Word Matrix

เอกสาร (Web Page)	ค่า TF-IDF			
	W_1	W_2	…	W_m
Doc_1	W_{11}	W_{12}	...	W_{1m}
Doc_2	W_{21}	W_{22}	...	W_{2m}
…	...	...	...	...
Doc_n	W_{n1}	W_{n2}	...	W_{nm}

3.3 ขั้นตอนที่ 3 การประเมินประสิทธิผล

ขั้นตอนนี้ประกอบด้วย 2 ขั้นตอนย่อยดังนี้

3.3.1 การจำแนกประเภทเว็บเพจ (Web Page Classification)

การทดลองนี้ใช้ตัวจำแนกประเภท (Classifier) 3 ประเภท คือ Support Vector Machine (SVM) ต้นไม้ตัดสินใจ C4.5 และ RBF Neural Networks

3.3.1.1 Support Vector Machine (SVM)

แนวคิดหลักของ SVM [10] นำมาใช้เพื่อหาระนาบการตัดสินใจในการแบ่งข้อมูลออกเป็นสองส่วน ใช้สำหรับข้อมูลที่มีมิติของข้อมูลสูง กำหนดให้ $(x_1, y_1), \ldots, (x_n, y_n)$ เป็นตัวอย่างที่ใช้สำหรับการสอน n คือ จำนวนข้อมูลตัวอย่าง m คือ จำนวนมิติข้อมูลเข้า และ y คือ ผลลัพธ์ มีค่า +1 หรือ -1 ดังสมการที่ (3)

$$(x_i, y_i), \ldots, (x_n, y_n) \text{ เมื่อ } x \in R^m \ , y \in \{+1, -1\} \quad (3)$$

สำหรับปัญหาเชิงเส้น มิติข้อมูลขนาดสูงได้ถูกแบ่งเป็น 2 กลุ่มโดยระนาบตัดสินใจ ซึ่งคำนวณได้ดังสมการที่ (4)

$$(w \cdot x) + b = 0 \quad (4)$$

เมื่อ w คือ ค่าน้ำหนัก และ b คือ ค่า bias สมการที่ (5) ใช้สำหรับจำแนกประเภทของข้อมูล

$$(w \cdot x) + b > 0 \text{ ถ้า } y_i = +1 \text{ และ } (w \cdot x) + b < 0 \text{ ถ้า } y_i = -1 \quad (5)$$

อย่างไรก็ตาม SVM มีเคอร์เนลฟังก์ชัน (Kernel Function) ที่ผู้ใช้สามารถประยุกต์ใช้ในการแก้ปัญหาได้หลายวิธี สำหรับงานวิจัยนี้ได้เลือก Polynomial Kernel เป็นฟังก์ชันที่ใช้ในการทดลอง

3.3.1.2 ต้นไม้ตัดสินใจ C4.5

การสร้างต้นไม้ตัดสินใจ C4.5 [11] ใช้ค่ามาตรฐานอัตราส่วนเกน (Gain Ratio) เพื่อเลือกคุณลักษณะที่จะใช้เป็นรากหรือโหนด ถ้าให้ชุดของข้อมูล M ประกอบด้วยค่าที่เป็นไปได้ คือ $\{m_1, m_2, \ldots, m_n\}$ และให้ความน่าจะเป็นที่จะเกิดค่า m_i มีค่าเท่ากับ $P(m_i)$ จะได้ว่าค่าเกนสารสนเทศ (Information Gain) ของ M เขียนแทนด้วย $I(M)$ คำนวณได้ดังสมการที่ (6)

$$I(M) = \sum_{i=1}^{n} - P(m_i) \log_2 P(m_i) \quad (6)$$

ถ้าให้ข้อมูลสอน คือ T และคุณลักษณะที่เป็นโหนด คือ x และมีค่าทั้งหมดที่เป็นไปได้ n ค่า โหนดปัจจุบันจะแบ่งตัวอย่าง T ออกตามกิ่งเป็น $\{t_1, t_2, \ldots, t_n\}$ ตามค่าที่เป็นไปได้ของ x ดังนั้นจึงสามารถคำนวณค่าเกนสารสนเทศหลังจากแบ่งตามคุณลักษณะ x ได้ดังสมการที่ (7)

$$I_x(T) = \sum_{i=1}^{n} \frac{|t_i|}{|T|} I(t_i) \quad (7)$$

ค่ามาตรฐานเกน (Gain) ของคุณลักษณะ x ได้ดังสมการที่ (8)

$$Gain(x) = I(T) - I_x(T) \quad (8)$$

จากนั้นคำนวณค่าสารสนเทศของการแบ่งแยก (Split Information) ของคุณลักษณะแต่ละตัว ถ้าให้ T คือ ชุดของตัวอย่าง เมื่อแบ่งตัวอย่างนี้ตามคุณลักษณะ x จะได้ชุดของตัวอย่างย่อยในแต่ละกิ่ง คือ $\{t_1, t_2, \ldots, t_n\}$ จำนวน n ชุด ตามค่าที่เป็นไปได้ในคุณสมบัติ x เมื่อคำนวณค่าสารสนเทศของการแบ่งแยกได้ดังสมการที่ (9)

$$Split\ Information = - \sum_{i=1}^{n} \frac{|t_i|}{|T|} \log_2 \frac{|t_i|}{|T|} \quad (9)$$

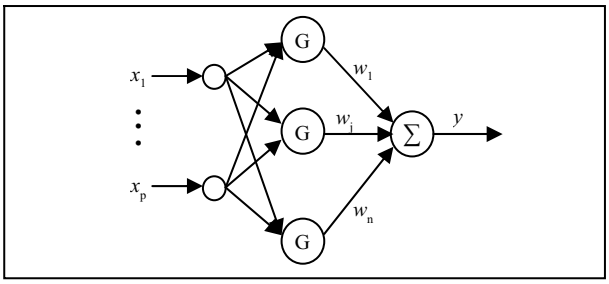
คำนวณค่ามาตรฐานอัตราส่วนเกน (Gain Ratio) ได้ดังสมการที่ (10)

$$Gain\ Ratio = Gain - Split\ Information \quad (10)$$

ท้ายสุดจึงเลือกค่า *Gain Ratio* สูงสุดเป็นคุณลักษณะเริ่มต้น และเลือกคุณลักษณะถัดไปตามค่า *Gain Ratio* น้อยลงตามลำดับ

3.3.1.3 Radial Basis Function Neural Networks (RBFNN)

RBF Neural Networks [12, 13] ประกอบด้วยชั้นข้อมูลเข้า (Input Layer) ชั้นซ่อน (Hidden Layer) และชั้นข้อมูลออก (Output Layer) ดังรูปที่ 5 ซึ่งมีเกาซ์เซียนฟังก์ชัน (Gaussian Function) เป็นฟังก์ชันกระตุ้นในชั้นซ่อน ดังสมการที่ (11)



รูปที่ 5 RBF Neural Networks

$$\phi_j(x) = \exp\left[- \frac{\left\| x - c_j E \right\|}{2\sigma_j^2} \right] \text{ เมื่อ } j = 1,2,\ldots,n \quad (11)$$

โดยที่ ϕ คือ ข้อมูลออกของนิวรอลที่ j ในชั้นซ่อน x คือ เวกเตอร์ข้อมูลเข้า c_j และ σ_j คือ ศูนย์กลาง และช่วงกว้างของนิวรอลที่ j ตามลำดับ ข้อมูลออกของโครงข่าย RBF คำนวณดังสมการที่ (12)

$$y = i_c(k+1) = \sum_{j=1}^{n} w_j \phi_j(x) \quad (12)$$

โดยที่ n คือ จำนวนของนิวรอลในชั้นซ่อน w_j คือน้ำหนักระหว่างชั้นซ่อนและชั้นข้อมูลออก และ y คือผลลัพธ์

3.3.2 การประเมินประสิทธิผล

นิยมใช้วิธีทางด้านการค้นคืนสารสนเทศ ซึ่งผลของการจำแนกประเภทเว็บเพจสามารถเขียนเป็นตารางตัดสินใจการจำแนกประเภท ได้ดังตารางที่ 5 และประเมินค่าด้วยการวัดค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) และค่า F (F-measure)

ตารางที่ 5 ตารางตัดสินใจการจำแนกประเภท

ค่าที่จำแนกโดยตัวจำแนกประเภท	ค่าของ C_i	
	ใช่	ไม่ใช่
ใช่	TP_i	FP_i
ไม่ใช่	FN_i	TN_i

กำหนดให้ i คือ ลำดับของ Class (C) ที่ i จากตารางที่ 5 จะได้

TP_i (True Positive) คือ จำนวนเว็บเพจที่อยู่ในประเภท C_i และตัวจำแนกประเภททำนายว่าอยู่ในประเภท C_i

FP_i (Fasle Positive) คือ จำนวนเว็บเพจที่ไม่อยู่ในประเภท C_i แต่ตัวจำแนกประเภททำนายว่าอยู่ในประเภท C_i

FN_i (Fasle Negative) คือ จำนวนเว็บเพจที่อยู่ในประเภท C_i แต่ตัวจำแนกประเภททำนายว่าไม่อยู่ในประเภท C_i

TN_i (True Negative) คือ จำนวนเว็บเพจที่ไม่อยู่ในประเภท C_i และตัวจำแนกประเภททำนายว่าไม่อยู่ในประเภท C_i

สามารถคำนวณค่าความแม่นยำ (P_i) ค่าระลึก (R_i) และค่า F-measure (F_i) ได้ดังสมการที่ (13)-(15) ดังนี้

$$P_i = \frac{TP_i}{TP_i + FP_i} \quad (13)$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \quad (14)$$

$$F_i = \frac{2P_iR_i}{P_i + R_i} \quad (15)$$

ข้อสังเกต การวัดประสิทธิผลของการจำแนกประเภท ควรพิจารณาค่าความแม่นยำ และค่าระลึกร่วมกัน กับค่า F-measure เนื่องจากในการจำแนกประเภทนั้น กรณีที่มีค่า FN_i มาก แม้จะมีค่าความแม่นยำสูง (P_i) อาจมีค่าระลึก (R_i) ที่ต่ำได้ และในกรณีที่ ถ้ามีค่า FP_i มากแม้มีค่าความแม่นยำต่ำ (P_i) อาจให้ค่าความระลึก (R_i) ที่สูง เป็นต้น

4. วิธีการทดลองและผลการทดลอง

4.1 ชุดข้อมูลทดสอบ

งานวิจัยนี้ได้ใช้เอกสารเว็บเพจ 2 ชุดข้อมูล เพื่อทำการทดลอง โดยชุดข้อมูลแรกเป็นเอกสารเว็บเพจจาก WebKB [14] ซึ่งนิยมใช้กันอย่างแพร่หลายสำหรับการจำแนกประเภทเว็บเพจ ประกอบด้วยข้อมูล 7 ประเภท โดยการสุ่มจำนวน 271 เว็บเพจ จากข้อมูลทั้งหมด 8,282 เว็บเพจ ได้แก่ 1) Material จำนวน 49 เว็บเพจ 2) Energy จำนวน 39 เว็บเพจ 3) Financial จำนวน 21 เว็บเพจ 4) Healthcare จำนวน 43 เว็บเพจ 5) Technology จำนวน 45 เว็บเพจ 6) Transportation 36 เว็บเพจ และ 7) Utilities 38 เว็บเพจ ชุดข้อมูลที่สองเอกสารเว็บเพจจากเว็บไซต์ Yahoo [15] ซึ่งเป็นเว็บเพจที่มีเฉพาะเนื้อหาเกี่ยวกับข่าวกีฬาจำนวน 5 ประเภท ได้แก่ 1) Boxing 2) Golf 3) Soccer 4) Tennis และ 5) Cycling โดยใช้ข้อมูลประเภทละ 50 เว็บเพจ รวมเป็น 250 เว็บเพจ ทำการทดสอบแบบ 10-Fold Cross Validation โดยใช้โปรแกรม WEKA [16]

4.2 ผลการทดลอง

จากผลการทดลองทั้ง 2 ชุดข้อมูล โดยทดสอบกับค่าความถี่เอกสารต่าง ๆ เปรียบเทียบกับตัวจำแนกประเภททั้ง 3 ประเภทตามขั้นตอนการจำแนกเว็บเพจ WPC_DF_SVM จากนั้นจึงนำค่า F-measure ที่ได้ของทุกประเภท (Class) มาหาค่าเฉลี่ย โดยสามารถพิจารณาประสิทธิภาพของขั้นตอนวิธีดังประเด็นต่อไปนี้

ตารางที่ 6 ค่าเฉลี่ย F-measure ชุดข้อมูลที่ 1 เว็บเพจ WebKB

ความถี่เอกสาร (Document Frequency)	จำนวนคุณลักษณะ	ค่าเฉลี่ย F-measure (%)		
		SVM	C4.5	RBF
$\lambda \geq 0$	12,722	**93.19**	**89.57**	**74.06**
$\lambda \geq 5$	1,666	92.68	89.22	72.42
$\lambda \geq 10$	812	90.98	89.50	67.96
$\lambda \geq 15$	511	90.98	89.50	67.96
$\lambda \geq 20$	357	91.57	85.22	63.91
$\lambda \geq 25$	262	91.53	84.90	65.55
$\lambda \geq 30$	187	89.10	86.23	67.08
$\lambda \geq 35$	144	88.78	88.06	65.04
$\lambda \geq 40$	120	88.75	88.08	62.39
$\lambda \geq 45$	102	88.44	88.74	62.62
$\lambda \geq 50$	88	89.56	87.56	63.33

4.2.1 ประสิทธิภาพค่าความถูกต้องการลดขนาดคุณลักษณะ

จากตารางที่ 6 ของข้อมูลชุดที่ 1 เมื่อลดขนาดคุณลักษณะโดยใช้ความถี่เอกสาร (Document Frequency) ที่มีค่า Threshold (λ) ต่างๆ คือ $\geq 0, \geq 5, \ldots, \geq 50$ ตามลำดับ สามารถลดจำนวนคุณลักษณะจาก 12,722 จำนวนเหลือเพียง 88 จำนวน และจากตารางที่ 7 ของข้อมูลชุดที่ 2 สามารถลดจำนวนคุณลักษณะจาก 9,608 จำนวนเหลือเพียง 492 จำนวน

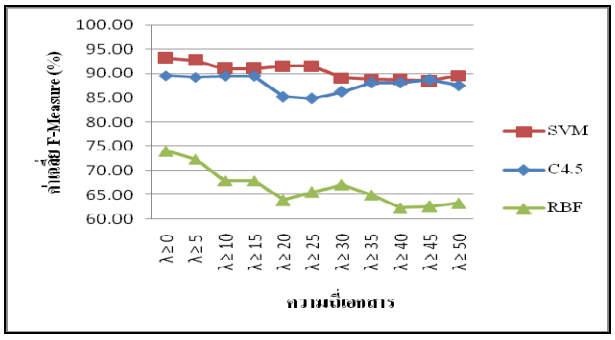
เท่านั้น โดยพบว่าจากข้อมูลชุดที่ 1 SVM ยังคงให้ค่าความถูกต้องที่ดี โดยมีค่าความต่างระหว่างค่าเฉลี่ย F-measure มากที่สุด 93.19% และค่าเฉลี่ย F-measure น้อยที่สุด 88.44% มีค่าประมาณ 5% ผลการทดลองข้อมูลชุดที่ 2 SVM มีค่าเฉลี่ย F-measure 100% ทั้งในกรณีก่อนการลดขนาดคุณลักษณะและหลังการลดขนาดคุณลักษณะ

4.2.2 ประสิทธิภาพของตัวจำแนกประเภท

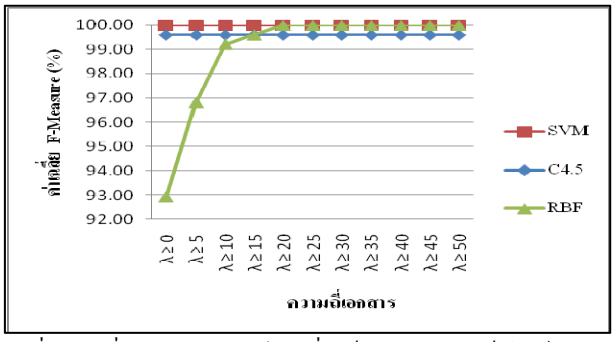
เมื่อพิจารณาตัวจำแนกทั้ง 3 ประเภท จากข้อมูลชุดที่ 1 WebKB รูปที่ 6 แสดงให้เห็นว่าเมื่อเลือก (λ) มากกว่าหรือเท่ากับ 0 จนถึงค่า (λ) ที่มากกว่าหรือเท่ากับ 50 SVM ให้ค่า F-measure สูงที่สุด รองลงมาคือ C4.5 และ RBFNN ให้ค่า F-measure น้อยที่สุด จากชุดข้อมูลที่ 2 เว็บเพจข่าวกีฬา Yahoo รูปที่ 7 ที่ขนาดคุณลักษณะมากที่สุด ($\lambda \geq 0$) SVM ให้ค่าดีที่สุด คือ 100% ทุกกรณี C4.5 ให้ค่า 99.6% ทุกกรณี และ RBFNN ให้ค่า 92.95% โดยมีค่าเพิ่มขึ้นเป็น 100% เมื่อค่า $\lambda \geq 20$ ซึ่งพบว่าเมื่อลดจำนวนคุณลักษณะลงจะทำให้ตัวจำแนกประเภทสามารถทำงานได้ดียิ่งขึ้น

ตารางที่ 7 ค่าเฉลี่ย F-measure ชุดข้อมูลที่ 2 เว็บเพจข่าวเว็บไซต์ Yahoo

ความถี่เอกสาร (Document Frequency)	จำนวนคุณลักษณะ	ค่าเฉลี่ย F-measure (%)		
		SVM	C4.5	RBF
$\lambda \geq 0$	9,608	100.00	99.60	92.95
$\lambda \geq 5$	2,285	100.00	99.60	96.82
$\lambda \geq 10$	1,364	100.00	99.60	99.22
$\lambda \geq 15$	1,019	100.00	99.60	99.60
$\lambda \geq 20$	803	100.00	99.60	100.00
$\lambda \geq 25$	691	100.00	99.60	100.00
$\lambda \geq 30$	617	100.00	99.60	100.00
$\lambda \geq 35$	565	100.00	99.60	100.00
$\lambda \geq 40$	533	100.00	99.60	100.00
$\lambda \geq 45$	511	100.00	99.60	100.00
$\lambda \geq 50$	492	100.00	99.60	100.00



รูปที่ 6 ค่าเฉลี่ย F-measure ชุดข้อมูลที่ 1 WebKB



รูปที่ 7 ค่าเฉลี่ย F-measure ชุดข้อมูลที่ 2 เว็บเพจข่าวกีฬา เว็บไซต์ Yahoo

5. บทสรุป

บทความนี้นำเสนอการจำแนกประเภทเว็บเพจ WPC_DF_SVM โดยนำเอกสารเว็บเพจมาสกัดเอาเฉพาะส่วนของข้อความเพียงอย่างเดียว หารากศัพท์ของคำ กำจัดคำที่ไม่มีความจำเป็นด้วยคำหยุด “Stoplist” สร้างคุณลักษณะให้อยู่ในรูป Term Frequency และ Document Frequency ลดขนาดคุณลักษณะด้วยค่าความถี่เอกสารตามค่าที่กำหนด (λ) ด้วยจำนวนที่แตกต่างกัน แล้วทำดัชนี ผลการทดลองพบว่า การจำแนกประเภทโดยใช้ค่าความถี่เอกสารร่วมกับตัวจำแนกประเภท SVM ให้ผลการทดลองที่ดีที่สุด โดยสามารถลดขนาดคุณลักษณะในชุดข้อมูลที่ 1 จาก 12,722 จำนวน ($\lambda \geq 0$) ให้เหลือเพียง 88 จำนวน ($\lambda \geq 50$) ซึ่งยังได้ผลการทดลองค่า F-measure ที่สูง และในชุดข้อมูลที่ 2 ลดขนาดคุณลักษณะจาก 9,608 จำนวน ($\lambda \geq 0$) ให้เหลือเพียง 492 จำนวน ($\lambda \geq 50$) และยังคงให้ค่า F-measure ที่ถูกต้อง 100%

6. งานวิจัยที่ดำเนินการต่อไป

การนำคุณลักษณะอื่น ๆ ของเว็บเพจ เช่น URL Title และ HTML tag มาวิเคราะห์เป็นคุณลักษณะร่วมกับข้อความเพื่อเพิ่มประสิทธิภาพของการจำแนกประเภทเว็บเพจให้ดียิ่งขึ้น

7. กิตติกรรมประกาศ

งานวิจัยนี้ได้รับทุนสนับสนุนนักศึกษาระดับบัณฑิตศึกษาเป็นผู้ช่วยนักวิจัย (Research Assistant) ปีการศึกษา 2550 จากกองทุนคณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

เอกสารอ้างอิง

[1] S. R. Gunn, “Support Vector Machines for Classification and Regression,” University of Southampton, 1998.

[2] W. Xue, H. Bao, W. Huang, and Y. Lu, “Web Page Classification Based on SVM,” 6th World Congress on Intelligent Control and Automation, Dalian, China, pp. 6111-6114, June 21-23, 2006.

[3] C. Jujie, J. Rongbing, “Research of Web Classification Mining

Based on RBF Neural Network," 8th International Conference on Control, Automation, Robotics and Vision Kunming, China, pp. 1365-1367, December 6-9, 2004.

[4] V. Vidulin, M. Lustrek, M. Gams, "Training the Genre Classifier for Automatic Classification of Web Pages," ITI 2007 29th Internation Conference on Information Technology Interfaces, Cavtat, Croatia, pp. 93-98, June 25-28, 2007.

[5] R. C. Chen, C. H. Hsieh, "Web Page Classification Based On A Support Vector Machine Using A Weighted Vote Schema," Expert Systems with Applications, vol. 31, pp. 427-435, August, 2006.

[6] F. Sebastian, "Machine Learning in Automated Text Categorization," ACM Computing Surveys, vol. 34, pp. 1-47, March, 2002.

[7] Text Extractor from Web page, Freeware, http://www.nirsoft.net/utils/htmlastext.html.

[8] M. F. Porter, "Algorithm for Suffix Stripping," pp.313–316, 1997.

[9] T. Joachims., "A Probabilistic Analysis of The Rocchio Algorithm with TFIDF for Text Categorization," Proceedings of ICML-97 International Conference On Machine Learning, 1997.

[10] C. Cortes and V. Vapnik, "Support Vector Networks," Machine Learning, vol. 30, no.3, 1995.

[11] J. R. Quinla, "C.45: Programs for Machine Learning," San Mateo, CA: Morgan Kaufmann, 1993.

[12] W. Xuhong, X. Jinhua, "RBF Neural Network Based Predictive Control of Active Power Filter," TENCON 2004. 2004 IEEE Region 10 Conference, vol 4, pp.109-112, November 21-24, 2004.

[13] D. Castro, J. Fernando, "An Evolutionary Immune Network for Data Clustering," In Proc. of the SBRN, vol.11, pp. 84-89, 2000.

[14] WebKB (1999). Available: http://www.cs.cmu.edu/~WebKB/ [26/June/2008].

[15] Yahoo Web Pages, 2008, Available: http://www.yahoo.com [2/July/2008].

[16] Data Mining Software in Java, The University of Waikato, http://www.cs.waikato.ac.nz/ml/weka/.

ประวัติผู้เขียนบทความ

พรพล ธรรมรงค์รัตน์ นักศึกษาปริญญาโท ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ งานวิจัยที่สนใจได้แก่ Web Intelligent, Neural Networks, Data Mining

ดร.วิภาดา เวทย์ประสิทธิ์ อาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ งานวิจัยที่สนใจได้แก่ Neural Networks, Artificial Intelligent, Knowledge Management, Data Mining

ดร.ลัดดา ปรีชาวีรกุล อาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ งานวิจัยที่สนใจได้แก่ Internet Computing, Image and Video Retrieval, Information Security

ภาคผนวก ค

ผลงานตีพิมพ์

เรื่อง A Novel Voting Algorithm of Multi-Class SVM for Web Page Classification

Conference The 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009)

สถานที่ Beijing China

วันที่ 8-11 August 2009

A Novel Voting Algorithm of Multi-Class SVM for Web Page Classification

Pornpon Thamrongrat
Artificial Intelligence Research Laboratory
Computer Science Department
Prince of Songkla University, Thailand
pthamrongrat@hotmail.com

Ladda Preechaveerakul
*i*STAR Research Laboratory
Computer Science Department
Prince of Songkla University, Thailand
ladda.p@psu.ac.th

Wiphada Wettayaprasit
Artificial Intelligence Research Laboratory
Computer Science Department
Prince of Songkla University, Thailand
wwettayaprasit@yahoo.com

***Abstract*—The increasing numbers of web pages on the cyber world result to the less effectiveness of document retrieval that matches the need of users. The classification of web pages is one of the solutions to solve this problem. This paper proposes VAMSVM_WPC model which is a novel voting algorithm for classifying the web pages, which uses a multi-class SVM method. First, feature is generated from text and title, and then reduces the number of features by two feature selection techniques. Use these two types of features to give input to multi-class SVM. Finally, on the output of SVM, a voting algorithm is used to determine the category of the web pages. Results on CMU benchmark dataset show that using text and title feature with 1vsAll_Voting Algorithm gives the highest F-measure value.**

Keywords-support vector machine; web page classification voting; feature selection.

I. INTRODUCTION

The rapid development of information technology contributes to the developing of data, information, and knowledge into the form of web pages for the purposes of facilitating and speeding up of searching for users. In addition, when the number of web pages is increased, the efficiency does not match the need of users. One of solutions to answer this problem is web page classification.

There are web page classifications using different kinds of classifiers. For example, Weimin and Aixin [1] used body, title, heading and meta text as feature by using SVM and Naive Bayesian classifier. The result shows that combination of these features with SVM classifier gives higher efficiency for web page classification system. Xin Jin *et al.* [2] used ReliefF, Information Gain, Gain ratio and Chi Square as feature selection technique for improving the web page classification performance. Rung-Ching and Chung-Hsun [3] proposed a web page classification method by using two types of features as inputs to SVM classification. The output of two SVM is used as inputs of voting schema to determine the category of the web page. The voting improves the performance when compares with the traditional methods. Rung Fang *et al.* [4] proposed a web page classification by using five classification methods. The output of these SVMs is used as inputs of voting method and picks the class with the most votes as the final classification result. This method improves the performance when compared with the individual classifiers.

This paper contains of the following sections. Section II is the Support Vector Machine. Section III is a novel Voting Algorithm of Multi-Class SVM for Web Page Classification (VAMSVM_WPC) model. Section IV is experimental results and discussion. Section V is conclusion.

II. SUPPORT VECTOR MACHINE

A. Binary SVM Classification

Support Vector Machine [5] is used to find a hyper plane to do binary division. Let (x_i, y_i), …, (x_n, y_n) be training samples, n be the input dimension and y belongs to the class of +1 or -1. For a linearly problem, a hyper plane is divided into two categories by equation (1).

$$(w \cdot x) + b = 0 \tag{1}$$

where w is weight and b is bias. The classification can be calculated in equation (2).

$$y = \begin{cases} +1 & if\ (w \cdot x) + b > 0 \\ -1 & if\ (w \cdot x) + b < 0 \end{cases} \tag{2}$$

The example with the desired output y = +1 is called positive example and the example with the desired output y = -1 is called negative example.

B. Multi-class Classification based on binary SVM

1) One-vs-One (1vs1). A SVM for a pair of classes (k,m) is constructed using training examples belonging to the two classes only. The desired output y for a training example x can be calculated in equaltion (3).

$$y = \begin{cases} +1 & if\ \ c = k \\ -1 & if\ \ c = m \end{cases} \tag{3}$$

The Maxwins strategy is used to determine the class of a test pattern x in this approach. In this strategy, a majority voting scheme is used. If the value of the discriminant function of the SVM for a pair of classes (k,m) is positive, then class k wins a vote. Otherwise, class m wins a vote. Outputs of SVMs are used to determine the number of votes won by each class. The class with maximum number of votes is assigned to the test pattern [6].

2) One-vs-All (1vsAll). A test pattern x is classified by using the winner-takes-all decision strategy, the class with the maximum value of the discriminant function is assigned to it. All the training examples are used in constructing an SVM for a class. The SVM for class k is constructed using the set of training examples and their desired output,

(x_i, y_i). The desired output y for a training example x can be calculated in equation (4) [7].

$$y = \begin{cases} +1 & if \quad c = k \\ -1 & if \quad c \neq k \end{cases} \tag{4}$$

III. A NOVEL VOTING ALGORITHM OF MULTI-CLASS SVM FOR WEB PAGE CLASSIFICATION MODEL

A novel Voting Algorithm of Multi-class SVM for Web Page Classification (VAMSVM_WPC) model composes of four steps as shows in Fig. 1. Step 1 is web page preprocessing. Step 2 is feature generation. Step 3 is feature selection. Step 4 is SVM classification and new voting technique.

A. Step 1: Web Page Preprocesing

This step composed of three processes as follows.

1.1) Extracting text and title from web pages, the result will be the content and title of web page document without the parts of html, meta tag, picture, or multimedia.

1.2) Stemming process, after receiving text and title from the web pages, take those words for Porter stemming algorithm [8] by modifying suffix in English such as words "Run" and "Running" will be modified to the same stem that is "run".

1.3) Stopping Process: Stopping is one of the important processes because some words are not needed for the analysis. Then such words should be eliminated to leave only significant words. These words are called "Stoplist" such as words "a", "been", "ago", and etc.

B. Step 2: Feature Generation

2.1) Text Feature Generation.

2.1.1 Take words from text to generate in the term document matrix as shows in Table I.

2.1.2 Giving word weight value in the term document matrix by the TF-IDF method [9] when TF (Term Frequency) is the frequency of each word and IDF (Inverse Document Frequency) is the inverse part of document frequency value of each word. w_{mn} is the TF-IDF value of web page *m* of *word n*.

2.1.3 Select terms that have the document frequency greater than the threshold value [10].

2.2) Title Feature Generation.

2.2.1 Take words from title to generate in the term document matrix as shows in Table I.

2.2.2 Giving word weight value by TF-IDF method.

2.3) Combining both text features and title features.

C. Step 3: Feature Selection

After feature generation step, the feature is created into two features that are Leftfeature and Rightfeature.

3.1) Select two feature selection techniques from ReliefF (RF) [11], Information Gain (IG) [12], Chi Square (CS) [13], and Gain Ratio (GR) [12].

3.2) Assign one technique to Leftfeature and the other technique to Rightfeature.

Step 1: Web Page Preprocessing
1.1 Extract text and title from web page. 1.2 Porter Stemming Algorithm. 1.3 Eliminate Stoplist words.
Step 2: Feature Generation
2.1 Text Feature Generation. 2.1.1 Generate term document matrix from text. 2.1.2 Weighting words by TF-IDF method. 2.1.3 Select terms that have the document frequency greater than the threshold value. 2.2 Title Feature Generation. 2.2.1 Generate term document matrix from title. 2.2.2 Weighting words by TF-IDF method. 2.3 Combining both text features and title features.
Step 3: Feature Selection
3.1 Select two feature selection techniques from the follows. • ReliefF feature selection. • Information Gain feature selection. • Chi Square feature selection. • Gain Ratio feature selection. 3.2 Assign one technique to Leftfeature and the other technique to Rightfeature.
Step 4: SVM classification and new voting technique
4.1 Choose one of the classification strategy for Voting algorithm. (a) One-vs-One (1vs1). • Classification both LeftFeature and RightFeature using 1vs1 SVM classification. • Recive predicted class using 1vs1_Voting algorithm as shows in Fig. 2. (b) One-vs-All (1vsAll). • Classification both LeftFeature and RightFeature using 1vsAll SVM classification. • Recive predicted class using 1vsAll_Voting algorithm as shows in Fig. 3. 4.2 Evaluate the classification results from F-measure technique.

Figure 1. VAMSVM_WPC Model.

TABLE I. TERM DOCUMENT MATRIX

Web Page	**Features**			
	***Word*$_1$**	***Word*$_2$**	**…**	***Word*$_n$**
Web$_1$	W_{11}	W_{12}	…	W_{1n}
Web$_2$	W_{21}	W_{22}	…	W_{2n}
…	…	…	…	…
Web$_m$	W_{m1}	W_{m2}	…	W_{mn}

D. Step 4: SVM classification and voting technique

4.1) Both Leftfeature and Rightfeature from step 3 are sent to be classified by multi-class SVM that can choose 1vs1 or 1vsAll strategy. After classification, the output of two classifications is sent to the input of voting algorithm for determining the category of the web pages.

(a) One-vs-One (1vs1)

• SVM classification, both Leftfeature and Rightfeature are classified by 1vs1 SVM method and received Leftclassifier and Rightclassifier.

• The 1vs1_Voting algorithm will consider in two parts. Part 1 will consider the number of vote values from predicted class classification results both left side and right side. Part 2 will consider the summation weight value of the discriminant values of the SVM classification results. The voting algorithm shows in Fig. 2 (a). The example of 1vs1 SVM classification outputs shows in Fig. 2 (b). In Fig. 2 (c) from the 1vs1_Voting algorithm, because the total numvote value (Part 1) for both C1 and C2 is equaled to 3, then we will consider the sum weight value (Part 2) of C1 = 1.78 is greater than C2 = 1.67. Therefore, the result will be C1.

(b) One-vs-All (1vsAll)

- SVM classification, both Leftfeature and Rightfeature are sent to be classified by 1vsAll SVM method and received Leftclassifier and Rightclassifier.
- The 1vsAll_Voting algorithm will consider in three parts. Part 1 will consider the rank value of weight from predicted class classification results both left side and right side. Part 2 will consider the weight threshold value. Part 3 will consider the weight value of the discriminant value of the SVM classification results. The 1vsAll_Voting algorithm shows in Fig. 3 (a). The example of 1vsAll SVM classification outputs show in Fig. 3 (b). In Fig. 3 (c) from the algorithm, the total rank value (Part 1) for both C1 and C2 is equaled to 5. Next step, we will consider the weight threshold value (Part 2), both C1 and C2 is equaled to 0, then we will consider the weight value (Part 3) of C2 = 5.9 is greater than C1 = -1.24. Therefore, the result will be C2.

Let *i* be the identical number of class *i*, where *i* = 1, …, *n* and *n* be the number of classes.
Let *numvote_left* and *numvote_right* be the number of vote on the Leftclassifier and Rightclassifier, respectively.
Let *w_left* and *w_right* be the number of summation of absolute discriminant value on the Leftclassifier and Rightclassifier, respectively.
Input: *numvote_left*, *numvote_right, w_left*, *w_right*
Ouput: *i.name* // a name of class *i*

1: **For** each class *i* **do** // ***** Part 1: Numvote value *****
2: $total\_numvote_i = numvote\_left_i + numvote\_right_i$
3: **End for**
4: *total_numvote_max* = Max (all of $total\_numvote_i$)
5: **If** *total_numvote_max* have a single class **then**
6: **Return** class *i.name* where $total\_numvote_i == total\_numvote\_max$
7: **Else** //***** Part 2: Sum weight value *****
8: **For** each class *i* where $total\_numvote_i == total\_numvote\_max$ **do**
9: $total\_w_i = w\_left_i + w\_right_i$
10: **End for**
11: *total_w_max* = Max (all of $total\_w_i$)
12: **If** *total_w_max* have a single class **then**
13: **Return** class *i*.name where $total\_w_i == total\_w\_max$
14: **Else**
15: **Return** one of class *i.name* where $total\_w_i == total\_w\_max$
16: **End if** // *total_w_max*
17: **End if** // *total_numvote_max*

(a) 1vs1_Voting algorithm.

1vs1 SVM classification					
Classifier	**Output value**	**Class**			**Predicted Class**
		C1	**C2**	**C3**	
Leftclassifier	number of vote	**2**	1	0	C1
	∑|discriminant value|	0.94	0.65	0	
Rightclassifier	number of vote	1	**2**	0	C2
	∑|discriminant value|	0.84	1.02	0	

(b) Example of 1vs1 SVM classification.

1vs1_Voting algorithm					
Part	**Variable**	**Class**			**Voted Class**
		C1,*i*=1	**C2,*i*=2**	**C3,*i*=3**	
Part 1: Numvote value	*numvote_left*	2	1	0	
	numvote_right	1	2	0	
	total_numvote	**3**	**3**	0	C1,C2
Part 2: Sum weight value	*w_left*	0.94	0.65		
	w_right	0.84	1.02		
	total_w	**1.78**	1.67		**C1**

(c) Output example of 1vs1_Voting algorithm.

Figure 2. One-vs-One (1vs1).

4.2) Evaluate the classification results from F-measure technique.

Let *i* be the identical number of class *i*, where *i* = 1, …, *n* and *n* be the number of classes.
Let *w_left* and *w_right* be the number of discriminant value on the Leftclassifier and Rightclassifier, respectively.
Input: *w_left*, *w_right*
Ouput: *i.name* // a name of class *i*

1: For each class *i*, rank *w_left* in descending order, then assign it to $rank\_left_i$ // ***** Part 1: Rank value *****
2: For each class *i*, rank *w_right* in descending order, then assign it to $rank\_right_i$
3: **For** each class *i* **do**
4: $total\_rank_i = rank\_left_i + rank\_right_i$
5: **End for**
6: *total_rank_max* = Max (all of $total\_rank_i$)
7: **If** *total_rank_max* have a single class **then**
8: **Return** class *i*.name where $total\_rank_i == total\_rank\_max$
9: **Else** // ***** Part 2: Weight threshold value *****
10: **For** each class *i* where $total\_rank_i == total\_rank\_max$ **do**
11: **If** $w\_left_i >= 0$ **then**
12: $E\_left_i = 1$
13: **Else**
14: $E\_left_i = -1$
15: **End if**
16: **If** $w\_right_i >= 0$ **then**
17: $E\_right_i = 1$
18: **Else**
19: $E\_right_i = -1$
20: **End if**
21: $total\_E_i = E\_left_i + E\_right_i$
22: **End for**
23: *total_E_max* = Max (all of $total\_E_i$)
24: **If** *total_E_max* have a single class **then**
25: **Return** class *i*.name where $total\_E_i == total\_E\_max$
26: **Else** // ***** Part 3: Weight value *****
27: **For** each class *i* where $total\_E_i == total\_E\_max$ **do**
28: $total\_w_i = w\_left_i + w\_right_i$
29: **End for**
30: *total_w_max* = Max (all of $total\_w_i$)
31: **If** *total_w_max* have a single class **then**
32: **Return** class *i.name* where $total\_w_i == total\_w\_max$
33: **Else**
34: **Return** one of class *i.name* where $total\_w_i == total\_w\_max$
35: **End if** // *total_w_max*
36: **End if** // *total_E_max*
37: **End if** // *total_rank_max*

(a) 1vsAll_Voting algorithm.

1vsAll SVM classification					
Classifier	**Output value**	**Class**			**Predicted Class**
		C1	**C2**	**C3**	
Leftclassifier	discriminant value	**0.65**	-0.94	-1.90	C1
Rightclassifier	discriminant value	-1.89	**6.84**	-4.19	C2

(b) Example of 1vsAll SVM classification.

1vsAll_Voting algorithm					
Part	**Variable**	**Class**			**Voted Class**
		C1,*i*=1	**C2,*i*=2**	**C3,*i*=3**	
Part 1: Rank value	*rank_left*	3	2	1	
	rank_right	2	3	1	
	total_rank	**5**	**5**	2	C1,C2
Part 2: Weight threshold value	*E_left*	1	-1		
	E_right	-1	1		
	total_E	**0**	**0**		C1,C2
Part 3: Weight value	*w_left*	0.65	-0.94		
	w_right	-1.89	6.84		
	total_w	-1.24	**5.9**		**C2**

(c) Output example of 1vsAll_Voting algorithm.

Figure 3. One-vs-All (1vsAll).

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The experiment uses benchmark dataset of web page document from CMU Industry Sector [14] which is a collection of web pages from various economic sectors. We use 630 web pages from 3,533 html web pages with 7 classes that are energy, financial, healthcare, materials, technology, transportation, and utilities. Each class has 90 web pages. After web page preprocessing step, the number of text features is 22,938 and the number of title features is 716. After document frequency threshold, the number of text features is 1,033, and the number of text and title features is 1,749.

We use four different experimental methods (A, B, C, and D) to evaluate the performance of 10-fold cross validations as shows in Table II. Choosing feature selection technique with RF, IG, CS and GR. The numbers of feature after feature selection are 100, 150, 200, 250 300, 350, 400, and all. The results of model A, B, C, and D show as Table III, IV, V, and VI, respectively. There are 4 issues to consider that are 1) choosing feature selection technique, 2) the multi-class SVM strategy, 3) Voting algorithm compare with individual classification, and 4) Comparison of the Proposed VAMSVM_WPC.

TABLE II. EXPERIMENTAL DESIGN

Exp.	Experiment Name	Feature		SVM and Voting	
		Text Only	*Text&Title*	*1vs1*	*1vsAll*
A	TO_1vs1	✓	×	✓	×
B	TO_1vsAll	✓	×	×	✓
C	TT_1vs1	×	✓	✓	×
D	TT_1vsAll	×	✓	×	✓

TABLE III. THE PERCENTAGE OF F-MEASURE VALUE OF A: TO_1VS1

No. Features	1vs1 SVM Classification				1vs1_Voting		
	RF	*IG*	*CS*	*GR*	*RF&IG*	*RF&CS*	*RF&GR*
100	89.10	86.50	87.79	79.61	91.15	92.73	91.91
150	89.41	84.19	82.19	83.50	90.30	90.35	90.50
200	88.92	85.75	84.81	82.41	89.22	89.08	87.69
250	87.31	85.13	85.14	85.13	87.81	87.63	87.81
300	87.49	84.67	84.67	84.67	87.15	87.15	87.15
350	86.16	83.41	83.41	83.41	85.83	85.83	85.83
400	86.80	81.59	81.59	81.59	84.61	84.61	84.61
all	80.16	80.16	80.16	80.16	80.16	80.16	80.16

TABLE IV. THE PERCENTAGE OF F-MEASURE VALUE OF B: TO_1VSALL

No. Features	1vsAll SVM Classification				1vsAll_Voting		
	RF	*IG*	*CS*	*GR*	*RF&IG*	*RF&CS*	*RF&GR*
100	91.44	86.50	87.50	80.14	91.44	92.37	89.34
150	91.28	85.43	83.49	83.13	91.28	91.89	90.48
200	91.13	87.18	88.45	84.81	92.72	92.72	92.87
250	90.82	88.29	87.02	87.81	91.46	91.63	91.61
300	91.28	88.15	88.15	88.15	91.80	91.80	91.80
350	90.67	86.56	86.56	86.56	91.78	91.78	91.78
400	89.89	84.15	84.15	84.15	89.91	89.91	89.91
all	86.23	86.23	86.23	86.23	86.23	86.23	86.23

TABLE V. THE PERCENTAGE OF F-MEASURE VALUE OF C: TT_1VS1

No. Features	1vs1 SVM Classification				1vs1_Voting		
	RF	*IG*	*CS*	*GR*	*RF&IG*	*RF&CS*	*RF&GR*
100	90.35	87.00	87.46	77.09	91.63	92.73	90.52
150	89.70	84.35	87.27	81.32	90.22	91.31	91.30
200	89.21	84.76	84.76	81.07	90.82	91.16	90.62
250	89.06	86.24	85.45	83.58	90.36	90.23	90.05
300	88.29	85.29	85.12	83.60	88.13	87.66	88.99
350	87.99	84.97	85.72	84.78	88.47	88.31	88.47
400	88.46	84.05	83.53	84.47	86.74	87.96	87.35
all	81.57	81.57	81.57	81.57	81.57	81.57	81.57

TABLE VI. THE PERCENTAGE OF F-MEASURE VALUE OF D: TT_1VSALL

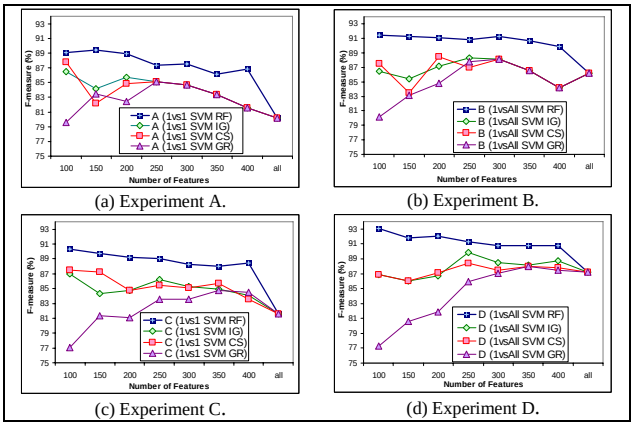
No. Features	1vsAll SVM Classification				1vsAll_Voting		
	RF	*IG*	*CS*	*GR*	*RF&IG*	*RF&CS*	*RF&GR*
100	93.04	86.85	86.87	77.32	93.35	93.01	91.13
150	91.77	86.06	86.04	80.61	92.56	92.68	90.01
200	92.07	86.71	87.15	81.84	92.87	93.17	92.37
250	91.30	89.87	88.43	85.91	**93.98**	**93.18**	**92.23**
300	90.78	88.47	87.51	87.06	93.18	93.18	93.19
350	90.81	88.12	87.97	87.97	93.03	92.41	92.40
400	90.79	88.77	87.84	87.51	92.09	91.78	92.25
all	87.22	87.22	87.22	87.22	87.22	87.22	87.22

A. *Issue of Choosing Feature Selection Technique*

RF feature selection technique gives highest F-measure value when compares with IG, CS, and GR. For example, at 100 features, 1) on exp. A, the study shows in Fig. 4 (a), RF gives the highest F-measure value at 89.10% while IG is 86.50%, CS is 87.79%, and GR is 79.61%, 2) on exp. B, the study shows in Fig. 4 (b), RF gives the highest F-measure value at 91.44%, 3) on exp. C, the study as shows in Fig. 4 (c), RF gives the highest F-measure value at 90.35% and 4) on exp. D, the study as shows in Fig. 4 (d), RF gives the highest F-measure value at 93.04%.

B. *Issue of the Multi-Class SVM strategy*

1vsAll SVM strategy gives higher F-measure value than 1vs1 SVM strategy. The study shows in Fig. 5. For example, at 100 features with RF, 1) on text only, the F-measure value of 1vsAll SVM strategy (exp. B) gives higher F-measure value at 91.44% when 1vs1 SVM strategy (exp. A) is only 89.10%, 2) on text and title, the F-measure value of 1vsAll strategy (exp. D) gives higher F-measure value at 93.04% when 1vs1 strategy (exp. C) is only 90.35%.



(a) Experiment A. (b) Experiment B.
(c) Experiment C. (d) Experiment D.

Figure 4. The F-measure value of the SVM classification with RF, IG, CS, and GR feature selction techniques.

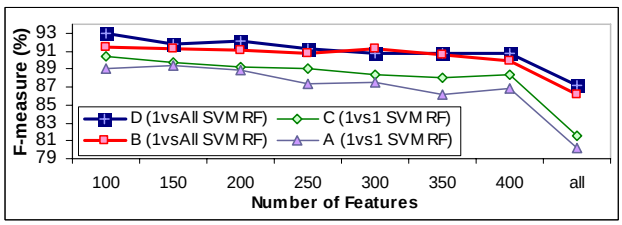
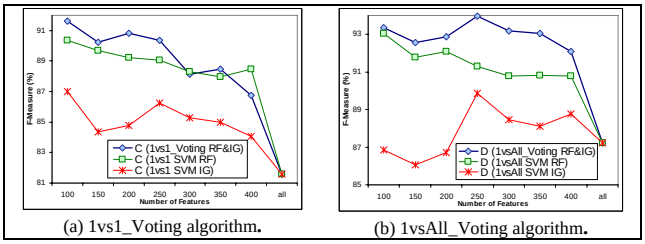


Figure 5. The F-measure value of the SVM classification with ReliefF feature selection technique.



(a) 1vs1_Voting algorithm. (b) 1vsAll_Voting algorithm.

Figure 6. The F-measure value of the voting algorithm of RF and IG.

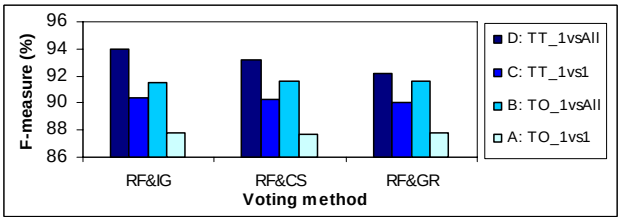


Figure 7. The F-measure value of the voting algorithm at 250 features.

C. Issue of Voting algorithm compare with individual classification

Since RF gave the highest performance among other feature selection techniques. Therefore, we use RF for the Leftclassifier and other techniques to Rightclassifier.

Performance of 1vs1_Voting algorithm, 1vs1_Voting algorithm gives higher F-measure value than individual classification (1vs1 SVM classification). For example, at 250 features, 1) on exp. C, the study shows in Fig. 6 (a), 1vs1_Voting algorithm of RF and IG gives higher F-measue value at 90.36% when 1vs1 SVM classification with RF is 89.06% and IG is 86.24%, 2) on exp. A, 1vs1_Voting algorithm of RF and IG gives higher F-measue value at 87.81% when 1vs1 SVM classification with RF is 87.31% and IG is 85.13%.

Performance of 1vsAll_Voting algorithm, 1vsAll_Voting algorithm gives higher F-measure value than individual classification (1vsAll SVM classification). For example at 250 features, 1) on exp. D, the study as shows in Fig. 6 (b), 1vsAll_Voting algorithm of RF and IG gives higher F-measue at 93.98% when 1vsAll SVM classification with RF is 91.30% and IG is 89.87%, 2) on exp. B, 1vsAll_Voting algorithm of RF and IG gives higher F-measue at 91.46% when 1vsAll SVM classification with RF is 90.82% and IG is 88.29%.

D. Issue of Comparison of the Proposed VAMSVM_WPC

Using text and title features with 1vsAll_Voting algorithm (exp. D) gives the highest F-measure value when compares with using text only with 1vs1_Voting algorithm (exp. A), using text only with 1vsAll_Voting algorithm (exp. B), using text and title features with 1vs1_Voting algorithm (exp. C). The study at 250 features shows in Fig. 7. For example, 1) on the voting algorithm of RF and IG, the F-measure value of exp. D is 93.98% while exp. A is 87.81%, exp. B is 91.46%, and exp. C is 90.36%, 2) on the voting algorithm of RF and CS, the F-measure value of exp. D is 93.18% while exp. A is 87.63%, exp. B is 91.63%, and exp. C is 90.23%, 3) on the voting algorithm of RF and GR. The F-measure value of exp. D is 92.23% while exp. A is 87.81%, exp. B is 91.61%, and exp. C is 90.05%.

V. CONCLUSION

This paper presents voting algorithm from multi-class SVM for web page classification. The experimental results concluded as following. 1) Using ReliefF feature selection technique gives the highest F-measure value when compares with Information Gain, Chi Square, and Gain Ratio. 2) On multi-class SVM, 1vsAll SVM strategy gives higher F-measure value than 1vs1 SVM strategy 3) voting algorithm gives higher F-measure value than the individual classification. 4) Using text and title features with 1vsAll_Voting algorithm gives the highest F-measure value. Hence, the VAMSVM_WPC can improve the efficiency of the traditional web page classification methods.

ACKNOWLEDMEDGMENT

This paper receives financial support for graduate studies from Research Fund from the Faculty of Science at Prince of Songkla University, Hatyai Campus, Thailand for academic year 2007-2008.

REFERENCES

[1] W. Xue, H. Bao, W. Huan, and Y. Lu, “Web Page Classification Based on SVM,” 6th World Congress on Intelligent Control and Automation, Dalian, China, pp. 6111-6114, 2006.

[2] J. Xin, L. Rongyan, S. Xian, and B. Rongfang, “Automatic Web Pages Categorization with ReliefF and Hidden Naive Bayes,” Proceedings of the 2007 ACM symposium on Applied computing, Seoul, Korea, pp. 617-621, 2007.

[3] R. Chen, C. Hsieh, and H. C., “Web Page Classification Based On A Support Vector Machine Using A Weighted Vote Schema,” Expert Systems with Applications, vol. 31, pp. 427-435, 2006.

[4] F. Rui, M. Alexander, and T. Babis, “A Voting Method for the classification of Web Pages,” IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, pp. 610-613, 2006.

[5] S. R. Gunn, “Support Vector Machines for Classification and Regression,” University of Southampton, 1998.

[6] S. Knerr, L. Personnaz, and G. Dreyfus, “Nurocosingle-layer learning revisited: A stepwise procedure for building and training a neural network,” Springer, 1990.

[7] V.Vapnik, “Statistical learning theory,” Wiley, NewYork, 1998.

[8] Porter stemming algorithm: Porter, M., F.: Algorithm for Suffix Stripping. pp. 313-316, 1997.

[9] T. Joachims, “A Probabilistic Analysis of The Rocchio Algorithm with TFIDF for Text Categorization,” Proceedings of ICML-97 International Conference On Machine Learning, 1997.

[10] Stemming: http://www.comp.lancs.ac.uk/computing/research/stemming/general/ (Access 2006).

[11] M. Robnik-Sikonja, and I. Kononenko, “Theoretical and Empirical Analysis of ReliefF and RReliefF,” Machine Learning 53(1-2):23.69, 2003.

[12] I.H. Witten, and E. Frank, “Data Mining Practical Machine Learning Tools and Technique,” San Francisco, Morgan Kaufman, 2005.

[13] Y. Yang, and J. O. Pedersen, “A Comparative Study on Feature Selection in Text Categorization,” Fourteenth International Conference on Machine Learning (ICML’97), pp. 412-420, 1997.

[14] CMU Industry sector dataset: http://www.cs.cmu.edu/~TextLearning/datasets.html [26/June/2008].

ประวัติผู้เขียน

ชื่อ สกุล นายพรพล ธรรมรงค์รัตน์

รหัสประจำตัวนักศึกษา 5010220080

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วท.บ. (วิทยาการคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2548

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนสนับสนุนนักศึกษาระดับบัณฑิตศึกษา เป็นผู้ช่วยนักวิจัย (Research Assistant) จากกองทุนวิจัยคณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2550

การตีพิมพ์เผยแพร่ผลงาน

1. พรพล ธรรมรงค์รัตน์ ลัดดา ปรีชาวีรกุล และ วิภาดา เวทย์ประสิทธิ์. 2551. การจำแนกประเภทเว็บเพจโดยใช้ค่าความถี่เอกสารและซัพพอร์ตเวกเตอร์แมชชีน. การประชุมวิชาการวิทยาการคอมพิวเตอร์และวิศวกรรมคอมพิวเตอร์แห่งชาติ ครั้งที่ 12 (NCSEC 2008). ชลบุรี, ประเทศไทย, 20-21 พฤศจิกายน 2551. หน้า 498-504.
2. Thamrongrat, P., Preechaveerakul, L., and Wettayaprasit, W. 2009. A Novel Voting Algorithm of Multi-Class SVM for Web Page Classification. In Proceedings The 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009). Beijing, China, August 8-11, 2009.