ซัพพอร์ตเวกเตอร์แมชชีนแบบหลายประเภทโดยการแตกครึ่งแบบสมดุล

นาย ณรงค์ บุญสิริสัมพันธ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2546

ISBN 974-17-5634-8

MULTICLASS SUPPORT VECTOR MACHINES BY BALANCED DICHOTOMIZATION

Mr. Narong Boonsirisumpun

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2003

ISBN 974-17-5634-8

หัวข้อวิทยานิพนธ์	ซัพพอร์ตเวกเตอร์แมชชีนแบบหลายประเภทโดยการแตกครึ่งแบบสมดุล
โดย	นายณรงค์ บุญสิริสัมพันธ์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

………………………………………….. คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

…………………………………………… ประธานกรรมการ

(รองศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

………………………………………….. อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

…………………………………………… กรรมการ

(อาจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์)

…………………………………………… กรรมการ

(อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

ณรงค์ บุญสิริสัมพันธ์ : ซัพพอร์ตเวกเตอร์แมชชีนแบบหลายประเภทโดยการแตกครึ่งแบบสมดุล. (MULTICLASS SUPPORT VECTOR MACHINES BY BALANCED DICHOTOMIZATION) อ. ที่ปรึกษา : ผศ. ดร. บุญเสริม กิจศิริกุล, 67 หน้า. ISBN 974-17-5634-8.

การสร้างซัพพอร์ตเวกเตอร์แมชชีนแบบหลายประเภทเป็นงานวิจัยที่ท้าทาย งานวิจัยฉบับนี้ได้นำเสนอวิธีการใหม่ เรียกว่า วิธีการแตกครึ่งแบบสมดุล เพื่อขยายความสามารถของซัพพอร์ตเวกเตอร์แมชชีนในการแก้ปัญหาการจำแนกข้อมูลแบบหลายประเภท วิธีนี้พัฒนามาจากวิธีการจำแนกแบบหนึ่งต่อหนึ่ง โดยเริ่มต้นที่การสร้างตัวจำแนกแบบหนึ่งต่อหนึ่งสำหรับแต่ละคู่ของประเภทข้อมูล แล้วทำการค้นหาระนาบของตัวจำแนกที่อยู่ในตำแหน่งสมดุลที่สุดเมื่อเทียบกับตำแหน่งของข้อมูลทั้งหมดมาใช้ในการจำแนกข้อมูล

วิธีการแตกครึ่งแบบสมดุลนี้จะสามารถตัดประเภทข้อมูลที่ไม่ถูกต้องออกไปได้สูงสุดครั้งละครึ่งหนึ่งของจำนวนประเภทข้อมูลทั้งหมด ซึ่งเป็นจำนวนที่สูงกว่าวิธีการจำแนกแบบหนึ่งต่อหนึ่งแบบอื่น ๆ เช่น วิธีดีดีเอจี วิธีเอดีเอจีและวิธีอาร์เอดีเอจี ที่สามารถตัดประเภทข้อมูลที่ไม่ถูกต้องออกไปได้เพียงครั้งละ 1 ประเภท นอกจากนี้ งานวิจัยยังได้นำเสนอจำนวนครั้งของการจำแนกจากผลการทดลองจริงเปรียบเทียบกับค่าที่คาดหวังไว้คือ $\lceil \log_2 K \rceil$ ครั้ง สำหรับปัญหาการจำแนกประเภทข้อมูล k ประเภท

ภาควิชา.........วิศวกรรมคอมพิวเตอร์......... ลายมือชื่อนิสิต..
สาขาวิชา......วิทยาศาสตร์คอมพิวเตอร์...... ลายมือชื่ออาจารย์ที่ปรึกษา......................................
ปีการศึกษา….2546…… ลายมือชื่ออาจารย์ทีปรึกษาร่วม.................................

4570299421 : MAJOR COMPUTER SCIENCE

KEY WORD: MULTICLASS SUPPORT VECTOR MACHINES / BALANCED DICHOTOMIZATION

NARONG BOONSIRISUMPUN : MULTICLASS SUPPORT VECTOR MACHINES BY BALANCED DICHOTOMIZATION. THESIS ADVISOR : BOONSERM KIJSIRIKUL, Ph.D, 67 pp. ISBN 974-17-5634-8.

Constructing multiclass Support Vector Machines (SVMs) is a challenging research problem. This paper presents a novel method, called Balanced Dichotomization to extend SVMs for multiclass problems. Balanced Dichotomization is considered as an one-against-one method. Once the system has constructed all binary SVMs, one for each pair of classes, it searches for the hyperplane at the optimal balanced position among all data classes in consideration, called *balanced dichotomization* classifier.

Using a balanced dichotomization classifier can remove half of the candidate classes during an evaluation for the correct class, that is a higher number of elimination compared to other methods, such as the DDAG, the ADAG and the RADAG, which eliminate only one class using an ordinary binary classifier. Results show that the proposed method maintains accuracy comparable to Max Wins, the RADAG and the ADAG. We also empirically evaluated the number of decisions to derive an answer compared to the expected value of $\lceil \log_2 K \rceil$ times for an K-class problem.

Department........Computer Engineering.....Student's signature..

Field of study.........Computer Science........Advisor's

Academic year ……2003......... Co-advisor's signature..

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างสูงจาก ผู้ช่วยศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้ความรู้ คำแนะนำ ข้อคิดเห็นและข้อเสนอแนะต่างๆ ที่เป็นประโยชน์แก่งานวิจัยชิ้นนี้ และขอขอบคุณ อาจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์ ที่ให้คำแนะนำในการเขียนรูปเล่มวิทยานิพนธ์รวมทั้งต้นฉบับภาษาอังกฤษ รวมทั้งขอขอบคุณรองศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา และ อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ ที่สละเวลาให้คำแนะนำ ตรวจและแก้ไขวิทยานิพนธ์ฉบับนี้

นอกจากนี้ ขอขอบคุณ คุณนพวรรณ เจ้าหน้าที่ธุรการภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความช่วยเหลือและแนะนำในเรื่องการส่งเอกสารต่างๆ และขอขอบคุณเพื่อนๆ ภาควิชาวิศวกรรมคอมพิวเตอร์และเพื่อนสมาชิกห้องปฏิบัติการอัจฉริยะภาพเครื่องกลและการค้นพบความรู้ ที่คอยให้ความช่วยเหลือ

สุดท้ายนี้ ขอกราบขอบพระคุณบิดา มารดา ครอบครัวและญาติพี่น้องที่คอยสนับสนุนและให้ความช่วยเหลือผู้วิจัยมาโดยตลอดจนทำงานวิจัยชิ้นนี้ให้สำเร็จลุล่วงไปได้ด้วยดี

สารบัญ

หน้า

สารบัญ (ต่อ)

สารบัญตาราง

หน้า

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ซัพพอร์ตเวกเตอร์แมชชีนเป็นวิธีการเรียนรู้ของเครื่องที่มีศักยภาพสูงในการใช้งานจริง โดยใช้คุณสมบัติเชิงเรขาคณิตในการคำนวณหาระนาบหลายมิติ (Hyperplane) ที่ดีที่สุดในการแยกข้อมูลออกจากกัน นอกจากนี้ยังจัดการกับปัญหาที่ไม่สามารถแบ่งแยกเชิงเส้นได้อีกด้วย อย่างไรก็ดีซัพพอร์ตเวกเตอร์แมชชีนมีข้อจำกัดในการใช้งาน คือสามารถจำแนกข้อมูลได้เพียง 2 ประเภท (Class) คือใช่หรือไม่ใช่เท่านั้น ด้วยเหตุนี้การนำซัพพอร์ตเวกเตอร์แมชชีนมาใช้กับการแยกข้อมูลหลายประเภท จำต้องอาศัยกระบวนการที่มีประสิทธิภาพ

มีผู้นำเสนอวิธีการแก้ปัญหาแบบหลายประเภทนี้หลายแนวทางเช่น การจำแนกแบบหนึ่งต่อที่เหลือ (One-against-the-rest) คือทำการเปรียบเทียบระหว่างประเภทข้อมูลหนึ่งกับประเภทข้อมูลอื่นทั้งหมด และการจำแนกแบบหนึ่งต่อหนึ่ง (One-against-one) เป็นการเปรียบเทียบประเภทข้อมูลหนึ่งกับประเภทข้อมูลอื่นทีละประเภท [5]

อัลกอริทึมแมกซ์วินที่เสนอโดย Friedman [4] เป็นหนึ่งในวิธีการจำแนกแบบหนึ่งต่อหนึ่ง โดยมีหลักการในการเปรียบเทียบข้อมูลประเภทใด ๆ กับข้อมูลประเภทอื่น ๆ ทีละประเภท แล้วเลือกประเภทที่ได้รับการเลือกมากครั้งที่สุดมาเป็นคำตอบ วิธีนี้ใช้เวลาในการสอนน้อยกว่าวิธีแบบหนึ่งต่อที่เหลือ แต่ใช้เวลาจำแนกสูงเนื่องจากต้องทำการเปรียบเทียบประเภทข้อมูลทีละคู่ให้ครบทุกคู่ ต่อมา Platt, et al. [7] ได้นำเสนอวิธีดีดีเอจี (Decision Directed Acyclic Graph (DDAG)) โดยใช้การสุ่มคู่ของประเภทข้อมูลมาทำการจำแนกทีละคู่ ประเภทที่ไม่ถูกเลือกจะถูกตัดทิ้งทันที ส่วนประเภทที่ถูกเลือกจะนำไปจำแนกกับประเภทข้อมูลที่เหลือต่อไป วิธีนี้ทำให้ลดจำนวนครั้งของการจำแนกลงจากวิธีแมกซ์วิน เนื่องจากไม่ต้องเปรียบเทียบข้อมูลให้ครบทุกคู่ แต่ให้ค่าความถูกต้องไม่แน่นอนขึ้นอยู่กับลำดับของประเภทข้อมูลที่ถูกเลือกมาทำการจำแนก ถ้าประเภทข้อมูลที่ถูกต้องถูกเลือกมาก่อนก็จะถูกจำแนกกับประเภทข้อมูลอื่นหลายครั้งทำให้มีโอกาสที่จะจำแนกผิดมาก

ต่อมา Ussivakul & Kijsirikul ได้เสนอวิธีเอดีเอจี (Adaptive Directed Acyclic Graph (ADAG)) [9] ซึ่งเป็นวิธีที่ดัดแปลงมาจากวิธีดีดีเอจี แต่ปรับวิธีการจำแนกให้ข้อมูลทุกประเภทมีจำนวนครั้งที่จะถูกจำแนกเท่ากัน โดยให้ทุกประเภทข้อมูลถูกเลือกมาจำแนกพร้อมกันในการจำแนกแต่ละชั้นแต่มีการจับคู่จำแนกแตกต่างกันไปตามลำดับของโนด วิธีนี้ลดการขึ้นต่อกันของลำดับการจำแนกในวิธีดีดีเอจีลงได้ แต่มีปัญหาในเรื่องของการเรียงลำดับโนดเพื่อจับคู่ของประเภทข้อมูลแทน ต่อมา Phetkaew, et al. ได้เสนอวิธีอาร์เอดีเอจี (Reordering Adaptive Directed Acyclic Graph (RADAG)) [8] ซึ่งวิธีนี้จะมีการเลือกลำดับของโนดที่เหมาะสมเพื่อมีโอกาสที่จะจำแนกผิดพลาดน้อยที่สุดในแต่ะชั้นโดยใช้อัลกอริทึมของการจับคู่สมบูรณ์แบบน้ำหนักน้อยสุด (Minimum-Weight Perfect Matching) [3] และจะมีการเรียงลำดับใหม่ในทุกชั้นการจำแนกด้วย ทำให้วิธีนี้ให้ค่าความถูกต้องที่แน่นอนและเชื่อถือได้และการเลือกลำดับที่ทำให้มีการจำแนกผิดน้อยที่สุดในแต่ละชั้น ทำให้วิธีอาร์เอดีเอจีมีค่าความถูกต้องสูงกว่าวิธีดีดีเอจีและวิธีเอดี-เอจี

อย่างไรก็ตามในวิธีที่กล่าวมาข้างต้น สำหรับการจำแนกข้อมูล k ประเภทใดๆ วิธีดีดีเอจี วิธีเอดีเอจีและวิธีอาร์เอดีเอจีต่างก็ต้องใช้จำนวนครั้งของการจำแนกข้อมูลเป็น $k-1$ ครั้ง เนื่องจากในการจำแนกข้อมูลแต่ละครั้งเป็นการจำแนกข้อมูลแบบหนึ่งต่อหนึ่ง ทำให้สามารถตัดข้อมูลที่ไม่ถูกต้องออกไปได้เพียง 1 ประเภทต่อครั้งเท่านั้น ในงานวิจัยนี้จะนำเสนอวิธีการใหม่มีชื่อว่า การแตกครึ่งแบบสมดุล (Balanced Dichotomization) โดยจะค้นหาระนาบของตัวจำแนกข้อมูลแบบหนึ่งต่อหนึ่งที่มีตำแหน่งของระนาบอยู่ในตำแหน่งที่แบ่งประเภทข้อมูลทั้ง 2 ด้านของประเภทข้อมูลทั้งหมดได้สมดุลกันที่สุดมาใช้จำแนกข้อมูล เพื่อให้การจำแนกข้อมูลแต่ละครั้งสามารถตัดประเภทที่ไม่ถูกต้องออกไปได้มากกว่า 1 ประเภท ทำให้จำนวนครั้งของการจำแนกรวมลดลงจากวิธีแบบหนึ่งต่อหนึ่งแบบต่าง ๆ ที่กล่าวมาข้างต้น

1.2 วัตถุประสงค์ของการวิจัย

ออกแบบและพัฒนาวิธีการเพื่อเพิ่มประสิทธิภาพในการเรียนรู้ของซัพพอร์ตเวกเตอร์แมชชีนแบบหลายประเภทให้มีจำนวนครั้งของการจำแนกลดลงจากวิธีการจำแนกข้อมูลแบบหนึ่งต่อหนึ่งแบบอื่น ๆ

1.3 แนวทางของการวิจัย

1. ศึกษาและทดสอบประสิทธิภาพของวิธีการจำแนกข้อมูลแบบหนึ่งต่อหนึ่งแบบต่าง ๆ ของซัพพอร์ตเวกเตอร์แมชชีนแบบหลายประเภท เช่น วิธีแมกซ์วิน วิธีเอดีเอจี และวิธีอาร์เอดี-เอจี
2. ออกแบบขั้นตอนวิธีการของการจำแนกแบบแตกครึ่งแบบสมดุล
3. ทดสอบประสิทธิภาพของวิธีการแตกครึ่งแบบสมดุลเพื่อเปรียบเทียบกับวิธีการจำแนกแบบต่าง ๆ ข้างต้น
4. วิเคราะห์ผลการเปรียบเทียบประสิทธิภาพของวิธีการแตกครึ่งแบบสมดุลกับวิธีการจำแนกแบบต่าง ๆ และสรุปผลการทดลอง

1.4 ขอบเขตของการวิจัย

1. การเปรียบเทียบประสิทธิภาพจะดูจากจำนวนครั้งของการจำแนกและค่าความถูกต้องของการจำแนกข้อมูล
2. ข้อมูลที่นำมาใช้ทดสอบจะใช้ข้อมูลพื้นฐานของ UCI Machine Learning Repository [2]

1.5 ประโยชน์ที่คาดว่าจะได้รับ

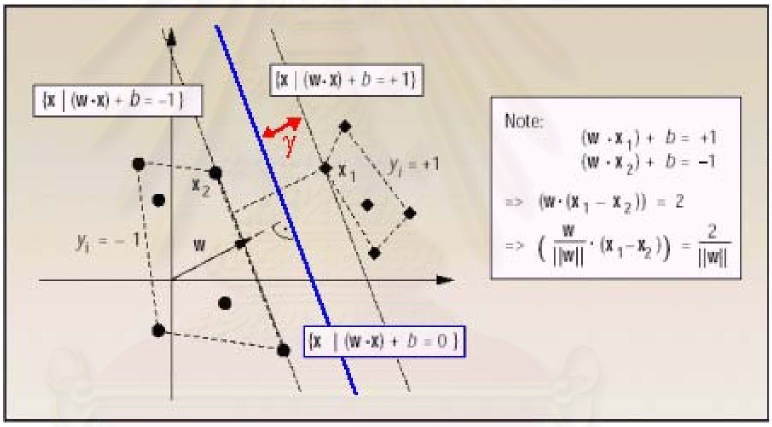
ได้อัลกอริทึมใหม่ในการเรียนรู้ของซัพพอร์ตเวกเตอร์แมชชีนแบบหลายประเภทที่มีจำนวนครั้งของการจำแนกลดลงจากวิธีการจำแนกแบบหนึ่งต่อหนึ่งแบบอื่น ๆ

.

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 **ซัพพอร์ตเวกเตอร์แมชชีน** (Support Vector Machines)

ซัพพอร์ตเวกเตอร์แมชชีนหนึ่งในเป็นเทคนิคการเรียนรู้ของเครื่องที่คิดค้นโดย Vapnik [10] ตั้งแต่ช่วงปี 1960 แต่เพิ่งเริ่มได้รับความนิยมในช่วงทศวรรษที่ผ่านมาเนื่องจากมีการนำไปใช้และให้ผลดี โดยมีแนวคิดหลัก ๆ อยู่ที่การสร้างระนาบหลายมิติที่ใช้แยกดีที่สุด (ดังตัวอย่างในรูปที่ 1) เพื่อใช้ในการแบ่งข้อมูล 2 ประเภทใด ๆ ออกจากกันด้วยระยะห่างที่กว้างที่สุด โดยซัพพอร์ตเวกเตอร์แมชชีนสามารถใช้แยกข้อมูลได้ทั้งปัญหาที่แยกแบบเชิงเส้นได้และปัญหาที่ไม่สามารถแยกแบบเชิงเส้นได้



รูปที่ 1 ตัวอย่างระนาบหลายมิติที่ใช้แยกดีที่สุด

2.1.1 **ซัพพอร์ตเวกเตอร์แมชชีนแบบเชิงเส้น** (Linear support vector machine)

สมมติมีเซตของข้อมูล D ที่ประกอบด้วยตัวอย่างจำนวน l ตัวในปริภูมิอันดับ n ที่มี 2 ประเภท (+1 และ -1)

$$D = \{(\mathbf{x}_k, y_k) \mid k \in \{1,..,l\}, \mathbf{x}_k \in \Re^n, y_k \in \{+1,-1\}\}. \quad (1)$$

ระนาบหลายมิติในปริภูมิอันดับ n ถูกกำหนดโดย $(\mathbf{w},b)$ เมื่อ $\mathbf{w}$ คือ เวกเตอร์ในปริภูมิอันดับ n ที่ตั้งฉากกับระนาบหลายมิติและ b คือค่าคงที่ระนาบหลายมิติ $(\mathbf{w}\cdot\mathbf{x}) + b$ จะแบ่งข้อมูลได้ก็ต่อเมื่อ

$$(\mathbf{w}\cdot\mathbf{x}_i)+b>0 \quad \text{if} \quad y_i=+1$$
$$(\mathbf{w}\cdot\mathbf{x}_i)+b<0 \quad \text{if} \quad y_i=-1. \tag{2}$$

ถ้าเราต้องการค่า $\mathbf{w}$ และ b ที่ทำให้จุดที่อยู่ใกล้ระนาบหลายมิติมากที่สุดมีระยะห่าง 1/|**w**| แล้ว จะได้

$$(\mathbf{w}\cdot\mathbf{x}_i)+b\geq 1 \quad \text{if} \quad y_i=+1$$
$$(\mathbf{w}\cdot\mathbf{x}_i)+b\leq -1 \quad \text{if} \quad y_i=-1 \tag{3}$$

ซึ่งเท่ากับ

$$y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b]\geq 1 \quad \forall i. \tag{4}$$

ในการค้นหาระนาบหลายมิติที่ใช้แบ่งข้อมูลดีที่สุด จะต้องค้นหาระนาบหลายมิติที่มีระยะห่างระหว่างข้อมูลที่ใช้สอนกับระนาบหลายมิติที่น้อยที่สุดมีค่ามากที่สุด ระยะห่างระหว่างข้อมูลตัวอย่างสองตัวจากประเภทที่แตกต่างกันมีค่าเท่ากับ

$$d(\mathbf{w},b)=\min_{\{\mathbf{x}_i|y_i=1\}}\frac{(\mathbf{w}\cdot\mathbf{x}_i)+b}{|\mathbf{w}|}-\max_{\{\mathbf{x}_i|y_i=-1\}}\frac{(\mathbf{w}\cdot\mathbf{x}_i)+b}{|\mathbf{w}|}. \tag{5}$$

จากสมการที่ (3) ค่าที่น้อยที่สุดและมากที่สุดที่เหมาะสมคือ ± 1 ดังนั้น จำเป็นต้องเพิ่มค่าของฟังก์ชัน

$$d(\mathbf{w},b)=\frac{1}{|\mathbf{w}|}-\frac{-1}{|\mathbf{w}|}=\frac{2}{|\mathbf{w}|}. \tag{6}$$

ให้สูงที่สุด ดังนั้นปัญหานี้จึงเท่ากับ

- ลดค่าของ $|\mathbf{w}|^2/2$ ให้ต่ำที่สุด
- โดยขึ้นกับเงื่อนไขต่อไปนี้

  (1) $y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b]\geq 1 \quad \forall i.$

สำหรับกรณีที่ไม่สามารถแบ่งข้อมูลได้โดยเงื่อนไขข้างต้น จะต้องปรับเงื่อนไขใหม่ โดยเพิ่มพจน์ค่าปรับซึ่งประกอบด้วยผลรวมของความคลาดเคลื่อน ξ_i จากขอบเขตเข้าไปในเงื่อนไข ดังนั้นปัญหาตอนนี้คือ

- ลดค่าของ $\dfrac{|\mathbf{w}|^2}{2}+C\sum_{i=1}^{l}\xi_i$ ให้ต่ำที่สุด
- โดยขึ้นกับเงื่อนไขต่อไปนี้

  (1) $y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b]\geq 1-\xi_i,$

  (2) $\xi_i \geq 0 \quad \forall i.$

พจน์ค่าปรับสำหรับตัวอย่างข้อมูลสอนที่ทำนายผิดพลาดถูกเพิ่มน้ำหนักโดย ค่าคงที่ C การเลือกค่า C สูงจะมีผลให้เพิ่มค่าของความคลาดเคลื่อน ξ_i และเพิ่มการคำนวณโดยทำให้การค้นหาหนทางที่จะลดจำนวนตัวอย่างข้อมูลสอนที่ทำนายผิดพลาดเพิ่มขึ้น

นำลากรองเกรียนมาใช้กับปัญหานี้ สามารถแปลงปัญหาเป็น

- ลดค่าของฟังก์ชันนี้ให้ต่ำที่สุด

$$L(\mathbf{w}, b, \alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{7}$$

- โดยขึ้นกับเงื่อนไขต่อไปนี้

(1) $0 \le \alpha_i \le C$, $\forall i$

(2) $\sum_{i=1}^{l} \alpha_i \, y_i = 0$

เมื่อ α_i เรียกว่าตัวคูณลากรอง ตัวอย่างข้อมูลสอนแต่ละตัวจะมีตัวคูณลากรองหนึ่งตัว ตัวอย่างที่มีค่า $\alpha_i > 0$ เรียกว่าซัพพอร์ตเวกเตอร์ และเป็นซัพพอร์ตเวกเตอร์ที่ทำให้ค่าทางขวามือของสมการที่ (4) เท่ากับ 1 ส่วนตัวอย่างอื่นๆที่มี $\alpha_i = 0$ สามารถตัดออกไปจากเซตของตัวอย่างข้อมูลสอนได้ โดยไม่มีผลกระทบใด ๆ ต่อผลลัพธ์ของระนาบหลายมิติ

ให้ α^0 ซึ่งเป็นเวกเตอร์ในปริภูมิอันดับ l แทนค่าต่ำสุดของ $L(\mathbf{w},b,\alpha)$ ถ้า $\alpha_i^0 > 0$ แล้ว $\mathbf{x}_i$ เป็นซัพพอร์ตเวกเตอร์ของระนาบหลายมิติที่แบ่งแยกดีสุด $(\mathbf{w}^0, b^0)$ สามารถเขียนในเทอมของ α^0 และข้อมูลสอน โดยเฉพาะอย่างยิ่งในเทอมของซัพพอร์ตเวกเตอร์ได้ดังนี้

$$\mathbf{w}^0 = \sum_{i=1}^{l} \alpha_i^0 y_i \mathbf{x}_i = \sum_{\text{support vectors}} \alpha_i^0 y_i \mathbf{x}_i . \tag{8}$$

$$b^0 = 1 - \mathbf{w}^0 \cdot \mathbf{x}_i \ \text{ for } \mathbf{x}_i \text{ with } y_i = 1 \text{ and } 0 < \alpha_i < C \tag{9}$$

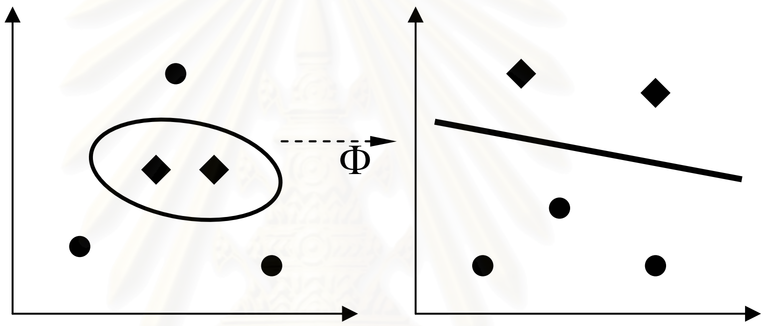
ระนาบหลายมิติที่แบ่งแยกดีสุดจะจำแนกจุดต่างๆ ตามเครื่องหมายของผลลัพธ์ของฟังก์ชัน $f(\mathrm{x})$

$$\begin{aligned} f(\mathbf{x}) &= \operatorname{sign}\left(\mathbf{w}^0 \cdot \mathbf{x} + b^0\right) \\ &= \operatorname{sign}\left(\sum_{\text{suport vectors}} \alpha_i^0 \, y_i (\mathbf{x}_i \cdot \mathbf{x}) + b^0 \right). \end{aligned} \tag{10}$$

ซัพพอร์ตเวกเตอร์ $\mathbf{x}_i$ ที่มี $\alpha_i^0 = C$ อาจจะถูกจำแนกผิดพลาดหรือไม่ก็ได้ แต่ $\mathbf{x}_i$ ตัวอื่นๆ จะจำแนกได้อย่างถูกต้อง

2.1.2 **ซัพพอร์ตเวกเตอร์แมชชีนแบบไม่เชิงเส้น** (Non-linear Support vector machine)

อัลกอริทึมที่ได้กล่าวมาแล้วใช้ได้กับข้อมูลที่แบ่งด้วยระนาบหลายมิติแบบเชิงเส้นได้เท่านั้น แต่กับข้อมูลที่ไม่อาจแบ่งแบบเชิงเส้นได้ ซัพพอร์ตเวกเตอร์แมชชีนแก้ปัญหานี้โดยแมปข้อมูลตัวอย่างไปยังปริภูมิอันดับสูง โดยมีสมมติฐานว่าข้อมูลที่ไม่สามารถแบ่งข้อมูลแบบเชิงเส้นในปริภูมิอันดับต่ำ เมื่อแมปไปสู่ปริภูมิอับดับสูงจะมีความสัมพันธ์เป็นเชิงเส้นได้ โดยเลือกใช้ฟังก์ชันการแมปที่ไม่เป็นเชิงเส้น นั่นคือเราเลือกฟังก์ชันในการแมป $\Phi : \Re^n \mapsto \mathrm{H}$ เมื่อปริภูมิของ $\mathbf{H}$ มีอันดับสูงกว่าปริภูมิอันดับ n เราสามารถค้นหาระนาบที่แบ่งแยกดีสุดในปริภูมิอันดับสูงที่เทียบเท่ากับการแยกที่ไม่เป็นเชิงเส้นใน $\Re^n$ (ดังแสดงในรูปที่ 2)



รูปที่ 2 แนวคิดการแมปข้อมูลไม่เชิงเส้นไปสู่ปริภูมิอันดับสูง

ตัวอย่างข้อมูลสอนที่ใช้ในสมการที่ (7) อยู่ในรูปของผลคูณเชิงสเกลาร์ระหว่างเวกเตอร์ ดังนั้นในปริภูมิอันดับสูง เราสามารถจัดการกับข้อมูลในรูปของ $\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$ ถ้าปริภูมิของ $\mathbf{H}$ มีอันดับสูงจะทำให้ยุ่งยากหรือใช้การคำนวณมาก อย่างไรก็ตามถ้าเรามีฟังก์ชันเคอร์เนลที่คำนวณ $k(\mathbf{x}_i,\mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$ แล้วเราสามารถใช้ฟังก์ชันนี้แทนที่ $\mathbf{x}_i\cdot\mathbf{x}_j$ ทุก ๆ ที่ในการคำนวณ และไม่จำเป็นต้องรู้ฟังก์ชันที่ใช้แมป Φ จริงๆ ฟังก์ชันเคอร์เนลที่นิยมใช้ได้แก่

Polynomial degree d: $$k(\mathbf{x},\mathbf{y}) = \left|\mathbf{x}\cdot\mathbf{y}+1\right|^d \qquad (11)$$

Radial basis function: $$k(\mathbf{x},\mathbf{y}) = e^{-\left|\mathbf{x}-\mathbf{y}\right|^2/c} \qquad (12)$$

2.2 **การจำแนกแบบหนึ่งต่อที่เหลือ** (One-against-the-rest)

แนวคิดเริ่มแรกที่ถูกนำมาใช้ในการแก้ปัญหาซัพพอร์ตเวกเตอร์แมชชีนแบบหลายประเภทคือ การจำแนกแบบหนึ่งต่อที่เหลือ [5] โดยมีแนวคิดในการเปรียบเทียบข้อมูลระหว่างประเภทข้อมูลหนึ่งกับประเภทข้อมูลที่เหลือทั้งหมดพร้อมกัน ทำให้มีการสร้างตัวจำแนก k ตัวโดยที่ k คือจำนวนประเภทข้อมูลทั้งหมด และ i คือประเภทข้อมูลใด ๆ ตัวจำแนกสำหรับประเภทข้อมูล i ใด ๆ ตามเงื่อนไขต่อไปนี้

$$
\begin{aligned}
&\min_{w^i,b^i,\xi^i} \quad \frac{1}{2}(\mathbf{w}^i)^T\mathbf{w}^i + C\sum_{j=1}^{l}\xi_j^i(\mathbf{w}^i)^T \\
&(\mathbf{w}^i)^T\Phi(\mathbf{x}_j)+b^i \ge 1-\xi_j^i, \quad \text{if } y_j = i \\
&(\mathbf{w}^i)^T\Phi(\mathbf{x}_j)+b^i \le -1+\xi_j^i, \quad \text{if } y_j \ne i \\
&\xi_j^i \ge 0, \quad j=1,\ldots,l
\end{aligned} \tag{13}
$$

หลังจากนั้นเราจะได้ตัวจำแนกสำหรับ k ประเภทข้อมูลจำนวน k ตัวดังนี้

$$
\begin{aligned}
&(\mathbf{w}^1)^T\Phi(\mathbf{x})+b^1 \\
&\qquad\vdots \\
&(\mathbf{w}^k)^T\Phi(\mathbf{x})+b^k.
\end{aligned}
$$

และจะได้ว่าตัวอย่าง x ใด ๆ เป็นคำตอบของประเภทข้อมูล i ใดๆ เมื่อค่าของฟังก์ชันตัวจำแนกสำหรับประเภทข้อมูล i นั้นให้ค่าสูงที่สุดจากตัวจำแนกทั้ง k ตัวข้างต้น

$$
\text{class of } \mathbf{x} \equiv \arg\max_{i=1,\ldots k}((\mathbf{w}^i)^T\Phi(\mathbf{x})+b^i)\ . \tag{14}
$$

แต่วิธีการจำแนกแบบหนึ่งต่อที่เหลือมีข้อเสียอยู่ที่ใช้เวลาในการสร้างระนาบสูง เนื่องจากการเปรียบเทียบประเภทหนึ่งประเภทกับประเภทที่เหลือพร้อมกันมีจำนวนข้อมูลสอนมีขนาดใหญ่และสร้างระนาบที่ใช้แยกได้ยาก

2.3 **การจำแนกแบบหนึ่งต่อหนึ่ง** (One-against-one)

อัลกอริทึมแมกซ์วิน (Max Wins) คือแนวคิดพื้นฐานของการจำแนกแบบหนึ่งต่อหนึ่ง โดยมีแนวคิดในการเปรียบเทียบประเภทใด ๆ กับประเภทที่เหลือทีละประเภทจนครบทุกคู่

โดยสร้างตัวจำแนกข้อมูลทั้งหมดจำนวน *k(k-1)/2* ตัว โดยที่ *k* คือจำนวนประเภทข้อมูลทั้งหมด ตัวจำแนกแต่ละตัวจะถูกสอนโดยข้อมูลจาก 2 ประเภท *i* และ *j* ใดๆ ตามเงื่อนไขดังนี้

$$\begin{aligned}
&\min_{w^{ij},b^{ij},\xi^{ij}} \quad \frac{1}{2}(\mathbf{w}^{ij})^T\mathbf{w}^{ij} + C\sum_t \xi_t^{ij}(\mathbf{w}^{ij})^T \\
&(\mathbf{w}^{ij})^T\Phi(\mathbf{x}_t) + b^{ij} \geq 1-\xi_t^{ij}, \quad \text{if } y_t = i \\
&(\mathbf{w}^{ij})^T\Phi(\mathbf{x}_t) + b^{ij} \leq -1+\xi_t^{ij}, \quad \text{if } y_t = j \\
&\xi_t^{ij} \geq 0.
\end{aligned} \tag{15}$$

แล้วใช้ฟังก์ชัน $sign((\mathbf{w}^{ij})^T\Phi(\mathbf{x}) + b^{ij})$ จำแนกให้คำตอบว่าตัวอย่าง *x* ใด ๆ อยู่ในประเภท *i* หรือ *j* หลังจากได้คำตอบทั้งหมดแล้ว วิธีแมกซ์วินจะนำคำตอบที่ได้จากตัวจำแนกทั้ง *k(k-1)/2* ตัว มาโหวตรวมกัน โดยประเภทที่ได้ผลโหวตสูงที่สุด ก็จะเป็นคำตอบของแมกซ์วิน ส่วนในกรณีที่มีผลโหวตสูงสุดมากกว่า 1 ประเภท ก็จะทำการสุ่มคำตอบจากประเภทเหล่านั้น

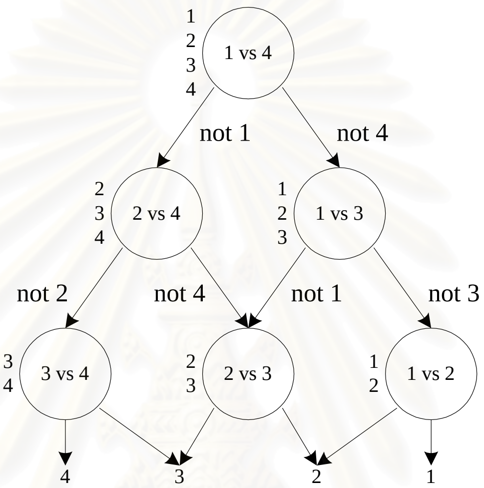
วิธีแมกซ์วินนี้ให้ค่าความถูกต้องในการจำแนกสูงกว่าวิธีแบบหนึ่งต่อที่เหลือ และใช้เวลาในการสร้างระนาบหลายมิติน้อยกว่าวิธีแบบหนึ่งต่อที่เหลือ [4] แต่ใช้เวลาในการจำแนกนานกว่าเนื่องจากต้องทำการจำแนกถึง *k(k-1)/2* ครั้ง

2.4 **วิธีดีดีเอจี** (Decision Directed Acyclic Graph (DDAG))

Platt, et al. [7] ได้เสนอวิธีดีดีเอจี โดยใช้โทโปโลยีของกราฟมาวิเคราะห์ ซึ่งนำตัวจำแนกแบบ 2 ประเภทหลายตัวมารวมกันเป็นตัวจำแนกแบบหลายประเภทหนึ่งตัว สำหรับปัญหาที่มี *k* ประเภท ในขั้นตอนการเรียนรู้จะสร้างตัวจำแนกแบบ 2 ประเภทจำนวน *k(k-1)/2* ตัว เช่นเดียวกับวิธีแมกซ์วิน แต่ในขั้นตอนการจำแนกจะใช้กราฟไม่มีวงแบบมีทิศทางโดยแต่ละโนดจะมีฟังก์ชันแบบ 2 ประเภท ซึ่งมีทั้งหมด *k(k-1)/2* โนดและมี *k* ใบ (ดูรูปที่ 3) แต่ละโนดคือตัวจำแนกข้อมูลของประเภทที่ *i* และ *j* เมื่อต้องการจำแนกข้อมูล *x* จะเริ่มจากโนดราก แล้วฟังก์ชันแบบ 2 ประเภทที่โนดจะถูกนำมาคำนวณ ถ้าได้ค่าของฟังก์ชันน้อยกว่าศูนย์ก็ออกที่กิ่งซ้ายแต่ถ้าได้มาก กว่าหรือเท่ากับศูนย์ก็ออกที่กิ่งขวา หลังจากนั้นฟังก์ชันที่โนดต่อมาจะถูกคำนวณจนถึงโนดใบซึ่งก็ คือคำตอบของดีดีเอจี

ในงานวิจัย [9] ได้แสดงให้เห็นถึงข้อเสียของดีดีเอจี อย่างแรกก็คือ ดีดีเอจีจะให้ผลลัพธ์ที่มีการแจกแจงความน่าจะเป็นไม่สม่ำเสมอ อันเนื่องมาจากการขึ้นอยู่กับลำดับของตัว

จำแนกในกราฟ ซึ่งมีผลต่อความน่าเชื่อถือของอัลกอริทึม นอกจากนี้ประเภทที่ถูกต้องที่วางอยู่ใกล้โนดราก จะมีข้อเสียเมื่อเปรียบเทียบกับประเภทที่อยู่ใกล้โนดใบ เนื่องจากจะมีโอกาสที่จะถูกจำแนกมากครั้งกว่าทำให้โอกาสที่จะถูกจำแนกผิดมีมากขึ้น อย่างที่สองคือจำนวนครั้งที่ประเภทที่ถูกต้องถูกจำแนกสูงทำให้มีความผิดพลาดสะสมสูงและส่งผลให้จำแนกผิดได้ เส้นทางการจำแนกของดีดีเอจีมีค่าความลึกเท่ากับ *k-1* นั่นคือจำนวนครั้งที่ประเภทที่ถูกต้องถูกจำแนกกับประเภทอื่นจะแปรผันตามจำนวนประเภทข้อมูลทั้งหมด



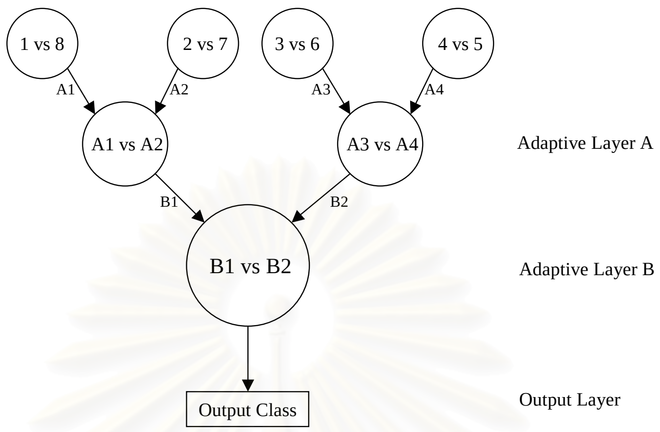
รูปที่ 3 โครงสร้างการจำแนกข้อมูลของดีดีเอจีสำหรับปัญหา 4 ประเภท

2.5 **วิธีเอดีเอจี** (Adaptive Directed Acyclic Graph (ADAG))

Ussivakul & Kijsirikul [9] ได้เสนอวิธีการที่จะลดข้อเสียของวิธีดีดีเอจี เรียกว่าวิธีเอดีเอจี เป็นกราฟไม่มีวงแบบมีทิศทางที่ปรับโครงสร้างการจำแนกข้อมูลเป็นรูปคล้ายสามเหลี่ยมคว่ำ โดยจะเลือกตัวจำแนกแบบ 2 ประเภทมาใช้จำแนกเพียง *k-1* ตัวในปัญหา *k* ประเภท ในขั้นตอนการจำแนกจะมีการแบ่งเป็นชั้นโดยในแต่ละชั้นจะเลือกตัวจำแนกแบบ 2 ประเภทมาจำนวนครึ่งหนึ่งในแต่ละชั้นเพื่อจับคู่ให้ทุกประเภทข้อมูลถูกจำแนก 1 ครั้งในแต่ละชั้น และเมื่อจำแนกไปแล้วจำนวนประเภทข้อมูลจะลดลงเหลือครึ่งหนึ่งในชั้นถัดมาจนเหลือประเภทเดียวในระดับล่างสุด (ดูรูปที่ 4)

จากโครงสร้างการจำแนกข้อมูลของวิธีเอดีเอจีทำให้ทุกประเภทข้อมูลมีโอกาสของจำนวนครั้งที่จะถูกจำแนกเท่ากัน ทำให้ลดจำนวนครั้งที่ประเภทที่ถูกต้องถูกจำแนกกับประเภทอื่นและลดความผิดพลาดสะสมลงได้ อย่างไรก็ตามความถูกต้องของวิธีนี้ยังคงขึ้นกับลำดับการจับคู่เพื่อจำแนกของประเภทข้อมูลในแต่ละชั้น ในวิธีอาร์เอดีเอจี [8] จะเสนอวิธีการเลือกลำดับที่

เหมาะสมในการ จับคู่จำแนกข้อมูลในแต่ละชั้น เพื่อให้มีโอกาสจำแนกข้อมูลผิดในแต่ละชั้นน้อยที่สุด

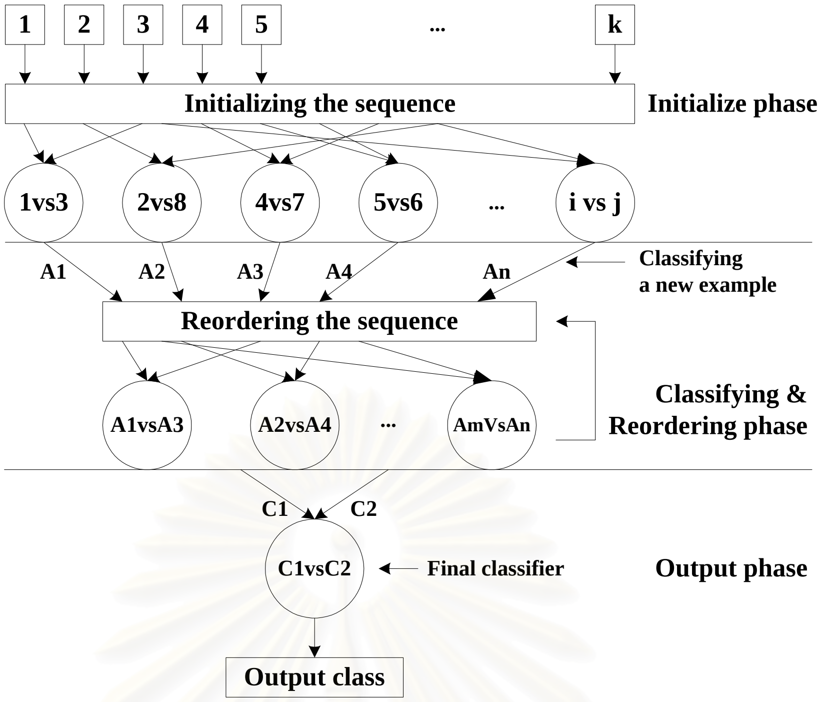


รูปที่ 4 โครงสร้างการจำแนกข้อมูลของเอดีเอจีสำหรับปัญหา 8 ประเภท

2.6 วิธีอาร์เอดีเอจี (Reordering Adaptive Directed Acyclic Graph (RADAG))

Phetkaew, et al. [8] นำเสนอวิธีการใหม่เรียกว่าวิธีอาร์เอดีเอจี เพื่อปรับปรุงความถูกต้องของวิธีเอดีเอจีเดิม โดยโครงสร้างการจำแนกข้อมูลนั้นจะเป็นรูปคล้ายสามเหลี่ยมคว่ำเหมือนวิธีเอดีเอจี แต่จะมีการเลือกลำดับที่ทำให้เกิดความผิดพลาดน้อยที่สุดของการจำแนกในแต่ละชั้น โดยจะใช้อัลกอริทึมของการจับคู่สมบูรณ์แบบน้ำหนักน้อยสุด (Minimum-Weight Perfect Matching) [3] ในการเลือกเพื่อจับคู่ให้ค่าขอบเขตความผิดพลาดรวมในแต่ละชั้นมีค่าน้อยที่สุด โดยค่าขอบเขตความผิดพลาดของฟังก์ชันแบบ 2 ประเภทแต่ละตัวหาได้จากค่าประสิทธิภาพโดยนัยทั่วไปของซัพพอร์ตเวกเตอร์แมชชีน [1] และเมื่อมีการจำแนกผ่านไปสู่ชั้นใหม่ก็จะมีการจัดลำดับการจำแนกใหม่ (Reordering) อีกจนเหลือประเภทเดียวในชั้นสุดท้ายซึ่งเป็นคำตอบของอาร์เอดีเอจี

จากโครงสร้างการจำแนกข้อมูลของอาร์เอดีเอจีที่มีการจัดลำดับการจำแนกข้อมูลใหม่ ในแต่ละชั้น (ดูรูปที่ 5) เพื่อหาลำดับที่ทำให้เกิดการจำแนกข้อมูลผิดพลาดน้อยที่สุดทำให้วิธีนี้มีความผิดพลาดลดลงจากวิธีเอดีเอจี อย่างไรก็ตามวิธีอาร์เอดีเอจียังคงต้องใช้จำนวนครั้งของการจำแนกข้อมูลเป็น *k-1* ครั้ง เหมือนวิธีเอดีเอจีและวิธีดีดีเอจี สำหรับการจำแนกข้อมูล *k* ประเภทใดๆ เนื่องจากในการจำแนกข้อมูลแต่ละครั้งจะสามารถตัดข้อมูลที่ไม่ถูกต้องออกไปได้เพียง 1 ประเภทต่อครั้งเท่านั้น ในบทถัดไปจะนำเสนอวิธีการใหม่ ที่ทำให้การจำแนกแต่ละครั้งตัดประเภทข้อมูลที่ไม่ถูกต้องออกไปได้ครั้งละมากกว่า 1 ประเภท เพื่อลดจำนวนครั้งของการจำแนกลง



รูปที่ 5 โครงสร้างการจำแนกข้อมูลของอาร์เอดีเอจี

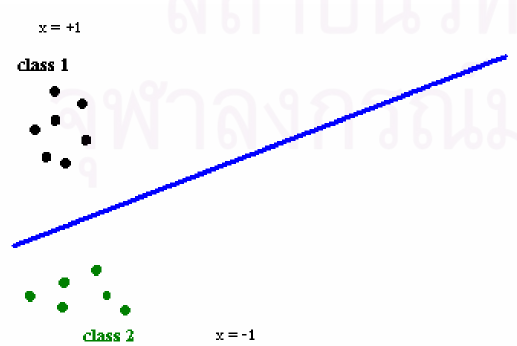
บทที่ 3
การแตกครึ่งแบบสมดุล

3.1 การจำแนกข้อมูลโดยการแตกครึ่งแบบสมดุล (Balanced Dichotomization Classification)

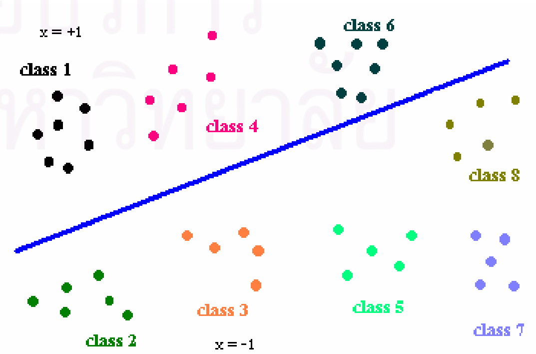
งานวิจัยนี้ได้นำเสนอวิธีการจำแนกข้อมูลแบบใหม่ มีชื่อว่า การจำแนกข้อมูลโดยการแตกครึ่งแบบสมดุล โดยมีแนวคิดในการจำแนกข้อมูลโดยใช้ระนาบหลายมิติที่อยู่ในตำแหน่งกึ่งกลางที่สุดของข้อมูลมาจำแนกเพื่อลดจำนวนครั้งของการจำแนกลง โดยมีขั้นตอนในการจำแนกข้อมูลดังนี้

3.1.1 การค้นหาตำแหน่งของระนาบหลายมิติ

ในการสร้างฟังก์ชันของระนาบหลายมิติในการจำแนกแบบ 2 ประเภทตามปกตินั้น เราจะได้ตำแหน่งของระนาบอยู่กึ่งกลางระหว่าง 2 ประเภทข้อมูลที่เราใช้สอน โดยที่ตำแหน่งของข้อมูลทั้ง 2 ประเภทจะอยู่คนละด้านของระนาบ ตัวอย่างเช่น ถ้าสร้างระนาบระหว่างประเภท 1 กับประเภท 2 ก็จะได้ระนาบหลายมิติอยู่กึ่งกลางระหว่าง 1 กับ 2 และได้ตำแหน่งของประเภท 1 อยู่ในด้าน $x = +1$ และตำแหน่งของประเภท 2 อยู่ในด้าน $x = -1$ เป็นต้น (ดูรูปที่ 6 (ก)) แต่จะไม่รู้ตำแหน่งเทียบกับประเภทข้อมูลอื่น ทำให้การจำแนกข้อมูล 1 ครั้ง ตัดประเภทที่ไม่ถูกต้องออกไปได้เพียง 1 ประเภท ในวิธีนี้จึงมีการค้นหาตำแหน่งของระนาบเทียบกับประเภทข้อมูลอื่นเพิ่มโดย นำฟังก์ชันของระนาบแบบ 2 ประเภทที่มีอยู่แล้วไปคำนวณหาตำแหน่งของประเภทข้อมูลอื่นเพิ่มเติมก็จะได้ตำแหน่งของระนาบเมื่อเทียบกับข้อมูลทั้งหมด (ดูรูปที่ 6 (ข))



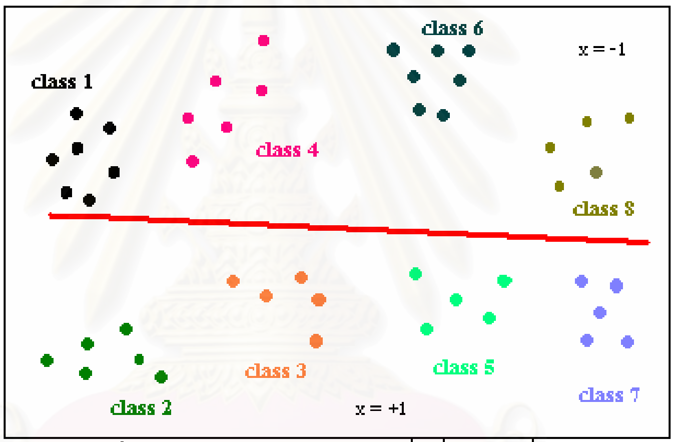
รูปที่ 6 (ก) ข้อมูลเรียนรู้ของระนาบตามปกติ



รูปที่ 6 (ข) ข้อมูลเรียนรู้เมื่อมีการค้นหาตำแหน่งเทียบกับประเภทข้อมูลอื่นเพิ่ม

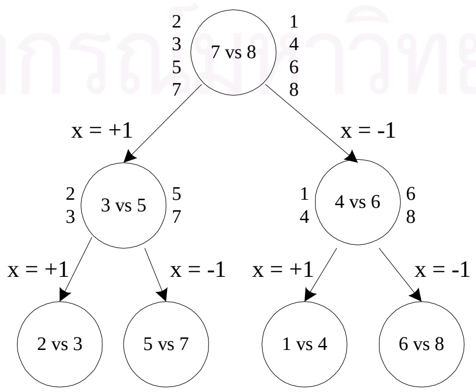
3.1.2 การค้นหาระนาบแตกครึ่งแบบสมดุล

เมื่อทำการคำนวณเพื่อค้นหาตำแหน่งของระนาบหลายมิติเทียบกับประเภทข้อมูลอื่นๆ ตามขั้นตอนข้างต้นแล้ว การจำแนกข้อมูลแบบแตกครึ่งแบบสมดุลจะทำการค้นหาระนาบที่สามารถแบ่งข้อมูลในตำแหน่งสมดุลที่สุดของข้อมูลทั้งหมดมาจำแนก เพื่อให้การจำแนกข้อมูลแต่ละครั้งสามารถตัดประเภทข้อมูลที่ไม่ถูกต้องออกไปได้มากที่สุด ตัวอย่างเช่น ในรูปที่ 6 (ข) ระนาบตัวนี้สามารถแบ่งข้อมูลได้เหลือ 3 ประเภทในด้าน $x = +1$ และเหลือ 5 ประเภทในด้าน $x = -1$ ซึ่งระนาบนี้ยังไม่ใช่ระนาบที่สามารถแบ่งข้อมูลได้ในตำแหน่งกึ่งกลางที่สุด รูปที่ 7 แสดงให้เห็นถึงตัวอย่างของระนาบแตกครึ่งแบบสมดุลที่สามารถแบ่งข้อมูลได้ในตำแหน่งกึ่งกลางที่สุดของข้อมูลโดยที่สามารถแบ่งประเภทข้อมูลในแต่ละด้านให้เหลือครึ่งหนึ่งได้เท่ากันพอดี



รูปที่ 7 ตัวอย่างระนาบแตกครึ่งที่สมดุลที่สุด

ซึ่งถ้าเราสามารถหาระนาบที่แบ่งข้อมูลเหลือครึ่งหนึ่งได้ทุกรอบ จะทำให้จำนวนครั้งของการจำแนกที่ดีที่สุดลดลงเหลือ $log_2(k)$ ครั้งในปัญหา k ประเภท และจะได้โครงสร้างของการจำแนกในรูปที่ 8 เป็นตัวอย่างของกรณีดีสุดสำหรับปัญหาการจำแนกข้อมูล 8 ประเภท



รูปที่ 8 ตัวอย่างโครงสร้างของวิธีการแตกครึ่งแบบสมดุลในกรณีดีสุด

โดยในการคำนวณหาระนาบแตกครึ่งที่สมดุลที่สุดจะคำนวณจากจำนวนกลุ่มที่ตกในแต่ละด้านของระนาบโดยใช้ฟังก์ชัน

$$UnBalancedScore = CountLeft + CountRight + (0.5)\,|CountLeft - CountRight|$$

โดยที่ *CountLeft* คือจำนวนประเภทข้อมูลที่มีตำแหน่งอยู่ในฝั่งซ้ายของระนาบ และ *CountRight* คือจำนวนประเภทข้อมูลที่มีตำแหน่งอยู่ในฝั่งขวาของระนาบ ซึ่งระนาบที่ได้ค่าของฟังก์ชัน *UnBalancedScore* น้อยที่สุดก็จะถูกเลือกให้เป็นระนาบที่แตกครึ่งแบบสมดุลที่สุด ตัวอย่างเช่นจากรูปที่ 6 (ข) มีจำนวนประเภทข้อมูลที่ตกในแต่ละด้านเป็น 3 และ 5 ประเภทจะคำนวณค่าของฟังก์ชันนี้ได้เท่ากับ 3 + 5 + (0.5) |3-5| = 9 ส่วนรูปที่ 7 มีประเภทข้อมูลที่ตกในแต่ละด้านเป็น 4 ประเภททั้ง 2 ด้านจะคำนวณค่าได้เท่ากับ 4 + 4 +(0.5) |4-4| = 8 จะได้ว่าระนาบในรูปที่ 7 มีค่าของฟังก์ชัน *UnBalancedScore* น้อยกว่าในรูปที่ 6 (ข) ก็จะทำการเลือกระนาบในรูปที่ 7 เป็นระนาบแตกครึ่งแบบสมดุลที่สุด

นัยสำคัญในการคำนวณของฟังก์ชัน *UnBalancedScore* นี้แบ่งออกเป็น 2 ส่วน ส่วนแรกคือ *CountLeft* + *CountRight* เป็นการนับผลรวมของประเภทข้อมูลที่ตกอยู่ทั้ง 2 ด้าน เพื่อให้ความสำคัญกับระนาบที่แบ่งข้อมูลแล้วเหลือประเภทข้อมูลรวมน้อยที่สุด ซึ่งก็คือระนาบที่สามารถแบ่งข้อมูลได้โดยมีประเภทข้อมูลใดที่มีตำแหน่งตกอยู่ทั้ง 2 ด้านของระนาบน้อยที่สุด และส่วนที่ 2 คือ (0.5) |*CountLeft* – *CountRight*| เป็นการให้ความสำคัญกับระนาบที่แบ่งข้อมูลให้มีประเภทข้อมูลเหลือในแต่ละด้านได้สมดุลกันที่สุด ถ้าระนาบแบ่งข้อมูลแล้วเหลือจำนวนประเภทข้อมูล 2 ด้านแตกต่างกันมาก ค่าของฟังก์ชัน *UnBalancedScore* ส่วนนี้ก็จะเพิ่มขึ้น แต่ถ้าระนาบแบ่งข้อมูลแล้วได้จำนวนประเภทข้อมูลทั้ง 2 ด้านเท่ากันคือสมดุลกันทั้ง 2 ด้าน ค่าของฟังก์ชันส่วนนี้ก็จะกลายเป็นศูนย์ ซึ่งก็คือได้ค่า *UnBalancedScore* ในส่วนนี้น้อยที่สุด

โดยที่ค่าพารามิเตอร์ 0.5 เป็นการกำหนดเพื่อให้น้ำหนักความสำคัญของฟังก์ชัน *UnBalancedScore* ทั้ง 2 ส่วนนี้ มีความเท่าเทียมกัน ถ้ากำหนดค่าพารามิเตอร์นี้สูงกว่า 0.5 จะเป็นการให้ความสำคัญกับระนาบที่แบ่งข้อมูลได้สมดุลกันมากกว่าก่อนระนาบที่แบ่งข้อมูลได้เหลือประเภทข้อมูลรวมน้อยกว่า แต่ถ้ากำหนดค่าพารามิเตอร์นี้ต่ำกว่า 0.5 จะให้ความสำคัญกับระนาบที่แบ่งข้อมูลได้เหลือประเภทข้อมูลรวมน้อยกว่าก่อนระนาบที่แบ่งข้อมูลได้สมดุลกันมากกว่า ตัวอย่างเช่น มีระนาบ 2 ตัว คือ A และ B สำหรับการจำแนกข้อมูล 12 ประเภท โดยระนาบ A และ B แบ่งข้อมูลในแต่ละด้านได้เป็น 3 และ 9 ประเภท กับ 6 และ 8 ประเภทตามลำดับ จะเห็นได้ว่าระนาบ A มีผลรวมของประเภทข้อมูลหลังการจำแนกน้อยกว่าระนาบ B

แต่ระนาบ B มีความสมดุลกันของประเภทข้อมูลทั้ง 2 ด้านมากกว่าระนาบ A ถ้าใช้พารามิเตอร์ 0.5 ในการคำนวณจะได้ค่า *UnBalancedScore* ของระนาบ A = 3 + 9 + (0.5) |3-9| = 15 และค่า *UnBalancedScore* ของระนาบ B = 6 + 8 + (0.5) |6-8| = 15 ซึ่งเท่ากันทั้ง 2 ระนาบ แต่เมื่อเราเพิ่มค่าพารามิเตอร์ให้มากกว่า 0.5 ค่าที่ได้จากระนาบ A จะสูงขึ้นเร็วกว่าค่าที่ได้จากระนาบ B เนื่องจากมีความสมดุลกันน้อยกว่า ทำให้เราเลือก B ที่จำนวนผลรวมทั้ง 2 ด้านมากกว่ามาเป็นระนาบที่ใช้ในการจำแนก แต่ถ้าลดค่าพารามิเตอร์ 0.5 ลง ระนาบ A ก็จะได้เปรียบเพราะค่าฟังก์ชันจะลดลงได้เร็วและมีค่าได้ต่ำกว่าระนาบ B เพราะมีค่าของผลรวมในส่วนแรกน้อยกว่า B ทำให้เราเลือกระนาบ A ที่มีความสมดุลกันน้อยกว่ามาใช้ในการจำแนก

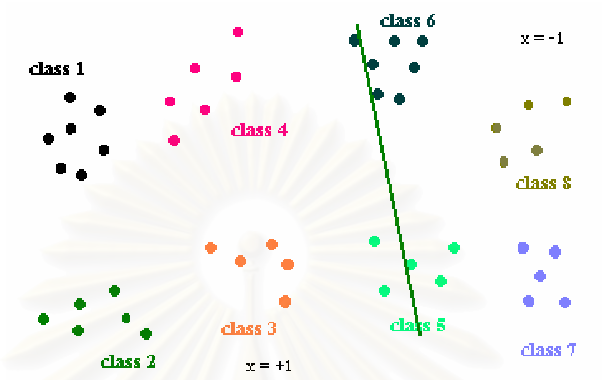
ด้วยเหตุนี้เราจึงกำหนดค่าพารามิเตอร์ของฟังก์ชัน *UnBalancedScore* นี้เป็น 0.5 เพื่อให้น้ำหนักของความสำคัญจากค่าที่ได้จากการคำนวณใน 2 ส่วนนี้เท่าเทียมกัน เพื่อให้ได้ระนาบที่แบ่งข้อมูลได้สมดุลกันมากที่สุด โดยที่มีจำนวนผลรวมของประเภทข้อมูลที่เหลือทั้ง 2 ด้านน้อยที่สุดเท่าที่จะเป็นไปได้ไปพร้อมกัน

3.2 การตัดเล็มและการกำหนดขอบเขตความผิดพลาด

จากหัวข้อที่แล้วเราได้ทราบว่า วิธีการจำแนกโดยการแตกครึ่งแบบสมดุลในกรณีดีสุดนั้นจะสามารถลดจำนวนครั้งของการจำแนกให้เหลือเพียง $log_2(k)$ ครั้งในปัญหา k ประเภท แต่ในการใช้งานจริง โอกาสที่จำนวนครั้งของการจำแนกจะลดลงดีที่สุดจนเหลือ $log_2(k)$ ครั้ง มีค่อนข้างน้อย เนื่องจากไม่สามารถหาระนาบที่แบ่งข้อมูลในตำแหน่งกึ่งกลางพอดีได้ เพราะระนาบที่เราเลือกมาจำแนกมักจะเจอประเภทข้อมูลเดียวกันตกอยู่ทั้ง 2 ด้านของระนาบทำให้ไม่สามารถจำแนกได้ ความลึกโครงสร้างของการจำแนกจึงไม่สามารถลดลงเหลือ $log_2(k)$ ครั้ง

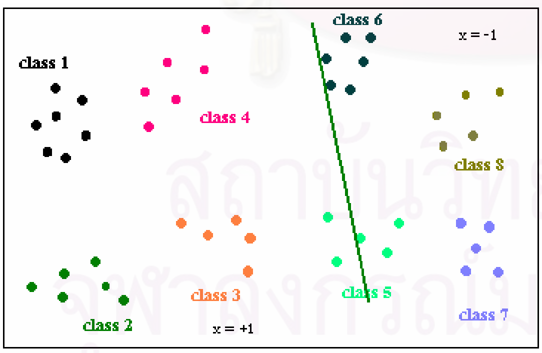
เราจึงต้องทำการตัดเล็มโครงสร้างของการจำแนกโดยการกำหนดค่าร้อยละในตัดเล็ม (P : Pruning Percentage) เพื่อกำจัดประเภทข้อมูลที่มีตำแหน่งอยู่ทั้ง 2 ด้านของระนาบในด้านที่มีจำนวนร้อยละน้อยกว่าที่เรากำหนดไว้ทิ้งไป เพื่อให้ระนาบสามารถจำแนกข้อมูลได้รวดเร็วขึ้น โดยการทำให้ระนาบแต่ละตัวไม่นำประเภทข้อมูลที่อยู่ในด้านที่มีจำนวนร้อยละน้อยกว่าค่าร้อยละในการตัดเล็มมานับรวมในการคำนวณหาจำนวนประเภทข้อมูลที่เหลือในแต่ละด้าน (*CountLeft*, *CountRight*) ซึ่งการกำจัดข้อมูลนี้จะเป็นการกำจัดข้อมูลตำแหน่งของประเภทข้อมูลที่ระนาบแต่ละตัวมองเห็นจะไม่ส่งผลกระทบต่อข้อมูลตำแหน่งของระนาบตัวอื่น ๆ

ตัวอย่างจากรูปที่ 9 แสดงให้เห็นว่ามีประเภทข้อมูลที่มีตำแหน่งอยู่ทั้ง 2 ด้านของระนาบอยู่ 2 ประเภทคือกลุ่ม 5 และกลุ่ม 6 โดยที่มีจำนวนร้อยละของข้อมูลในด้านที่น้อยกว่าของกลุ่ม 5 เป็น 40% และของกลุ่ม 6 เป็น 14.29%

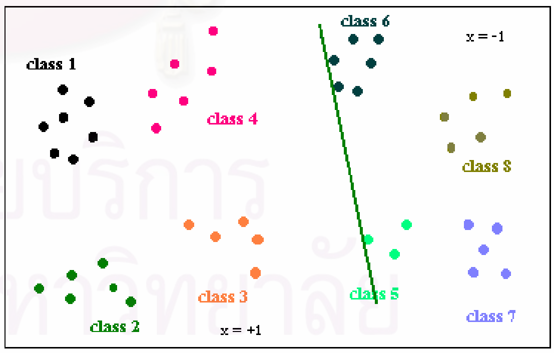


รูปที่ 9 ตัวอย่างประเภทข้อมูลที่มีตำแหน่งอยู่ทั้ง 2 ด้านของระนาบ

ถ้าเรากำหนดการกำหนดค่าร้อยละในตัดเล็มอยู่ที่ 15% ข้อมูลตำแหน่งของกลุ่ม 6 ในด้านที่น้อยกว่าจะถูกกำจัดทิ้ง (ดูรูปที่ 10 (ก)) แต่ถ้าเรากำหนดค่าร้อยละอยู่ที่ 40 % ข้อมูลตำแหน่งของทั้งกลุ่ม 5 และ 6 ในด้านที่น้อยกว่าจะถูกกำจัดทิ้งทั้ง 2 ประเภท ทำให้เหลือประเภทข้อมูลในแต่ละด้านเป็นครึ่งหนึ่งเท่ากันพอดี (ดูรูปที่ 10 (ข)) ทำให้ได้ระนาบที่อยู่ในตำแหน่งกึ่งกลางที่สุดในการจำแนกของวิธีแตกครึ่งแบบสมดุลได้พอดี



รูปที่ 10 **(ก)** กำหนดค่าร้อยละ 15 %



รูปที่ 10 **(ข)** กำหนดค่าร้อยละ 40 %

อีกหนึ่งปัญหาที่ต้องทำการกำหนดค่าพารามิเตอร์ควบคุมก็คือ ค่าความผิดพลาดของการจำแนก เนื่องจากในการค้นหาระนาบที่แตกครึ่งแบบสมดุลที่สุดนั้น เราได้ใช้ระนาบของการจำแนกแบบหนึ่งต่อหนึ่งทุกตัวมาทำการเลือก แต่ระนาบแต่ละตัวนั้นมีค่าความผิดพลาดของการจำแนกมากน้อยแตกต่างกัน ทำให้ระนาบที่เราเลือกเป็นระนาบที่อยู่กึ่งกลางที่สุดอาจจะมีค่า

ความผิดพลาดอยู่ในช่วงที่สูง เมื่อเรานำมาจำแนกแม้ว่าจะจำแนกข้อมูลได้อย่างรวดเร็วแต่ก็ทำให้เกิดความผิดพลาดสูงตามมาด้วย

ค่าประสิทธิภาพโดยนัยทั่วไป (Generalization Performance) [1] ของซัพพอร์ตเวกเตอร์แมชชีนสามารถบอกขอบเขตความผิดพลาดของระนาบที่ใช้จำแนกข้อมูลแต่ละตัวได้ โดยกำหนดชนิดของฟังก์ชันเป็นค่าตัวเลขจำนวนจริงภายในลูกบอลที่มีรัศมี R ค่ารัศมีไม่เกิน $\Re^n$ เป็นฟังก์ชันดังนี้ $F = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} : \|\mathbf{w}\| \le 1, \|\mathbf{x}\| \le R\}$ จะมีค่าคงที่ c ที่สำหรับการกระจายทุกประเภท ด้วยความน่าจะเป็นอย่างน้อย 1-δ บนตัวอย่าง z ที่เลือกมาอย่างอิสระจากกัน m ตัว ถ้าตัวจำแนก $h = \operatorname{sgn}(f) \in \operatorname{sgn}(F)$ มีค่าระยะห่างระหว่างระนาบหลายมิติกับข้อมูลที่ใช้สอนอย่างน้อยเป็น γ สำหรับตัวอย่างทุกตัวใน z แล้ว ความผิดพลาดของตัวจำแนก h จะไม่เกิน

$$\frac{c}{m}\left(\frac{R^2}{\gamma^2}\log^2 m + \log\left(\frac{1}{\delta}\right)\right). \tag{16}$$

นอกเหนือจากนี้แล้ว ด้วยความน่าจะเป็นอย่างน้อย 1-δ สำหรับทุกตัวจำแนก $h \in \operatorname{sgn}(F)$ จะมีความผิดพลาดไม่เกิน

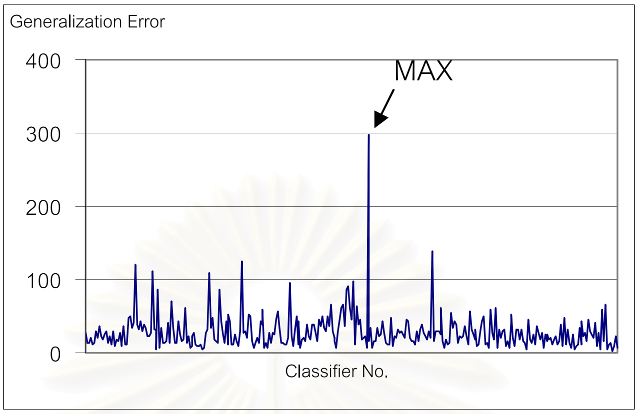
$$\frac{k}{m} + \sqrt{\frac{c}{m}\left(\frac{R^2}{\gamma^2}\log^2 m + \log\left(\frac{1}{\gamma}\right)\right)} \tag{17}$$

เมื่อ k คือจำนวนตัวอย่างข้อมูลใน z ที่มีระยะห่างระหว่างระนาบหลายมิติกับซัพพอร์ตเวกเตอร์น้อยกว่า γ

เมื่อรู้ค่าความผิดพลาดของระนาบแต่ละตัวแล้ว ในวิธีการแตกครึ่งแบบสมดุล จะมีการกำหนดขอบเขตของการจำแนก โดยกำหนดให้ระนาบที่เราจะนำมาค้นหาระนาบที่อยู่ในตำแหน่งกึ่งกลางที่สุดในแต่ละรอบ ต้องมีค่าประสิทธิภาพโดยนัยทั่วไปอยู่ในช่วง R เท่าของค่าประสิทธิภาพโดยนัยทั่วไปที่น้อยที่สุดของระนาบทั้งหมดในรอบนั้น ๆ เพื่อควบคุมไม่ให้เราเลือกระนาบที่มีความผิดพลาดสูงมาใช้จำแนก

เช่น ตัวอย่างในรูปที่ 11 แสดงตัวอย่างค่าประสิทธิภาพโดยนัยทั่วไปของระนาบที่ใช้จำแนกข้อมูล 325 ระนาบในปัญหา 26 ประเภท แสดงให้เห็นว่าระนาบแต่ละตัวมีค่า

ประสิทธิภาพโดยนัยทั่วไปต่าง ๆ กัน โดยมีค่าสูงที่สุดที่ระนาบ MAX (ลูกศรชี้) ถ้าระนาบสมดุลที่สุดที่เราเลือกเป็นระนาบตัวนี้ ก็ทำให้มีโอกาสจำแนกผิดสูงมีมากขึ้น



รูปที่ 11 ตัวอย่างช่วงค่าประสิทธิภาพโดยนัยทั่วไปของตัวจำแนกข้อมูล 325 ตัว

อย่างไรก็ตามในการกำหนดค่าพารามิเตอร์ของทั้งร้อยละการตัดเล็ม (P) และช่วงค่าของความผิดพลาด (R) นั้น จำเป็นต้องมีการคำนวณหาค่าที่เหมาะสมสำหรับพารามิเตอร์แต่ละตัว เนื่องจากถ้ากำหนดร้อยละการตัดเล็มมากจนเกินไปก็จะทำให้มีข้อมูลที่มีประโยชน์ถูกกำจัดทิ้งไปด้วยและทำให้การจำแนกผิดมีสูงขึ้น หรือถ้าเรากำหนดช่วงค่าของความผิดพลาดน้อยจนเกินไปก็จะทำให้เราไม่สามารถหาระนาบที่อยู่ในตำแหน่งกึ่งกลางที่สุดได้ โดยวิธีการเลือกหาค่าที่เหมาะสมของพารามิเตอร์ทั้ง 2 ตัวนี้ เราจะใช้การแบ่งข้อมูลสอน (Training Data) ออกเป็นข้อมูลสอนจริง (Actual Training Data) และข้อมูลทดสอบความถูกต้อง (Validation Data) แล้วนำข้อมูลสอนจริงไปสร้างระนาบ จากนั้นนำระนาบที่ได้มาทำการจำแนกกับข้อมูลทดสอบความถูกต้องโดยกำหนดค่าพารามิเตอร์ต่างๆ กันไป แล้วเลือกค่าพารามิเตอร์ที่ทำให้มีความผิดพลาดน้อยสุดในข้อมูลทดสอบความถูกต้องมาเป็นค่าที่เหมาะสมที่จะใช้กับข้อมูลทดสอบจริง (Test Data) ต่อไป

3.3 **อัลกอริทึมของการแตกครึ่งแบบสมดุล**

อัลกอริทึมของการแตกครึ่งแบบสมดุลแสดงไว้ในตารางที่ 1 ค่าพารามิเตอร์ P และ R คือค่าของร้อยละการตัดเล็มและช่วงค่าของค่าประสิทธิภาพโดยนัยทั่วไปของระนาบที่จะนำมาหาระนาบสมดุลที่สุด ดังที่ได้อธิบายไว้ในหัวข้อที่แล้ว โดยในการทำงานจะเริ่มจากการแบ่งข้อมูลสอนออกเป็น 2 ส่วน คือ ข้อมูลสอนจริงและข้อมูลทดสอบความถูกต้อง จากนั้นจะทำการคำนวณค่าจำนวนประเภทข้อมูลที่ตกในแต่ละด้านของระนาบและค่าประสิทธิภาพโดยนัยทั่วไปของระนาบทั้ง $k(k-1)/2$ ตัว และจะทำการจำแนกกับข้อมูลทดสอบความถูกต้องด้วยพารามิเตอร์ P

และ R ต่างๆ กันเพื่อค้นหาค่าพารามิเตอร์ P และ R ที่ทำให้การจำแนกมีค่าความถูกต้องสูงที่สุดในการจำแนกกับข้อมูลทดสอบความถูกต้องและเลือกโครงสร้างต้นไม้ของการจำแนกที่ได้จากค่า P และ R นั้นๆ มาใช้เป็นโครงสร้างสำหรับจำแนกกับข้อมูลทดสอบจริงต่อไป

ตารางที่ 1: อัลกอริทึมของการแตกครึ่งแบบสมดุล

1. Divide the training set into the actual training set *A* and the validation set *V*.
2. Use *A* to construct all binary SVMs. /* $k(k-1)/2$ hyperplanes */
3. FOR EACH binary SVM
   - Calculate the percentage of data of each class in each side of the hyperplane.
   - Calculate the generalization error.
4. *BestTree* $\leftarrow$ *nil*
5. Constructing the dichotomization tree by following steps.

   FOR P = 0.00 TO *MaxP*, increased by ΔP

   FOR R = 1.0 TO *MaxR*, increased by ΔR

   (1) Start from the root node of the tree with all possible binary SVMs.

   (2) Let *BinarySVMs* be a list of binary SVMs in the current under consideration.

   (3) *BinarySVMs* $\leftarrow$ sort *BinarySVMs* by increasing generalization error.

   (4) *CurrentSVM* $\leftarrow$ First(*BinarySVMs*) /* the lowest generalization error */

   (5) WHILE (generalization error of *CurrentSVM*) $\leq R$ x (the minimum value of generalization error of all SVMs in *BinarySVMs*) DO
   - Calculate the number of data class with the percentage of data in the left-hand-side more than *P* to *CountLeft*.
   - Calculate the number of data class with the percentage of data in the right-hand-side more than *P* to *CountRight*.
   - Calculate Unbalanced function

     UnBalancedScore = *CountLeft* + *CountRight* + (0.5)|*CountLeft* - *CountRight*|
   - *CurrentSVM* $\leftarrow$ Next(*BinarySVMs*)

   (6) *BestSVM* $\leftarrow$ the SVM with the smallest *UnBalancedScore*.

   (7) Assign *BestSVM* as the decision node in the tree.

   (8) Sort down classes and data to each side of the decision node.

   (9) IF (there is only one class in every leaf node)

   THEN GOTO step (10)

   ELSE GOTO step (2) for iterating over two leaf nodes.

   (10) Use the current tree to classify the data in the validation set *V*.

   (11) IF (the accuracy of the tree on *V* is higher than that of the *BestTree*) THEN

   BestTree $\leftarrow$ the current tree.

   End For

   End For
6. Output the best dichotomization tree, *BestTree*.

บทที่ 4
วิธีการทดลองและผลการทดลอง

4.1 วิธีการทดลอง

ในการทดลอง เราจะทำการทดลองโดยใช้ชุดข้อมูลทดสอบของ UCI Machine Learning Repository [2] มาทำการทดสอบประสิทธิภาพของการจำแนกกับวิธีการจำแนกข้อมูลแบบแตกครึ่งแบบสมดุลและวิธีการจำแนกข้อมูลแบบหนึ่งต่อหนึ่งอีก 3 วิธีคือ วิธีแมกซ์วิน วิธีเอดีเอจีและวิธีอาร์เอดีเอจี ซึ่งจะเปรียบเทียบประสิทธิภาพจากจำนวนครั้งของการจำแนกและค่าความถูกต้องของการจำแนก โดยจำนวนครั้งของการจำแนกจะทำการเปรียบเทียบกับค่าคาดหวังดีสุดคือ $\log_2 K$ ครั้งด้วย และในการทดลองจะทำการทดลองโดยใช้ฟังก์ชันเคอร์เนลในการเรียนรู้ 2 ฟังก์ชันคือ Polynomial Kernel และ RBF kernel

ตารางที่ 2 จะแสดงรายละเอียดของชุดข้อมูลที่นำมาทดสอบซึ่งมีทั้งหมด 8 ชุดข้อมูล โดยจะแสดงจำนวนประเภทข้อมูล (Class) จำนวนข้อมูลสอน (Training Data) จำนวนข้อมูลทดสอบ (Test Data) และจำนวนคุณสมบัติ (Feature) ของชุดข้อมูลต่าง ๆ ยกเว้นชุดข้อมูล glass และ segment จะไม่มีชุดข้อมูลทดสอบ ในการทดลองเราจะทำการทดลองด้วยวิธี 5-fold cross validation [6] แล้วทำการทดสอบหาค่าเฉลี่ยของทุกชุด

ตารางที่ 2: ลักษณะของข้อมูลที่นำมาทดสอบ

Dataset	#training data	#test data	#class	#feature
Glass	214	5-fold	6	9
Satimage	4,435	2,000	6	36
Segment	2,310	5-fold	7	18
Shuttle	43,500	14,500	7	9
Vowel	528	462	11	10
Soybean	290	340	15	35
Letter	15,963	4,037	26	16
Isolet	6,238	1,559	26	617

และสำหรับการทดลองของวิธีการแตกครึ่งแบบสมดุล เนื่องจากต้องทำการแบ่งข้อมูลสอนออกเป็นข้อมูลสอนจริงและข้อมูลทดสอบความถูกต้องเพื่อใช้ในการหาค่าพารามิเตอร์ P และ R ก่อนทำการจำแนกกับข้อมูลทดสอบจริง ในการทดลองเราจึงทำการแบ่งข้อมูลสอนที่ได้มาออกเป็น 2 ส่วนด้วยวิธี 5-fold คือแบ่งข้อมูลสอนที่ได้มาออกเป็น 5 ชุด คือ a,b,c,d,e แล้วทำการทดลองใน 5 โฟลเดอร์คือ A,B,C,D,E โดยที่โฟลเดอร์ A จะมีชุดข้อมูล a เป็นข้อมูลทดสอบ

ความถูกต้องและมีชุดข้อมูล b,c,d,e เป็นข้อมูลสอนจริง ส่วนโฟลเดอร์ B จะมีชุดข้อมูล b เป็นข้อมูลทดสอบความถูกต้องและมีชุดข้อมูล a,c,d,e เป็นข้อมูลสอนจริง เป็นแบบนี้ตามลำดับไปจนถึงโฟลเดอร์ E ก็จะได้ข้อมูลสอนจริงและข้อมูลทดสอบความถูกต้องที่ถูกแบ่งออกมาจากข้อมูลสอนเป็น 5 รูปแบบใน 5 โฟลเดอร์ และจะทำการทดลองตามอัลกอริทึมของการแตกครึ่งแบบสมดุลในทั้ง 5 โฟลเดอร์ เพื่อหาค่าเฉลี่ยความถูกต้องและจำนวนครั้งของการจำแนกของทั้ง 5 โฟลเดอร์

4.2 ผลการทดลอง

ในหัวข้อนี้จะแสดงผลที่ได้จากการทดลองในข้อมูลทดลอง 8 ชุดข้อมูล โดยแบ่งการแสดงผลออกเป็น 2 ส่วนคือ จำนวนครั้งของการจำแนกและค่าความถูกต้องของการจำแนก โดยในส่วนของจำนวนครั้งของการจำแนกจะแสดงค่าที่ได้จากวิธีแตกครึ่งแบบสมดุลเปรียบเทียบกับค่าคาดหวังดีสุด (Expected) คือ $\log_2 K$ ครั้ง และค่าที่ได้จากวิธีอาร์เอดีเอจี วิธีเอดีเอจี และวิธีแมกซ์วิน ส่วนค่าความถูกต้องจะแสดงค่าที่ได้จากวิธีแตกครึ่งแบบสมดุลเปรียบเทียบกับวิธีอาร์เอ-ดีเอจี วิธีเอดีเอจี และวิธีแมกซ์วิน โดยแยกประเภทการแสดงออกเป็น Polynomial Kernel และ RBF Kernel โดยแสดงค่าที่ได้จากการเรียนรู้ของแต่ละ Kernel Degree ต่าง ๆ (d = Polynomial degree, c = RBF degree) โดยวิธีที่ได้ค่าความถูกต้องสูงที่สุดในแต่ละ Degree จะถูกแสดงเป็นตัวหนา และค่าที่สูงที่สุดในแต่ละ Kernel จะใส่เครื่องหมาย * กำกับไว้

สำหรับผลการทดลองของวิธีการแตกครึ่งแบบสมดุลที่แสดงให้บทนี้จะเป็นการแสดงผลในรูปของค่าเฉลี่ยที่ได้จาก 5 โฟลเดอร์ (A,B,C,D,E) ดังที่กล่าวในหัวข้อที่แล้ว โดยผลการทดลองโดยละเอียดและค่าพารามิเตอร์ P และ R ที่หาได้ในแต่ละโฟลเดอร์ จะได้แสดงไว้ในภาคผนวกท้ายเล่มต่อไป

4.2.1 **ชุดข้อมูล** Glass

ชุดข้อมูลนี้มีความซับซ้อนน้อยในการทดลอง เนื่องจากมีจำนวนข้อมูลตัวอย่างน้อยและมีจำนวนประเภทข้อมูลเพียง 6 ประเภท แต่เนื่องจากไม่มีข้อมูลทดสอบให้มาจึงต้องทำการทดลองด้วยวิธี 5-fold cross validation แล้วทำการทดสอบหาค่าเฉลี่ยของทุกชุด

ในตารางที่ 3 (ก) แสดงให้เห็นว่าวิธีการแตกครึ่งแบบสมดุลจะมีจำนวนครั้งของการจำแนกลดลงจากวิธีอาร์เอดีเอจีเพียงเล็กน้อย ไม่สามารถลดลงเท่าค่าคาดหวังได้ โดยลดลงเหลือประมาณ 80% ของ วิธีอาร์เอดีเอจี

ตารางที่ 3 (ก): เปรียบเทียบจำนวนครั้งของการจำแนกในชุดข้อมูล Glass

Polynomial d	Balanced Dichotomization	Expected ($\log_2 k$)	RADAG, ADAG (k-1)	Max Wins (k(k-1)/2)
2	4.186	2.584	5	15
3	3.896	2.584	5	15
4	4.026	2.584	5	15
5	4.105	2.584	5	15
6	4.086	2.584	5	15
7	4.086	2.584	5	15
8	4.472	2.584	5	15
RBF c	**Balanced Dichotomization**	**Expected ($\log_2 k$)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
0.01	4.386	2.584	5	15
0.02	4.319	2.584	5	15
0.03	4.282	2.584	5	15
0.04	4.250	2.584	5	15
0.05	4.668	2.584	5	15
0.06	4.301	2.584	5	15
0.07	4.301	2.584	5	15
0.08	4.361	2.584	5	15
0.09	4.361	2.584	5	15
0.1	4.361	2.584	5	15

ตารางที่ 3 (ข): เปรียบเทียบค่าความถูกต้องของการจำแนกในชุดข้อมูล Glass

Polynomial d	Balanced Dichotomization	RADAG	ADAG	Max Wins
2	**71.528***	**71.528***	71.130	71.078
3	**69.646**	68.716	68.793	68.989
4	**69.646**	68.716	69.405	69.598
5	69.634	69.634	69.942	**70.173**
6	69.158	69.158	69.306	**69.553**
7	67.298	67.298	67.566	**67.902**
8	**68.671**	68.217	68.250	68.463
RBF c	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
0.01	**72.436**	**72.436**	71.958	71.871
0.02	**71.971**	71.506	71.060	70.918
0.03	**71.971**	71.506	71.060	70.918
0.04	**71.971**	**71.971**	71.789	70.616
0.05	71.052	**71.517**	70.123	69.943
0.06	73.378	**73.843**	72.304	72.088
0.07	**73.843**	**73.843**	72.615	72.385
0.08	**74.308**	**74.308**	72.750	72.506
0.09	**74.784***	74.536	71.943	73.328
0.1	72.913	**73.378**	71.688	71.367

ส่วนค่าความถูกต้องในตารางที่ 3 (ข) แสดงให้เห็นว่าวิธีแตกครึ่งแบบสมดุลมีค่าความถูกต้องใกล้เคียงกับวิธีอาร์เอดีเอจี ใน RBF kernel และใกล้เคียงกับวิธีแมกซ์วิน ใน Polynomial kernel

4.2.2 **ชุดข้อมูล** Satimage

ชุดข้อมูลนี้จำนวนประเภทข้อมูลเพียง 6 ประเภทเหมือนชุดข้อมูล Glass แต่ว่ามีจำนวนข้อมูลตัวอย่างมาก ทำให้ใช้เวลาในการสร้างระนาบนานกว่าชุดข้อมูล Glass และเมื่อนำมาจำแนกโดยวิธีการแตกครึ่งแบบสมดุล ปรากฏว่าหาระนาบที่สมดุลที่สุดได้ยากกว่าชุดข้อมูล Glass ทำให้จำนวนครั้งของการจำแนกสูงกว่าของชุดข้อมูล Glass เล็กน้อยแต่ยังคงน้อยกว่าวิธีอาร์เอดีเอจี ดังแสดงในตารางที่ 4 (ก)

ตารางที่ 4 (ก): **เปรียบเทียบจำนวนครั้งของการจำแนกในชุดข้อมูล** Satimage

Polynomial d	**Balanced Dichotomization**	**Expected ($\log_2 k$)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
2	4.396	2.584	5	15
3	4.445	2.584	5	15
4	4.480	2.584	5	15
5	4.603	2.584	5	15
6	4.605	2.584	5	15
7	4.387	2.584	5	15
8	4.384	2.584	5	15
RBF c	**Balanced Dichotomization**	**Expected ($\log_2 k$)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
0.5	4.512	2.584	5	15
1.0	4.498	2.584	5	15
1.5	5.000	2.584	5	15
2.0	5.000	2.584	5	15
3.0	5.000	2.584	5	15
4.0	4.262	2.584	5	15
5.0	4.310	2.584	5	15

ค่าความถูกต้องในตารางที่ 4 (ข) แสดงให้เห็นว่าวิธีการแตกครึ่งแบบสมดุล มีค่าความถูกต้องสูงที่สุดใน Polynomial kernel และมีค่าใกล้เคียงกับวิธีอาร์เอดีเอจีใน RBF kernel

ตารางที่ 4 (ข): เปรียบเทียบค่าความถูกต้องของการจำแนกในชุดข้อมูล Satimage

Polynomial d	Balanced Dichotomization	RADAG	ADAG	Max Wins
2	87.450	**87.500**	86.990	87.152
3	**87.600**	87.400	87.215	87.248
4	**87.900**	87.750	87.560	87.545
5	**88.750**	88.550	88.237	88.198
6	**88.950***	88.800	88.438	88.453
7	**88.500**	88.350	87.938	87.957
8	**88.200**	88.050	87.731	87.757
RBF c	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
0.5	**89.400**	89.250	89.079	89.123
1.0	**90.450**	**90.450**	90.333	90.340
1.5	90.950	**91.000**	90.821	90.834
2.0	91.550	**91.600**	91.533	91.542
3.0	91.950	91.950	91.964	**91.984**
4.0	**92.100***	91.800	91.801	91.820
5.0	**91.800**	91.450	91.482	91.496

4.2.3 ชุดข้อมูล Segment

ชุดข้อมูลนี้มีจำนวนประเภทข้อมูล 7 ประเภท แต่ไม่มีชุดข้อมูลทดสอบเหมือนชุดข้อมูล glass จึงต้องทำทดลองด้วยวิธี 5-fold cross validation เช่นกัน

ตารางที่ 5 (ก): เปรียบเทียบจำนวนครั้งของการจำแนกในชุดข้อมูล Segment

Polynomial d	Balanced Dichotomization	Expected ($\log_2 k$)	RADAG, ADAG (k-1)	Max Wins (k(k-1)/2)
2	4.913	2.807	6	21
3	5.065	2.807	6	21
4	5.179	2.807	6	21
5	4.606	2.807	6	21
6	4.463	2.807	6	21
7	4.468	2.807	6	21
8	4.501	2.807	6	21
RBF c	**Balanced Dichotomization**	**Expected ($\log_2 k$)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
0.5	4.444	2.807	6	21
0.6	4.620	2.807	6	21
0.7	4.297	2.807	6	21
0.8	4.297	2.807	6	21
0.9	4.757	2.807	6	21
1.0	4.808	2.807	6	21
1.5	4.320	2.807	6	21

ซึ่งชุดข้อมูลนี้มีจำนวนข้อมูลตัวอย่างมากใกล้เคียงกับชุดข้อมูล Satimage แต่เมื่อนำมาทดลอง ปรากฏว่าหาระนาบสมดุลได้ดีกว่าชุดข้อมูล Glass และ Segment จำนวนครั้งของการจำแนกลดลงอยู่ที่ประมาณ 75% ของวิธีอาร์เอดีเอจี ดังตารางที่ 5 (ก)

ตารางที่ 5 (ข): **เปรียบเทียบค่าความถูกต้องของการจำแนกในชุดข้อมูล** Segment

Polynomial d	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
2	**96.926**	96.623	96.519	96.598
3	**96.710**	96.537	96.310	96.324
4	**96.926**	96.840	96.621	96.631
5	**97.013**	96.753	96.539	96.556
6	**97.229**	96.710	96.581	96.567
7	**97.273**	96.364	96.357	96.349
8	**97.532***	96.364	96.254	96.238
RBF c	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
0.5	97.056	**97.230**	97.104	97.111
0.6	97.056	**97.402**	97.250	97.254
0.7	97.229	**97.446***	97.290	97.302
0.8	97.229	**97.316**	97.151	97.155
0.9	97.186	**97.273**	97.234	97.254
1.0	97.056	**97.186**	97.139	97.135
1.5	**97.056**	**97.056**	97.042	97.045

ค่าความถูกต้องในตารางที่ 5 (ข) แสดงให้เห็นว่าวิธีแตกครึ่งแบบสมดุล ให้ค่าความถูกต้องสูงที่สุดใน Polynomial kernel แต่ให้ค่าความถูกต้องน้อยกว่าวิธีอาร์เอดีเอจีใน RBF kernel

4.2.4 **ชุดข้อมูล** Shuttle

ชุดข้อมูลนี้จำนวนประเภทข้อมูล 7 ประเภทเหมือนชุดข้อมูล Segment แต่มีจำนวนข้อมูลตัวอย่างมากที่สุดจากทุกชุดข้อมูลที่นำมาทดสอบ ในการสอนจึงใช้เวลาในการสร้างระนาบนานที่สุด แต่เมื่อนำมาจำแนกข้อมูล ปรากฏว่าจำนวนครั้งของการจำแนกลดลงได้เป็นค่าที่ดีที่สุดคือน้อยกว่า $log_2(k)$ ครั้ง ที่เป็นเช่นนี้เพราะว่าสามารถหาระนาบที่แบ่งข้อมูลได้ใกล้เคียงระนาบสมดุลที่สุดในแต่ละรอบ และถึงแม้ว่าจำนวนข้อมูลตัวอย่างของ Shuttle จะมีจำนวนมาก แต่ตกอยู่ในด้านที่มีความลึกน้อยกว่าเป็นส่วนใหญ่ ทำให้จำนวนครั้งของการจำแนกเฉลี่ยลดลงได้ดีสุดตามอัลกอริทึม ดังตารางที่ 6 (ก)

ตารางที่ 6 (ก): เปรียบเทียบจำนวนครั้งของการจำแนกในชุดข้อมูล Shuttle

Polynomial d	**Balanced Dichotomization**	**Expected ($\log_2 k$)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
2	2.383	2.807	6	21
3	2.458	2.807	6	21
4	2.546	2.807	6	21
5	2.588	2.807	6	21
6	2.392	2.807	6	21
7	2.464	2.807	6	21
8	2.546	2.807	6	21
RBF c	**Balanced Dichotomization**	**Expected ($\log_2 k$)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
0.5	2.306	2.807	6	21
1.0	2.496	2.807	6	21
1.5	2.500	2.807	6	21
2.0	2.495	2.807	6	21
3.0	2.497	2.807	6	21
4.0	2.586	2.807	6	21
5.0	2.586	2.807	6	21

ตารางที่ 6 (ข): เปรียบเทียบค่าความถูกต้องของการจำแนกในชุดข้อมูล Shuttle

Polynomial d	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
2	**99.841**	99.834	99.832	99.834
3	**99.869**	**99.869**	99.868	**99.869**
4	**99.869**	**99.869**	99.867	**99.869**
5	**99.890**	**99.890**	99.889	**99.890**
6	**99.903**	**99.903**	**99.903**	**99.903**
7	**99.917**	**99.917**	**99.917**	**99.917**
8	**99.924***	**99.924***	**99.924***	**99.924***
RBF c	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
0.5	**99.834**	**99.834**	**99.834**	**99.834**
1.0	**99.862**	**99.862**	**99.862**	**99.862**
1.5	99.876	99.876	**99.879**	**99.879**
2.0	**99.883**	**99.883**	**99.883**	**99.883**
3.0	**99.897***	**99.897***	**99.897***	**99.897***
4.0	**99.897***	99.890	99.895	**99.897***
5.0	**99.897***	99.890	99.888	99.892

ค่าความถูกต้องในตารางที่ 6 (ข) แสดงให้เห็นว่าว่าข้อมูลชุดนี้ให้ความถูกต้องของการจำแนกใกล้เคียงกันหมดทั้ง 4 วิธี ทั้งใน Polynomial kernel และ RBF kernel

4.2.5 **ชุดข้อมูล** Vowel

ชุดข้อมูลนี้มีจำนวนประเภทข้อมูล 11 ประเภท แต่มีจำนวนข้อมูลตัวอย่างไม่มากนัก ทำให้ใช้เวลาในการสอนและการจำแนกเพียงเล็กน้อยใกล้เคียงกับชุดข้อมูล Glass และหาระนาบที่สมดุลที่สุดได้ดี ทำให้ลดจำนวนครั้งของการจำแนกได้เป็นครึ่งหนึ่งของวิธีอาร์เอดีเอจี ดังตารางที่ 7 (ก)

ตารางที่ 7 (ก): **เปรียบเทียบจำนวนครั้งของการจำแนกในชุดข้อมูล** Vowel

Polynomial d	**Balanced Dichotomization**	**Expected (log_2k)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
2	5.091	3.459	10	55
3	6.058	3.459	10	55
4	5.294	3.459	10	55
5	4.459	3.459	10	55
6	6.450	3.459	10	55
7	6.610	3.459	10	55
8	5.985	3.459	10	55
RBF c	**Balanced Dichotomization**	**Expected (log_2k)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
0.1	4.671	3.459	10	55
0.2	4.591	3.459	10	55
0.3	4.688	3.459	10	55
0.4	4.481	3.459	10	55
0.5	4.444	3.459	10	55
1.0	5.221	3.459	10	55
1.5	6.260	3.459	10	55
2.0	6.576	3.459	10	55

ค่าความถูกต้องในตารางที่ 7 (ข) แสดงให้เห็นว่าวิธีแตกครึ่งแบบสมดุลให้ค่าความถูกต้องสูงที่สุดทั้งใน Polynomial kernel และ ใน RBF kernel ในชุดข้อมูลนี้

ตารางที่ 7 (ข): **เปรียบเทียบค่าความถูกต้องของการจำแนกในชุดข้อมูล** Vowel

Polynomial d	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
2	**67.532***	63.420	63.870	63.918
3	**66.667**	64.719	64.298	64.329
4	61.472	62.771	62.660	**62.773**
5	**61.905**	60.606	60.405	60.346
6	57.143	**58.658**	58.559	58.649
7	**57.359**	56.710	56.721	56.725
8	**57.143**	55.628	55.654	55.647

RBF c	Balanced Dichotomization	RADAG	ADAG	Max Wins
0.1	**64.285**	62.338	61.342	61.169
0.2	**68.831***	67.100	65.585	65.340
0.3	**68.398**	65.801	65.402	65.203
0.4	**65.584**	64.935	63.924	64.108
0.5	64.586	**64.935**	63.677	63.677
1.0	**61.905**	61.255	61.064	61.214
1.5	59.957	**61.472**	60.902	60.833
2.0	59.091	**60.606**	60.403	60.290

4.2.6 **ชุดข้อมูล** Soybean

ชุดข้อมูลนี้มีจำนวนประเภทข้อมูลสูงกว่าชุดข้อมูล Vowel คือ 15 ประเภท แต่ค่อนข้างมีความคล้ายคลึงกัน คือมีจำนวนข้อมูลตัวอย่างน้อย และหาระนาบสมดุลที่สุดได้ดี ทำให้ลดจำนวนครั้งของการจำแนกได้เกือบครึ่งหนึ่งของวิธีอาร์เอดีเอจี ใกล้เคียงกับผลที่ได้จากชุดข้อมูล Vowel ดังตารางที่ 8 (ก)

ตารางที่ 8 (ก): **เปรียบเทียบจำนวนครั้งของการจำแนกในชุดข้อมูล** Soybean

Polynomial d	Balanced Dichotomization	Expected ($\log_2 k$)	RADAG, ADAG (k-1)	Max Wins (k(k-1)/2)
2	6.376	3.906	14	105
3	6.153	3.906	14	105
4	7.597	3.906	14	105
5	7.935	3.906	14	105
6	7.926	3.906	14	105
7	7.779	3.906	14	105
8	8.191	3.906	14	105
RBF c	**Balanced Dichotomization**	**Expected ($\log_2 k$)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
0.04	6.135	3.906	14	105
0.05	6.453	3.906	14	105
0.06	7.159	3.906	14	105
0.07	6.502	3.906	14	105
0.08	6.583	3.906	14	105
0.09	7.089	3.906	14	105
0.1	7.321	3.906	14	105
0.2	7.968	3.906	14	105

ส่วนของค่าความถูกต้องจะแตกต่างกันตรงที่ในชุดข้อมูล Soybean นี้วิธีแตกครึ่งแบบสมดุลให้ค่าความถูกต้องน้อยกว่าวิธีอาร์เอดีเอจีทั้งใน Polynomial kernel และ ใน RBF kernel แต่ยังมีค่าสูงกว่าวิธีแมกซ์วินและวิธีเอดีเอจี ดังแสดงในตารางที่ 8 (ข)

ตารางที่ 8 (ข): เปรียบเทียบค่าความถูกต้องของการจำแนกในชุดข้อมูล Soybean

Polynomial d	Balanced Dichotomization	RADAG	ADAG	Max Wins
2	90.294	**90.882**	89.645	89.768
3	90.882	**91.176***	90.407	90.471
4	**90.588**	**90.588**	90.112	89.968
5	90.588	**90.882**	90.440	90.353
6	**89.706**	**89.706**	89.584	89.506
7	**90.000**	88.824	88.688	88.668
8	87.705	**87.941**	87.805	87.785
RBF c	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
0.04	**90.294**	**90.294**	89.665	89.738
0.05	90.588	**90.882***	90.338	90.379
0.06	90.294	**90.588**	90.097	90.068
0.07	90.588	**90.882***	90.388	90.362
0.08	**90.588**	**90.588**	90.413	90.468
0.09	90.294	**90.588**	90.399	90.468
0.1	90.000	**90.294**	90.092	90.174
0.2	86.471	**86.765**	86.657	86.682

4.2.7 ชุดข้อมูล Letter

ชุดข้อมูลนี้มีจำนวนประเภทข้อมูลมากที่สุดใน 8 ชุดข้อมูลที่นำมาทดสอบคือ 26 ประเภท และมีจำนวนข้อมูลตัวอย่างมากเป็นอันดับที่ 2 รองจากชุดข้อมูล Shuttle และเมื่อนำมา

ตารางที่ 9 (ก): เปรียบเทียบจำนวนครั้งของการจำแนกในชุดข้อมูล Letter

Polynomial d	Balanced Dichotomization	Expected (log_2k)	RADAG, ADAG (k-1)	Max Wins (k(k-1)/2)
2	22.005	4.700	25	325
3	22.875	4.700	25	325
4	22.667	4.700	25	325
5	22.508	4.700	25	325
6	22.031	4.700	25	325
7	22.397	4.700	25	325
8	22.304	4.700	25	325
RBF c	**Balanced Dichotomization**	**Expected (log_2k)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
0.5	22.364	4.700	25	325
1.0	22.464	4.700	25	325
1.5	22.915	4.700	25	325
2.0	23.323	4.700	25	325
3.0	23.481	4.700	25	325
4.0	23.380	4.700	25	325
5.0	22.890	4.700	25	325

ทำการจำแนกข้อมูล ปรากฏว่าชุดข้อมูล Letter นี้ ลดของจำนวนครั้งของการจำแนกลงได้น้อยที่สุดใน 8 ชุดข้อมูลเนื่องจากไม่สามารถหาระนาบสมดุลที่ดีในการจำแนกได้ ทำให้ใช้เวลาในการจำแนกนาน ดังค่าในตารางที่ 9 (ก) ส่งผลให้ใช้เวลาในการคำนวณหาค่าพารามิเตอร์ P และ R มากตามไปด้วย

ตารางที่ 9 (ข): เปรียบเทียบค่าความถูกต้องของการจำแนกในชุดข้อมูล Letter

Polynomial d	Balanced Dichotomization	RADAG	ADAG	Max Wins
2	95.120	**95.541**	95.112	95.522
3	96.161	**96.185***	95.985	96.125
4	95.838	96.086	95.890	**96.120**
5	95.764	**95.789**	95.525	94.622
6	95.145	**95.566**	95.342	95.551
7	94.600	**95.021**	94.786	94.975
8	93.733	**94.377**	93.828	94.072
RBF c	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
0.5	96.743	**96.879**	96.652	96.674
1.0	97.374	**97.523**	97.379	97.427
1.5	97.498	**97.677**	97.585	97.629
2.0	97.597	**97.795**	97.632	97.661
3.0	97.894	**97.969***	97.907	97.918
4.0	97.272	**97.845**	97.798	97.815
5.0	97.659	**97.672**	97.673	97.689

ส่วนค่าความถูกต้องในตารางที่ 9 (ข) แสดงให้เห็นว่าชุดข้อมูล Letter ให้ค่าความถูกต้องของวิธีแตกครึ่งแบบสมดุลไม่น้อยกว่าวิธีอาร์เอดีเอจีทั้งใน Polynomial kernel และ RBF kernel

4.2.8 ชุดข้อมูล Isolet

ชุดข้อมูลนี้มีจำนวนประเภทข้อมูล 26 ประเภทเท่ากับชุดข้อมูล Letter และมีจำนวนข้อมูลตัวอย่างมากเป็นอันดับ 3 รองจาก Letter แต่หาระนาบที่สมดุลได้ดีกว่า Letter จึงลดจำนวนครั้งของการจำแนกได้เหลือครึ่งหนึ่งของวิธีอาร์เอดีเอจีเหมือนกับชุดข้อมูล Vowel และ Soybean ดังตารางที่ 10 (ก) แต่เนื่องจากเป็นข้อมูลที่มีจำนวนคุณสมบัติสูงที่สุด 617 คุณสมบัติ ทำให้ใช้เวลาในการจำแนกกับตัวอย่างแต่ละตัวนาน ถึงแม้จะลดจำนวนครั้งของการจำแนกลงได้มากก็ยังใช้เวลาในการจำแนกสูงอยู่

ตารางที่ 10 (ก): เปรียบเทียบจำนวนครั้งของการจำแนกในชุดข้อมูล Isolet

Polynomial d	**Balanced Dichotomization**	**Expected (log_2k)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
2	12.358	4.700	25	325
3	12.687	4.700	25	325
4	12.367	4.700	25	325
5	12.793	4.700	25	325
6	12.555	4.700	25	325
7	12.803	4.700	25	325
8	12.174	4.700	25	325
RBF c	**Balanced Dichotomization**	**Expected (log_2k)**	**RADAG, ADAG (k-1)**	**Max Wins (k(k-1)/2)**
0.001	12.483	4.700	25	325
0.002	12.325	4.700	25	325
0.003	12.195	4.700	25	325
0.004	12.106	4.700	25	325
0.005	12.150	4.700	25	325
0.01	12.465	4.700	25	325
0.02	13.063	4.700	25	325
0.03	11.722	4.700	25	325

ตารางที่ 10 (ข): เปรียบเทียบค่าความถูกต้องของการจำแนกในชุดข้อมูล Isolet

Polynomial d	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
2	96.183	**96.603**	96.556	96.586
3	96.408	**97.051***	97.027	97.040
4	96.376	**97.051***	96.987	97.024
5	96.376	96.603	96.659	**96.695**
6	96.536	**96.667**	**96.667**	96.666
7	**96.280**	96.154	96.137	96.133
8	**95.606**	95.577	95.493	95.488
RBF c	**Balanced Dichotomization**	**RADAG**	**ADAG**	**Max Wins**
0.001	95.799	**96.667**	96.558	96.554
0.002	95.959	**96.731**	96.602	96.619
0.003	96.216	**96.923**	96.895	96.889
0.004	96.440	**96.795**	96.742	96.726
0.005	96.248	**96.795**	96.698	96.681
0.01	96.408	**96.987***	96.932	96.916
0.02	**96.793**	96.731	96.738	96.731
0.03	**96.248**	95.769	95.683	95.680

ในส่วนของค่าความถูกต้องในชุดข้อมูล Isolet นี้ ก็ให้ค่าความถูกต้องของวิธีอาร์เอดีเอจีสูงกว่าวิธีแตกครึ่งแบบสมดุลทั้งใน Polynomial kernel และ ใน RBF kernel ดังตารางที่ 10 (ข)

4.3 เวลาที่ใช้ในการสอน

ปัญหาหนึ่งที่พบในการทดลองคือ เวลาที่ใช้ในการสอน แม้ว่าเวลาที่ใช้ในการจำแนกข้อมูลของวิธีแตกครึ่งแบบสมดุลจะลดลงตามจำนวนครั้งของการจำแนก แต่เวลาที่ใช้ในการสอนจะเพิ่มขึ้นจากวิธีการจำแนกแบบหนึ่งต่อหนึ่งแบบอื่นพอสมควร เนื่องจากในการจำแนกแบบหนึ่งต่อหนึ่งแบบอื่น ๆ นั้น เวลาที่ใช้ในการสอนจะมีเพียงเวลาที่ใช้สร้างระนาบเท่านั้น แต่วิธีแตกครึ่งแบบสมดุล จะต้องใช้เวลาในการค้นหาตำแหน่งของระนาบและค้นหาค่าพารามิเตอร์ P และ R ที่ให้ค่าดีที่สุดในข้อมูลทดสอบความถูกต้องด้วย ทำให้มีเวลาที่ใช้ในการสอนรวมเพิ่มขึ้นจากวิธีอื่น

ตารางที่ 11: เปรียบเทียบเวลาที่ใช้ในการสอนของวิธีแตกครึ่งแบบสมดุลกับวิธีแมกซ์วิน

Dataset	**สร้างระนาบ**	**ค้นหาตำแหน่งของระนาบ**	**คำนวณหาค่า P, R**	Total Time
*Glass				
Max Wins	44.89	0.00	0.00	44.89
Balanced	13.32	1.46	18.62	33.40
*Satimage				
Max Wins	2517.84	0.00	0.00	2517.84
Balanced	1339.91	34.17	216.52	1590.60
*Segment				
Max Wins	233.50	0.00	0.00	233.50
Balanced	198.31	11.23	111.00	320.54
*Shuttle				
Max Wins	4798.23	0.00	0.00	4798.23
Balanced	2455.01	181.34	2267.10	4903.45
*Vowel				
Max Wins	29.05	0.00	0.00	29.05
Balanced	28.61	10.73	52.73	92.07
*Soybean				
Max Wins	57.42	0.00	0.00	57.42
Balanced	52.63	18.11	55.96	126.70
*Letter				
Max Wins	1128.88	0.00	0.00	1128.88
Balanced	795.20	1434.12	9781.52	12010.84
*Isolet				
Max Wins	2506.98	0.00	0.00	2506.98
Balanced	2077.53	15806.34	2324.17	20208.04

ในตารางที่ 11 แสดงเวลาที่ใช้ในการสอนหน่วยเป็นวินาทีของวิธีการแตกครึ่งแบบสมดุล (Balanced) เปรียบเทียบกับวิธีการจำแนกแบบหนึ่งต่อหนึ่งแบบอื่น ๆ เช่น วิธีแมกซ์วิน (วิธีแมกซ์วิน วิธีเอดีเอจีและวิธีอาร์เอดีเอจีใช้เวลาในการสอนเท่ากัน คือใช้เวลาในการสร้างระนาบ

k(k-1)/2 ตัว แล้วนำระนาบไปใช้ได้ทันที ในที่นี้จึงยกตัวอย่างเวลาในการสอนของวิธีแมกซ์วินเพียงอย่างเดียว) โดยค่าที่ได้แสดงให้เห็นว่าวิธีแมกซ์วินใช้เวลาในการสอนเพื่อทำการสร้างระนาบหลายมิติอย่างเดียว แต่วิธีแตกครึ่งแบบสมดุลต้องใช้เวลาทั้งใน 3 ส่วน แต่วิธีแตกครึ่งแบบสมดุลจะใช้เวลาในการสร้างระนาบน้อยกว่าวิธีแมกซ์วิน เนื่องจากมีการแบ่งข้อมูลสอนบางส่วนออกไปเป็นข้อมูลทดสอบความถูกต้อง ทำให้มีข้อมูลสอนจริงที่ใช้สร้างระนาบน้อยลง จึงสร้างได้เร็วขึ้น แต่จะไปเสียเวลาเพิ่มขึ้นในการค้นหาตำแหน่งของระนาบและการคำนวณหาค่า P และ R แทน ซึ่งส่วนใหญ่จะใช้เวลาใน 2 ส่วนนี้เป็น 1 เท่าของเวลาในการสร้างระนาบ

จากตารางแสดงให้เห็นว่าเวลาที่ใช้ในการสอนของชุดข้อมูล Glass, Satimage, Segment และ Shuttle มีค่าใกล้เคียงกันทั้งวิธีแมกซ์วินและวิธีแตกครึ่งแบบสมดุล เพราะว่าวิธีแตกครึ่งแบบสมดุลใช้เวลาในการสร้างระนาบน้อยกว่าวิธีแมกซ์วินมาก ส่วนชุดข้อมูล Vowel และ Soybean จะใช้เวลาในการสร้างระนาบใกล้เคียงกัน ทำให้มีเวลาที่ใช้สอนรวมสูงกว่าวิธีแมกซ์วินเล็กน้อยตามเวลาที่ใช้ในอีก 2 ส่วนที่เหลือ ส่วนชุดข้อมูล Letter และ Isolet จะใช้เวลาในการสอนรวมสูงกว่าวิธีแมกซ์วินค่อนข้างมาก เนื่องจากวิธีแตกครึ่งแบบสมดุลลดจำนวนครั้งของการจำแนกได้น้อยในชุดข้อมูล Letter จึงใช้เวลาในการจำแนกเพื่อหาค่า P และ R นาน ส่วนชุดข้อมูล Isolet มีจำนวนคุณสมบัติของข้อมูลเป็นจำนวนมาก จึงใช้เวลาในการคำนวณหาตำแหน่งของระนาบนาน แต่เวลาที่ใช้ในการคำนวณค่า P และ R ยังอยู่ในช่วง 1 เท่าของเวลาที่ใช้สร้างระนาบ

จะเห็นได้ว่า วิธีการแตกครึ่งแบบสมดุล จะมีเวลาที่ใช้ในการสอนแปรผันตามเวลาที่ใช้ในการคำนวณหาค่า ค่า P และ R ถ้าเราสามารถลดเวลาในส่วนนี้ได้ก็จะทำให้วิธีแตกครึ่งแบบสมดุลมีเวลาที่ใช้ในการสอนใกล้เคียงกับวิธีแมกซ์วินและวิธีแบบหนึ่งต่อหนึ่งแบบอื่น ๆ

4.4 สรุปผลการทดลอง

จากการทดลองใน 8 ชุดข้อมูล แสดงให้เห็นว่าในทุกชุดข้อมูล วิธีการแตกครึ่งแบบสมดุลจะมีจำนวนครั้งของการจำแนกลดลงจากค่า $k-1$ ครั้งจากวิธีอาร์เอดีเอจีและวิธีเอดีเอจีเสมอ โดยที่ชุดข้อมูล Vowel, Soybean, Isolet สามารถลดจำนวนครั้งได้ถึงครึ่งหนึ่งของวิธีอาร์เอดีเอจี และชุดข้อมูล Shuttle เป็นกรณีที่ดีที่สุดสามารถลดจำนวนครั้งของการจำแนกได้เหลือน้อยกว่าค่า $log_2(k)$ อีกด้วย ดังแสดงไว้ในตารางที่ 12

แต่กับผลที่ได้ในชุดข้อมูล Glass, Satimage, Segment และ Letter แสดงให้เห็นปัญหาว่า วิธีการแตกครึ่งแบบสมดุลจะให้ค่าจำนวนครั้งของการจำแนกลดลงจากวิธีอาร์เอดีเอจี

ไม่ดีนัก ถ้าไม่มีระนาบที่ตัดข้อมูลได้มากตามที่คาดหวัง แม้จะมีการกำหนดพารามิเตอร์ค่าร้อยละของการตัดเล็มมาใช้ ก็ยังเพิ่มประเภทข้อมูลที่ถูกตัดได้เพียงเล็กน้อย

ตารางที่ 12: สรุปค่าเปรียบเทียบค่าจำนวนครั้งของการจำแนกทั้ง 8 ชุดข้อมูล

Dataset	Expected Value ($\log_2 k$)	Polynomial Kernel		RBF Kernel	
		RADAG, ADAG (k-1)	Balanced Dichotomization	**RADAG, ADAG** (k-1)	Balanced Dichotomization
Glass	2.584	5	4.186	5	4.361
Satimage	2.584	5	4.605	5	4.262
Segment	2.807	6	4.501	6	4.297
Shuttle	2.807	6	**2.546**	6	**2.497**
Vowel	3.459	10	5.091	10	4.591
Soybean	3.906	14	6.153	14	6.453
Letter	4.700	25	22.875	25	23.481
Isolet	4.700	25	12.555	25	13.063

ตารางที่ 13 แสดงผลสรุปการเปรียบเทียบค่าความถูกต้องของการจำแนกของวิธีการแตกครึ่งแบบสมดุล วิธีอาร์เอดีเอจี วิธีเอดีเอจีและวิธีแมกซ์วิน ใน Polynomial Kernel โดยแสดงค่าความถูกต้องที่ดีที่สุดที่แต่ละวิธีทำได้ในแต่ละชุดข้อมูล และวิธีที่ให้ค่าความถูกต้องสูงที่สุดในชุดข้อมูลนั้นจะแสดงเป็นตัวหนา ซึ่งค่าที่ได้แสดงให้เห็นว่าวิธีการแตกครึ่งแบบสมดุลมีค่าความถูกต้องสูงที่สุดใน 5 ชุดข้อมูลคือ Glass, Satimage, Segment, Shuttle และ Vowel

ตารางที่ 13: ผลสรุปการเปรียบเทียบค่าความถูกต้อง Polynomial kernel

Dataset	d	Balanced	d	RADAG	d	ADAG	D	Max Wins
Glass	2	**71.528**	2	**71.528**	2	71.130	2	71.078
Satimage	6	**88.950**	6	88.900	6	88.438	6	88.453
Segment	8	**97.532**	4	96.840	4	96.621	4	96.631
Shuttle	8	**99.924**	8	**99.924**	8	**99.924**	8	**99.924**
Vowel	2	**67.532**	3	64.719	3	64.298	3	64.329
Soybean	5	90.882	3	**91.176**	3	90.440	3	90.471
Letter	3	96.161	3	**96.185**	3	95.985	3	96.125
Isolet	6	96.536	3	**97.051**	3	97.027	3	97.040

ตารางที่ 14 แสดงผลสรุปการเปรียบเทียบค่าความถูกต้องของการจำแนกของวิธีการแตกครึ่งแบบสมดุล วิธีอาร์เอดีเอจี วิธีเอดีเอจีและวิธีแมกซ์วิน ใน RBF Kernel โดยแสดงค่าความถูกต้องที่ดีที่สุดที่แต่ละวิธีทำได้ในแต่ละชุดข้อมูล ซึ่งค่าที่ได้แสดงให้เห็นว่าวิธีการแตกครึ่งแบบสมดุลมีค่าความถูกต้องสูงที่สุดใน 4 ชุดข้อมูลคือ Glass, Satimage, Shuttle และ Vowel

ตารางที่ 14: ผลสรุปการเปรียบเทียบค่าความถูกต้อง RBF kernel

Dataset	c	Balanced	c	RADAG	c	ADAG	c	Max Wins
Glass	0.09	**74.784**	0.09	74.536	0.09	72.750	0.09	73.328
Satimage	4.0	**92.100**	3.0	91.950	3.0	91.964	3.0	91.984
Segment	0.8	97.229	0.7	**97.446**	0.7	97.290	0.7	97.302
Shuttle	3.0	**99.897**	3.0	**99.897**	3.0	**99.897**	3.0	**99.897**
Vowel	0.2	**68.831**	0.3	67.100	0.2	65.585	0.2	65.340
Soybean	0.07	90.588	0.07	**90.882**	0.07	90.413	0.08	90.468
Letter	3.0	97.894	3.0	**97.969**	3.0	97.907	3.0	97.918
Isolet	0.02	96.793	0.01	**96.987**	0.01	96.932	0.01	96.916

โดยสรุปแล้วผลการทดลองแสดงให้เห็นว่า วิธีการจำแนกโดยการแตกครึ่งแบบสมดุล จะมีจำนวนครั้งของการจำแนกน้อยกว่าวิธีอาร์เอดีเอจีและวิธีแบบหนึ่งต่อหนึ่งแบบอื่นๆ โดยในกรณีดีสุด จะมีจำนวนครั้งของการจำแนกลดลงเหลือ $log_2(k)$ ครั้ง ในขณะที่ยังคงให้ค่าความถูกต้องใกล้เคียงกับวิธีจำแนกแบบหนึ่งต่อหนึ่งแบบอื่นๆ อยู่

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในงานวิจัยนี้ได้นำเสนอวิธีการใหม่สำหรับการจำแนกข้อมูลของซัพพอร์ตเวกเตอร์แมชชีนแบบหลายประเภทมีชื่อว่า วิธีการแตกครึ่งแบบสมดุล (Balanced Dichotomization) โดยมีหลักการอยู่ที่การค้นหาระนาบแบบหนึ่งต่อหนึ่งที่สามารถแบ่งข้อมูลได้ในตำแหน่งที่สมดุลกันที่สุดของข้อมูล ทำให้ลดจำนวนครั้งของการจำแนกลงจากวิธีการจำแนกแบบหนึ่งต่อหนึ่งแบบเดิม เช่น วิธีแมกซ์วิน วิธีอาร์เอดีเอจีและวิธีเอดีเอจีได้ ในขณะที่ยังคงให้ค่าความถูกต้องใกล้เคียงกัน โดยลดจำนวนครั้งของการจำแนกทั้งหมดได้ดีที่สุดเหลือ $log_2(k)$ ครั้ง

แต่วิธีการแตกครึ่งแบบสมดุลมีข้อจำกัดอยู่ที่เป็นการนำระนาบของการจำแนกแบบหนึ่งต่อหนึ่งมาใช้ ซึ่งระนาบเหล่านั้นอาจไม่มีระนาบที่อยู่ในตำแหน่งที่แบ่งข้อมูลทั้ง 2 ด้านได้เหลือครึ่งหนึ่งตามที่คาดไว้ เพราะตำแหน่งของระนาบไม่สามารถเปลี่ยนแปลงได้

ปัญหาต่อมาของการนำระนาบแบบหนึ่งต่อหนึ่งมาใช้ คือเป็นการแบ่งข้อมูลโดยใช้ระนาบเพียง 1 ระนาบแบ่งข้อมูลเป็น 2 ด้านคือ ซ้ายและขวา ความถูกต้องในการจำแนกของตัวอย่าง x ใด ๆ จะขึ้นอยู่กับระยะห่างระหว่างตัวอย่างกับระนาบ ถ้าระนาบอยู่ใกล้ข้อมูลตัวอย่างที่จะทดสอบมาก ตัวอย่าง x จะมีระยะห่างกับประเภทข้อมูลที่อยู่ในอีกด้านหนึ่งของระนาบน้อยมาก แต่เมื่อมีการจำแนก ข้อมูลที่อยู่ในอีกด้านของระนาบกับตัวอย่าง x แม้จะมีระยะห่างกับตัวอย่าง x น้อยกว่าข้อมูลที่ไม่ถูกตัดทิ้งบางตัวเพราะอยู่ในด้านเดียวกัน

5.2 ข้อเสนอแนะ

1. จากปัญหาของข้อจำกัดของการใช้ระนาบของตัวจำแนกแบบหนึ่งต่อหนึ่งกับข้อมูลที่มีการกระจายตัวกันน้อย ทำให้ไม่สามารถหาระนาบที่แบ่งข้อมูลได้สมดุลกันนั้น แนวทางหนึ่งที่อาจเพิ่มประสิทธิภาพของวิธีแตกครึ่งแบบสมดุลได้นั้น คือการให้ระนาบแบบหนึ่งต่อหนึ่งที่เราเลือกมาเป็นระนาบแตกครึ่งดีสุดแล้ว สามารถย้ายแกนเปลี่ยนตำแหน่งของระนาบได้เพื่อหาตำแหน่งที่แบ่งข้อมูลได้สมดุลกว่าตำแหน่งเดิมในกรณีที่ยังมีประเภทข้อมูลเดียวกันตกอยู่ในทั้ง 2 ด้านของระนาบได้

2. ปัญหาต่อมาคือเรื่องของการใช้ระนาบเพียง 1 ระนาบในการแบ่งข้อมูล เราอาจแก้ไขได้ด้วยวิธีการเพิ่มระนาบสมมาตรเข้าไปในมิติ โดยให้ระนาบเสมือนมีตำแหน่งอยู่ในด้านตรงกันข้ามกับระนาบจริงโดยมีตัวอย่างที่จะใช้ทดสอบเป็นแกนกลางแล้วทำการจำแนกข้อมูลด้วยการขยายรัศมีของระนาบ กับตัวอย่างที่จะใช้ทดสอบออกไปจนได้ระยะห่างที่สามารถแบ่งประเภทข้อมูลได้เป็นจำนวนที่เราต้องการแทน แนวทางนี้จะช่วยลดปัญหาการตัดข้อมูลที่มีระยะห่างกับข้อมูลทดสอบน้อยแต่อยู่คนละด้านของระนาบได้ และยังสามารถกำหนดจำนวนประเภทข้อมูล ที่ต้องการให้เหลืออยู่ในการจำแนกแต่ละครั้งได้

3. อีกปัญหาหนึ่งคือเรื่องของเวลาที่ใช้ในการคำนวณหาค่า P และ R เราอาจแก้ปัญหาในเรื่องนี้ได้ด้วยการใช้ฮิวริสติกฟังก์ชันในการคำนวณหาค่าที่ดีที่สุดของพารามิเตอร์ 2 ตัวนี้แทนการนำค่าทุกค่าที่เป็นไปได้มาทดสอบทั้งหมด ก็จะลดเวลาในส่วนนี้ไปได้

รายการอ้างอิง

[1] Bartlett, P. L., and Shawe-Taylor, J., 1999. Generalization performance of support vector machines and other pattern classifiers. Advances in Kernel Methods - Support Vector Learning, 43-54, MIT Press, Cambridge, USA.

[2] Blake, C., Keogh, E., and Merz, C, 1998. UCI Repository of Machine Learning Databases [Online], Irvine: University of California, Department of Information and Computer Science Available from: http://www.ics.uci.edu/~mlearn/MLSummary.html [2000, August 21]

[3] Cook, W. and Rohe, A. 1997. Computing Minimum-Weight Perfect Matchings, Technical Report 97863, Forschungsinstitut für Diskrete Mathematik, Universität Bonn.

[4] Friedman, J. H. 1996, Another Approach to Polychotomous classification. Technical report. Department of Statistics, Stanford University.

[5] Hsu, C. and Lin, C., 2002. A Comparison of Methods for Multiclass Support Vector Machines, IEEE Transactions on Neural Networks 13: 415-425.

[6] Mitchell, T. 1997. Machine Learning. Carnegie Mellon University: McGraw-Hill.

[7] Platt, J., Cristianini, N. and Shawe-Taylor, J. 1999, Large Margin DAGs for Multiclass Classification. Advances in Neural Information Processing Systems 12: 547-553.

[8] Phetkaew, T., Kijsirikul, B. and Rivepiboon, W. 2003. Reordering Adaptive Directed Acyclic Graphs: An Improved Algorithm for Multiclass Support Vector Machines. The 2003 IEEE/INNS International Joint Conference on Neural Networks, Portland, Oregon.

[9] Ussivakul, N. and Kijsirikul, B. 2001. Adaptive DAG: Another Approach for Multiclass Classification. International Conference on Intelligent Technologies.

[10] Vapnik, V. 1998. Statistical Learning Theory. New York: Wiley.

ภาคผนวก

ภาคผนวก

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Glass 1

Polynomial kernel					
D	Fold A	Fold B	Fold C	Fold D	Fold E
2	72.093	67.442	72.093	60.465	72.093
3	69.767	62.791	72.093	60.465	72.093
4	67.442	62.791	69.767	60.465	74.419
5	69.767	60.465	72.093	65.116	74.419
6	69.767	60.465	72.093	67.442	72.093
7	67.442	60.465	72.093	62.791	67.442
8	76.744	60.465	72.093	65.116	72.093
RBF kernel					
C	Fold A	Fold B	Fold C	Fold D	Fold E
0.01	69.767	65.116	76.744	67.442	72.093
0.02	72.093	65.116	74.419	67.442	69.767
0.03	67.442	62.791	72.093	72.093	62.791
0.04	67.442	62.791	72.093	72.093	62.791
0.05	62.791	62.791	74.419	69.767	65.116
0.06	67.442	65.116	76.744	67.442	72.093
0.07	72.093	65.116	76.744	67.442	74.419
0.08	72.093	67.442	76.744	65.116	74.419
0.09	72.093	67.442	76.744	65.116	74.419
0.1	72.093	67.442	72.093	65.116	72.093

ค่า P และ R ในชุดข้อมูล Glass 1

Polynomial kernel										
	Fold A		Fold B		Fold C		Fold D		Fold E	
D	P	R	P	R	P	R	P	R	P	R
2	8	1.5	8	1.8	8	1.5	8	2.0	8	2.0
3	8	1.5	8	1.8	8	1.5	8	1.8	8	1.8
4	8	1.5	8	1.8	8	1.5	8	1.8	8	1.8
5	1	1.5	1	1.8	1	1.5	1	1.8	1	1.8
6	1	1.5	1	1.8	1	1.5	1	1.8	1	1.8
7	1	1.5	1	1.8	1	1.5	0	1.3	1	1.8
8	6	1.3	7	1.3	6	1.3	7	1.5	6	1.3
RBF kernel										
	Fold A		Fold B		Fold C		Fold D		Fold E	
C	P	R	P	R	P	R	P	R	P	R
0.01	6	1.5	7	1.5	8	1.5	7	1.5	7	1.5
0.02	2	1.8	3	1.5	2	1.8	3	1.8	3	1.8
0.03	8	1.5	8	1.8	8	1.5	8	1.8	8	1.8
0.04	8	1.5	8	1.8	8	1.5	8	1.8	8	1.8
0.05	7	1.3	8	1.0	7	1.3	8	1.0	8	1.0
0.06	3	1.5	4	1.5	3	1.5	4	1.5	4	1.5
0.07	3	1.5	3	1.5	3	1.5	4	1.5	3	1.5
0.08	3	1.5	4	1.5	4	1.5	5	1.5	6	1.5
0.09	3	1.5	2	1.5	3	1.5	4	1.5	4	1.5
0.1	3	1.5	4	1.5	2	1.5	4	1.5	2	1.5

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Glass 2

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	74.419	65.116	72.093	69.767	69.767
3	72.093	60.465	72.093	69.767	69.767
4	72.093	60.465	69.767	72.093	69.767
5	72.093	62.791	69.767	72.093	72.093
6	69.767	65.116	69.767	72.093	72.093
7	69.767	60.465	72.093	67.442	67.442
8	72.093	62.791	67.442	69.767	72.093
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.01	76.744	62.791	74.419	72.093	79.070
0.02	76.744	62.791	74.419	72.093	76.744
0.03	72.093	65.116	72.093	74.419	76.744
0.04	72.093	65.116	76.744	74.419	76.744
0.05	74.419	60.465	72.093	69.767	72.093
0.06	76.744	72.093	79.070	74.419	79.070
0.07	76.744	60.465	79.070	79.070	79.070
0.08	74.419	60.465	81.395	79.070	76.744
0.09	74.419	60.465	81.395	79.070	79.070
0.1	74.419	60.465	76.744	74.419	74.419

ค่า P และ R ในชุดข้อมูล Glass 2

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	8	1.5	8	2.0	8	2.0	8	1.5	8	1.8
3	8	1.5	8	1.8	8	1.8	8	1.5	7	1.8
4	8	1.5	8	1.8	7	2.0	8	1.5	7	1.8
5	1	1.3	1	1.8	1	1.8	1	1.5	1	1.8
6	1	1.5	2	1.8	1	1.8	1	1.5	1	1.8
7	2	1.5	0	1.3	1	1.8	1	1.5	0	1.3
8	6	1.3	7	1.5	6	1.0	7	1.3	7	1.5
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.01	7	1.5	7	1.5	8	1.5	8	1.5	7	1.5
0.02	3	1.8	3	1.5	2	1.8	2	1.8	2	1.5
0.03	8	1.8	8	1.8	8	1.5	8	1.5	8	1.3
0.04	7	1.8	8	1.8	8	1.5	8	1.5	8	1.3
0.05	8	1.0	8	1.0	7	1.3	7	1.3	8	1.0
0.06	2	1.5	4	1.5	3	1.5	3	1.5	4	1.5
0.07	4	1.5	3	1.5	3	1.5	3	1.5	3	1.5
0.08	7	1.5	8	1.5	3	1.5	7	1.5	9	1.5
0.09	3	1.5	2	1.5	3	1.5	3	1.5	4	1.5
0.1	4	1.5	4	1.5	3	1.5	3	1.5	2	1.5

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Glass 3

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	72.093	69.767	60.465	74.419	69.767
3	69.767	67.442	60.465	74.419	69.767
4	69.767	67.442	60.465	72.093	74.419
5	69.767	74.419	62.791	69.767	74.419
6	67.442	74.419	62.791	72.093	74.419
7	67.442	67.442	60.465	69.767	69.767
8	74.419	72.093	62.791	69.767	76.744
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.01	74.419	72.093	65.116	74.419	76.744
0.02	74.419	72.093	65.116	74.419	76.744
0.03	72.093	74.419	69.767	76.744	76.744
0.04	72.093	74.419	69.767	76.744	72.093
0.05	69.767	72.093	72.093	74.419	72.093
0.06	74.419	72.093	65.116	76.744	74.419
0.07	76.744	74.419	65.116	76.744	74.419
0.08	76.744	74.419	65.116	79.070	76.744
0.09	76.744	76.744	67.442	79.070	74.419
0.1	76.744	76.744	67.442	72.093	74.419

ค่า P และ R ในชุดข้อมูล Glass 3

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	8	1.5	8	1.8	8	1.5	8	2.0	8	2.0
3	8	1.5	8	1.8	8	1.5	8	1.8	8	1.8
4	8	1.5	8	1.8	8	1.5	8	1.8	8	1.8
5	1	1.5	1	1.8	1	1.5	1	1.8	1	1.8
6	1	1.5	1	1.8	1	1.5	1	1.8	1	1.8
7	1	1.5	1	1.8	1	1.5	0	1.3	1	1.8
8	6	1.3	7	1.3	6	1.3	7	1.5	6	1.3
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.01	8	1.5	7	1.5	8	1.5	7	1.5	7	1.5
0.02	2	1.8	3	1.5	2	1.8	3	1.8	3	1.8
0.03	8	1.5	8	1.8	8	1.5	8	1.8	8	1.8
0.04	8	1.5	8	1.8	8	1.5	8	1.8	8	1.8
0.05	7	1.3	8	1.0	7	1.3	8	1.0	8	1.0
0.06	3	1.5	4	1.5	3	1.5	4	1.5	4	1.5
0.07	3	1.5	3	1.5	3	1.5	4	1.5	3	1.5
0.08	3	1.5	3	1.5	4	1.5	4	1.5	4	1.5
0.09	3	1.5	2	1.5	3	1.5	4	1.5	4	1.5
0.1	3	1.5	4	1.5	2	1.5	4	1.5	2	1.5

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Glass 4

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	72.093	74.419	69.767	74.419	62.791
3	69.767	74.419	67.442	74.419	62.791
4	69.767	72.093	69.767	74.419	62.791
5	67.442	69.767	72.093	72.093	60.465
6	67.442	69.767	72.093	72.093	60.465
7	67.442	69.767	69.767	69.767	58.140
8	67.442	74.419	69.767	74.419	62.791
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.01	76.744	72.093	76.744	74.419	62.791
0.02	72.093	72.093	72.093	76.744	62.791
0.03	74.419	74.419	72.093	76.744	65.116
0.04	76.744	74.419	72.093	74.419	65.116
0.05	72.093	72.093	74.419	72.093	60.465
0.06	76.744	72.093	74.419	76.744	60.465
0.07	76.744	76.744	74.419	72.093	67.442
0.08	79.070	76.744	76.744	76.744	67.442
0.09	76.744	76.744	76.744	76.744	65.116
0.1	76.744	76.744	74.419	72.093	65.116

ค่า P และ R ในชุดข้อมูล Glass 4

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	8	1.8	8	1.8	8	1.5	8	1.5	8	2.0
3	8	1.8	8	1.8	7	1.8	7	1.8	8	1.8
4	8	1.8	7	1.8	8	1.5	8	1.5	8	1.8
5	1	1.5	1	1.8	0	1.8	1	1.8	1	1.5
6	0	1.3	0	1.3	1	1.8	0	1.3	1	1.5
7	0	1.5	1	1.8	1	1.3	0	1.3	1	1.8
8	7	1.3	7	1.5	6	1.5	7	1.5	6	1.3
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.01	7	1.5	7	1.5	8	1.5	8	1.5	7	1.5
0.02	3	1.5	3	1.8	2	1.8	2	1.8	3	1.8
0.03	8	1.8	8	1.8	8	1.5	8	1.5	8	1.8
0.04	8	1.8	8	1.8	8	1.5	8	1.5	8	1.8
0.05	8	1.0	8	1.0	7	1.3	7	1.3	8	1.0
0.06	4	1.5	4	1.5	3	1.5	4	1.5	4	1.5
0.07	3	1.5	4	1.5	3	1.5	3	1.5	3	1.5
0.08	3	1.5	3	1.5	4	1.5	3	1.5	3	1.5
0.09	3	1.5	3	1.5	3	1.5	4	1.5	4	1.5
0.1	3	1.5	2	1.5	3	1.5	4	1.5	4	1.5

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Glass 5

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	78.571	76.190	78.571	80.952	78.571
3	73.810	71.429	73.810	76.190	73.810
4	73.810	71.429	73.810	76.190	73.810
5	71.429	71.429	69.048	71.429	73.810
6	69.048	69.048	66.667	69.048	71.429
7	69.048	71.429	66.667	69.048	69.048
8	66.667	69.048	64.286	59.524	61.905
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.01	76.190	73.810	71.429	71.429	76.190
0.02	76.190	71.429	73.810	71.429	76.190
0.03	76.190	71.429	73.810	76.190	71.429
0.04	71.429	76.190	73.810	76.190	71.429
0.05	73.810	78.571	76.190	78.571	73.810
0.06	73.810	78.571	76.190	78.571	73.810
0.07	76.190	78.571	73.810	78.571	73.810
0.08	76.190	78.571	73.810	78.571	73.810
0.09	78.571	78.571	78.571	80.952	76.190
0.1	78.571	73.810	73.810	78.571	76.190

ค่า P และ R ในชุดข้อมูล Glass 5

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	8	1.5	8	1.5	8	1.5	7	1.8	7	1.8
3	8	1.5	8	1.5	8	1.5	8	1.8	8	1.8
4	8	1.5	8	1.8	8	1.5	8	1.8	8	1.8
5	1	1.5	1	1.8	1	1.5	1	1.8	1	1.8
6	1	1.5	1	1.8	1	1.5	1	1.8	1	1.5
7	1	1.5	1	1.8	1	1.5	1	1.8	1	1.5
8	6	1.3	7	1.3	6	1.3	6	1.3	6	1.3
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.01	8	1.5	7	1.5	7	1.5	8	1.5	8	1.5
0.02	2	1.8	3	1.5	3	1.8	2	1.8	2	1.8
0.03	8	1.5	8	1.8	8	1.8	8	1.5	8	1.5
0.04	8	1.5	8	1.8	8	1.8	8	1.5	8	1.5
0.05	7	1.3	8	1.0	8	1.0	7	1.3	7	1.3
0.06	3	1.5	3	1.5	4	1.5	3	1.5	3	1.5
0.07	3	1.5	3	1.5	3	1.5	3	1.5	3	1.5
0.08	3	1.5	3	1.5	3	1.5	4	1.5	4	1.5
0.09	3	1.8	4	1.5	3	1.5	3	1.5	3	1.5
0.1	3	1.8	4	1.5	3	1.5	4	1.3	4	1.3

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Satimage

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	87.450	87.600	87.500	87.400	87.300
3	87.600	87.750	87.700	87.500	87.450
4	87.950	88.000	88.100	87.700	87.750
5	88.650	88.800	88.750	88.700	88.850
6	88.800	89.050	88.900	89.000	89.000
7	88.500	88.700	88.600	88.400	88.300
8	88.200	88.250	88.300	88.150	88.100
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.5	89.500	89.400	89.300	89.450	89.350
1.0	90.600	90.500	90.350	90.400	90.400
1.5	91.000	91.050	90.850	90.900	90.950
2.0	91.700	91.650	91.500	91.400	91.500
3.0	91.900	92.050	91.850	91.950	92.000
4.0	92.000	92.000	92.100	92.150	92.250
5.0	91.700	91.900	91.650	91.850	91.900

ค่า P และ R ในชุดข้อมูล Satimage

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	5	1.5	4	1.5	5	1.5	4	1.8	4	1.5
3	5	1.5	5	1.5	5	1.5	5	1.5	5	1.5
4	4	1.3	4	1.5	5	1.3	5	1.3	4	1.5
5	4	1.3	4	1.5	5	1.5	4	1.5	4	1.3
6	1	1.5	1	1.3	1	1.3	1	1.3	1	1.5
7	1	1.0	1	1.3	0	1.0	1	1.3	1	1.3
8	5	1.3	5	1.5	4	1.3	5	1.3	4	1.3
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.5	7	1.3	7	1.3	7	1.3	8	1.3	8	1.3
1.0	8	1.5	8	1.5	8	1.3	8	1.5	8	1.5
1.5	8	1.3	7	1.5	8	1.3	8	1.5	7	1.5
2.0	1	1.5	1	1.3	1	1.5	1	1.5	1	1.5
3.0	1	1.3	1	1.3	1	1.3	1	1.5	1	1.5
4.0	1	1.3	1	1.5	1	1.5	1	1.3	1	1.3
5.0	3	1.5	3	1.5	2	1.8	3	1.5	2	1.8

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Segment 1

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	96.104	95.671	97.403	96.970	97.619
3	96.320	96.104	97.403	96.753	97.186
4	96.970	96.537	97.619	97.403	97.403
5	97.186	96.104	97.403	97.186	96.970
6	97.403	96.753	97.403	97.186	97.186
7	97.619	96.970	97.835	97.186	97.403
8	97.835	96.970	98.052	97.619	97.619
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.5	96.104	97.186	97.403	96.753	98.485
0.6	96.104	97.186	97.186	97.186	98.268
0.7	96.753	97.619	97.835	97.403	98.485
0.8	96.753	97.619	97.835	97.619	98.268
0.9	96.970	97.403	97.403	97.403	97.835
1.0	97.186	96.970	97.186	97.186	97.403
1.5	96.970	96.970	97.186	97.403	97.403

ค่า P และ R ในชุดข้อมูล Segment 1

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	0	1.3	0	1.5	1	1.3	0	1.5	0	1.3
3	0	1.3	0	1.3	0	1.5	0	1.5	0	1.3
4	1	1.3	1	1.5	1	1.3	1	1.3	1	1.5
5	2	1.5	2	1.3	3	1.0	2	1.5	2	1.3
6	0	1.8	1	1.8	1	1.8	2	1.3	0	1.8
7	0	1.8	0	1.8	0	1.5	1	1.3	0	1.8
8	0	1.3	1	1.3	0	1.3	0	1.8	0	1.3
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.5	0	1.8	1	1.3	1	1.3	0	1.8	0	1.8
0.6	1	1.3	2	1.3	2	1.3	1	1.3	1	1.8
0.7	1	1.3	0	1.8	1	1.3	0	1.5	0	1.5
0.8	1	1.3	0	1.8	1	1.5	0	1.5	0	1.8
0.9	2	1.3	1	1.5	1	1.5	1	1.3	1	1.3
1.0	0	1.5	1	1.3	1	1.0	0	1.5	1	1.3
1.5	0	1.5	0	1.8	0	1.5	0	1.8	0	1.8

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Segment 2

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	96.537	96.970	97.403	95.887	98.052
3	96.320	96.970	96.970	95.671	97.835
4	96.320	97.403	97.186	95.671	98.268
5	95.887	97.186	97.403	96.104	98.268
6	96.320	97.403	97.619	96.753	97.835
7	96.320	97.186	96.753	96.104	97.403
8	96.537	97.619	97.186	97.186	97.403
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.5	95.887	97.835	97.835	96.537	98.918
0.6	96.320	97.619	97.835	96.537	98.701
0.7	96.320	97.619	97.186	96.320	98.485
0.8	96.104	97.403	97.186	96.753	98.485
0.9	95.887	97.186	97.186	96.537	98.052
1.0	95.887	96.970	96.753	96.320	97.835
1.5	95.671	96.753	96.970	96.753	97.619

ค่า P และ R ในชุดข้อมูล Segment 2

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	0	1.5	0	1.5	0	1.3	1	1.3	0	1.5
3	0	1.3	0	1.3	1	1.3	0	1.3	1	1.5
4	1	1.3	1	1.5	1	1.3	1	1.5	1	1.5
5	2	1.5	2	1.3	2	1.5	2	1.3	2	1.3
6	1	1.5	1	1.8	0	1.8	1	1.8	0	1.8
7	1	1.0	0	1.8	0	1.8	0	1.8	0	1.8
8	0	1.5	1	1.3	0	1.3	1	1.3	0	1.3
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.5	0	1.5	1	1.3	1	1.0	0	1.0	0	1.8
0.6	1	1.0	2	1.3	2	1.3	2	1.3	1	1.3
0.7	0	1.3	0	1.8	1	1.3	1	1.3	0	1.3
0.8	2	1.5	0	1.5	0	1.5	0	1.5	0	1.5
0.9	2	1.3	2	1.3	1	1.5	1	1.5	1	1.5
1.0	0	1.8	1	1.0	1	1.3	1	1.3	1	1.3
1.5	0	1.8	0	1.5	0	1.5	0	1.5	0	1.5

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Segment 3

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	97.619	96.537	98.268	96.320	97.186
3	97.403	96.320	98.052	96.104	96.970
4	97.186	96.320	97.403	96.320	96.537
5	96.537	95.671	97.186	96.537	96.753
6	96.537	96.320	97.403	96.320	97.186
7	96.320	96.970	98.052	96.104	97.403
8	96.753	97.186	98.052	96.320	97.619
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.5	96.970	95.455	97.403	98.485	96.537
0.6	96.753	95.887	97.403	98.268	96.537
0.7	96.970	96.320	97.619	98.268	96.753
0.8	97.619	96.320	97.186	98.052	96.753
0.9	97.186	96.970	96.970	98.052	96.753
1.0	96.970	96.537	97.186	97.619	96.537
1.5	97.186	96.537	96.970	97.403	96.753

ค่า P และ R ในชุดข้อมูล Segment 3

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	0	1.5	1	1.3	0	1.5	0	1.5	0	1.5
3	0	1.3	0	1.3	0	1.3	0	1.3	0	1.5
4	1	1.5	1	1.5	1	1.5	1	1.3	1	1.3
5	2	1.3	2	1.3	2	1.3	2	1.5	2	1.5
6	1	1.8	1	1.8	1	1.8	1	1.5	2	1.3
7	0	1.8	0	1.8	0	1.8	1	1.0	1	1.3
8	1	1.3	1	1.3	1	1.3	0	1.5	0	1.8
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.5	1	1.5	1	1.5	1	1.0	0	1.3	0	1.5
0.6	1	1.0	2	1.3	2	1.3	2	1.3	1	1.3
0.7	1	1.3	1	1.5	1	1.3	1	1.3	0	1.3
0.8	2	1.5	0	1.5	0	1.5	0	1.5	0	1.5
0.9	2	1.3	2	1.3	1	1.5	1	1.5	1	1.5
1.0	0	1.5	1	1.0	1	1.3	1	1.3	1	1.3
1.5	0	1.5	0	1.5	0	1.5	0	1.5	1	1.3

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Segment 4

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	97.403	96.753	96.104	97.186	96.320
3	97.186	96.104	95.671	96.753	95.887
4	97.619	96.970	96.320	97.403	96.537
5	98.052	97.186	96.753	97.186	96.753
6	98.052	97.186	96.753	97.186	96.753
7	98.268	97.403	96.970	97.403	96.970
8	98.485	97.619	97.186	97.619	97.186
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.5	97.403	96.537	96.970	98.485	95.455
0.6	97.403	96.537	96.753	98.268	95.887
0.7	97.619	96.753	96.970	98.268	96.320
0.8	97.619	96.753	97.186	98.052	96.320
0.9	97.186	96.970	96.970	98.052	96.753
1.0	96.970	96.537	97.186	97.619	96.537
1.5	97.186	96.537	96.970	97.403	96.753

ค่า P และ R ในชุดข้อมูล Segment 4

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	0	1.5	0	1.5	0	1.5	1	1.3	0	1.5
3	0	1.3	0	1.3	0	1.5	0	1.3	0	1.3
4	1	1.5	1	1.5	1	1.3	1	1.5	1	1.3
5	2	1.3	2	1.3	2	1.5	2	1.3	2	1.5
6	1	1.8	1	1.8	2	1.3	1	1.8	1	1.5
7	0	1.8	0	1.8	1	1.3	0	1.8	1	1.0
8	1	1.3	1	1.3	0	1.8	1	1.3	0	1.5
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.5	0	1.3	0	1.5	0	1.5	1	1.0	1	1.5
0.6	2	1.3	1	1.0	2	1.3	2	1.3	1	1.0
0.7	1	1.3	1	1.3	1	1.5	1	1.3	1	1.3
0.8	0	1.5	2	1.5	0	1.5	0	1.5	2	1.5
0.9	1	1.5	2	1.3	2	1.3	1	1.5	1	1.5
1.0	1	1.3	0	1.5	1	1.0	1	1.3	1	1.3
1.5	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Segment 5

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	97.403	95.887	98.485	96.753	96.320
3	97.186	96.104	98.052	96.537	95.887
4	97.186	96.104	98.052	96.537	95.887
5	97.619	96.970	98.268	97.186	96.970
6	98.052	97.619	98.485	97.835	97.186
7	98.052	97.619	98.485	97.835	97.186
8	98.268	97.835	98.701	98.052	97.403
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.5	98.268	96.537	95.671	97.186	96.104
0.6	98.052	96.753	95.671	96.970	96.320
0.7	98.052	96.753	96.104	97.403	96.537
0.8	97.835	96.970	96.104	97.186	96.753
0.9	97.835	96.970	97.619	96.537	96.970
1.0	98.268	97.403	97.403	96.753	97.186
1.5	97.835	97.403	97.403	96.970	97.403

ค่า P และ R ในชุดข้อมูล Segment 5

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	1	1.5	0	1.5	0	1.5	1	1.3	0	1.5
3	0	1.3	0	1.3	0	1.5	0	1.3	0	1.3
4	1	1.5	1	1.5	1	1.3	1	1.5	1	1.3
5	2	1.3	2	1.3	2	1.5	2	1.3	2	1.5
6	1	1.8	1	1.8	2	1.3	1	1.8	1	1.5
7	0	1.8	0	1.8	1	1.3	0	1.8	1	1.0
8	1	1.3	1	1.3	0	1.8	1	1.3	0	1.5
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.5	0	1.3	0	1.5	1	1.5	1	1.0	1	1.5
0.6	2	1.3	2	1.0	1	1.0	2	1.3	1	1.0
0.7	1	1.3	1	1.3	1	1.5	1	1.3	1	1.3
0.8	0	1.5	2	1.5	0	1.5	0	1.5	2	1.5
0.9	1	1.5	1	1.3	2	1.3	1	1.5	2	1.5
1.0	1	1.3	0	1.5	1	1.0	1	1.3	1	1.3
1.5	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Shuttle

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	99.834	99.841	99.841	99.841	99.848
3	99.869	99.869	99.869	99.869	99.869
4	99.869	99.869	99.869	99.869	99.869
5	99.890	99.890	99.890	99.890	99.890
6	99.903	99.903	99.903	99.903	99.903
7	99.917	99.917	99.917	99.917	99.917
8	99.924	99.924	99.924	99.924	99.924
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.5	99.834	99.834	99.834	99.834	99.834
1.0	99.862	99.862	99.862	99.862	99.862
1.5	99.876	99.876	99.876	99.876	99.876
2.0	99.883	99.883	99.883	99.883	99.883
3.0	99.897	99.897	99.897	99.897	99.897
4.0	99.897	99.897	99.897	99.897	99.897
5.0	99.897	99.897	99.897	99.897	99.897

ค่า P และ R ในชุดข้อมูล Shuttle

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	1	1.0	1	1.0	1	1.0	0	1.3	0	1.5
3	1	1.0	0	1.5	1	1.0	0	1.5	0	1.5
4	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5
5	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
6	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5
7	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5
8	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.5	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
1.0	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
1.5	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
2.0	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5
3.0	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5
4.0	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5
5.0	0	1.5	0	1.5	0	1.5	0	1.5	0	1.5

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Vowel

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	69.264	66.667	66.234	68.182	67.314
3	67.749	66.234	65.801	67.100	66.450
4	60.173	61.472	62.338	62.771	60.606
5	61.255	62.554	62.771	61.039	61.905
6	56.926	56.494	57.576	58.442	56.277
7	57.359	56.710	56.494	58.658	57.576
8	55.844	57.792	56.277	57.359	58.442
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.1	66.450	62.771	64.719	63.853	63.632
0.2	68.831	70.130	67.749	68.615	68.831
0.3	69.264	68.398	66.234	67.100	70.996
0.4	67.532	65.584	64.069	66.017	64.719
0.5	65.584	63.636	64.502	66.439	62.771
1.0	61.472	61.039	60.390	61.905	64.719
1.5	59.957	60.606	58.442	59.091	61.688
2.0	59.091	60.173	58.442	58.225	59.524

ค่า P และ R ในชุดข้อมูล Vowel

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	3	1.5	3	1.5	3	1.5	3	1.5	3	1.5
3	3	1.5	3	1.5	3	1.5	3	1.5	3	1.5
4	1	1.8	1	1.8	1	1.8	1	1.8	1	1.8
5	10	1.3	10	1.3	10	1.3	8	1.3	10	1.3
6	1	1.8	1	1.8	1	1.8	1	1.8	1	1.8
7	1	1.5	1	1.5	1	1.5	1	1.5	1	1.5
8	8	1.3	8	1.3	8	1.3	8	1.3	8	1.3
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.1	8	1.5	8	1.5	8	1.5	8	1.5	8	1.5
0.2	10	1.5	10	1.5	8	1.5	10	1.5	10	1.5
0.3	10	1.8	10	1.8	10	1.8	10	1.8	10	1.8
0.4	10	1.5	10	1.8	10	1.5	10	1.3	10	1.8
0.5	3	1.8	3	1.8	3	1.8	3	1.8	3	1.8
1.0	6	1.0	5	1.5	5	1.5	5	1.5	6	1.0
1.5	1	1.3	1	1.3	1	1.5	1	1.3	1	1.3
2.0	1	1.3	1	1.8	1	1.3	1	1.8	1	1.3

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Soybean

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	90.000	89.706	89.118	90.882	91.765
3	90.588	90.294	89.706	91.471	92.353
4	90.294	90.000	89.412	91.176	92.059
5	90.294	90.000	89.412	91.176	92.059
6	89.412	89.118	88.824	90.000	91.176
7	90.000	88.824	89.118	90.588	91.471
8	86.471	85.882	87.059	88.965	90.147
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.04	90.294	89.412	92.647	90.882	88.235
0.05	91.176	90.000	92.353	90.588	88.824
0.06	91.471	88.824	91.765	90.294	89.118
0.07	91.176	89.412	92.059	90.588	89.706
0.08	91.176	89.412	92.059	90.588	89.706
0.09	90.882	88.824	91.765	90.882	89.118
0.1	90.000	88.235	91.176	91.176	89.412
0.2	85.882	86.471	86.471	88.235	85.294

ค่า P และ R ในชุดข้อมูล Soybean

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	3	1.5	4	1.5	3	1.3	3	1.5	4	1.5
3	3	1.5	3	1.5	3	1.5	3	1.3	3	1.3
4	1	1.5	1	1.5	1	1.8	1	1.8	1	1.8
5	5	1.3	5	1.3	5	1.3	5	1.3	5	1.3
6	1	1.8	0	1.8	1	1.0	0	1.8	1	1.0
7	1	1.5	0	1.5	0	1.5	1	1.5	1	1.5
8	1	1.3	1	1.3	1	1.3	1	1.3	1	1.3
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.04	3	1.5	3	1.5	3	1.5	3	1.5	3	1.5
0.05	3	1.5	3	1.5	3	1.5	3	1.5	3	1.5
0.06	1	1.8	1	1.8	1	1.8	1	1.8	1	1.8
0.07	1	1.3	1	1.3	1	1.3	1	1.3	1	1.3
0.08	1	1.3	1	1.3	1	1.3	1	1.3	1	1.3
0.09	3	1.8	4	1.5	3	1.8	3	1.8	3	1.8
0.1	3	1.5	4	1.5	3	1.5	3	1.5	3	1.5
0.2	1	1.3	1	1.3	1	1.3	1	1.3	1	1.3

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Letter

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	95.120	95.021	95.170	95.071	95.219
3	96.235	96.086	96.111	96.210	96.161
4	95.987	95.863	95.616	95.863	95.863
5	95.739	95.987	95.789	95.690	95.616
6	95.071	95.194	94.996	95.318	95.145
7	94.649	94.550	94.253	94.699	94.848
8	93.683	93.832	93.535	93.683	93.931
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.5	96.743	96.805	96.705	96.854	96.606
1.0	97.275	97.350	97.473	97.572	97.201
1.5	97.473	97.597	97.696	97.399	97.325
2.0	97.424	97.721	97.771	97.597	97.473
3.0	98.043	98.093	97.842	97.721	97.771
4.0	97.350	97.597	97.350	97.102	96.961
5.0	97.572	97.894	97.771	97.696	97.362

ค่า P และ R ในชุดข้อมูล Letter

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	0	1.8	0	1.8	0	1.3	0	1.8	0	1.8
3	0	1.3	0	1.3	0	1.3	1	1.3	0	1.3
4	0	1.8	0	1.8	0	1.3	0	1.8	0	1.8
5	0	1.3	0	1.8	0	1.3	0	1.3	0	1.3
6	0	1.8	0	1.8	0	1.8	0	1.8	0	1.8
7	1	1.3	1	1.8	1	1.3	1	1.8	1	1.8
8	1	1.8	1	1.8	1	1.8	1	1.8	1	1.8
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.5	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
1.0	0	1.8	0	1.8	0	1.8	0	1.8	0	1.8
1.5	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
2.0	0	1.8	0	1.8	0	1.8	0	1.8	0	1.8
3.0	1	1.3	0	1.3	1	1.3	0	1.3	1	1.3
4.0	1	1.3	1	1.3	0	1.3	1	1.3	1	1.3
5.0	0	1.8	1	1.8	0	1.8	1	1.8	1	1.8

ผลการทดลองของวิธีแตกครึ่งแบบสมดุลในชุดข้อมูล Isolet

Polynomial kernel					
D	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
2	96.216	95.831	96.376	96.023	96.472
3	96.472	96.087	96.536	96.280	96.665
4	96.376	96.151	96.472	96.344	96.536
5	96.376	96.151	96.472	96.344	96.536
6	96.536	96.280	96.665	96.472	96.729
7	96.216	96.087	96.344	96.280	96.472
8	95.510	95.253	95.574	95.638	96.055
RBF kernel					
C	**Fold A**	**Fold B**	**Fold C**	**Fold D**	**Fold E**
0.001	96.344	95.574	96.087	95.189	95.799
0.002	96.536	95.831	96.216	95.318	95.895
0.003	96.793	96.087	96.536	95.574	96.090
0.004	96.921	96.216	96.729	95.895	96.440
0.005	96.536	96.023	96.408	96.087	96.186
0.01	96.408	96.408	96.536	96.408	96.280
0.02	96.921	96.793	96.985	96.665	96.600
0.03	96.216	96.151	96.761	96.216	95.895

ค่า P และ R ในชุดข้อมูล Isolet

Polynomial kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
D	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
2	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
3	0	1.8	0	1.8	0	1.8	0	1.8	0	1.8
4	0	1.8	0	1.8	0	1.8	0	1.5	0	1.5
5	0	1.3	0	1.3	0	1.3	0	1.5	0	1.5
6	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
7	0	1.3	0	1.8	0	1.3	0	1.3	0	1.8
8	0	1.8	0	1.8	0	1.8	0	1.8	0	1.8
RBF kernel										
	Fold A		**Fold B**		**Fold C**		**Fold D**		**Fold E**	
C	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
0.001	0	1.3	1	1.3	0	1.3	1	1.3	0	1.3
0.002	0	1.8	0	1.3	0	1.8	0	1.8	0	1.8
0.003	0	1.8	1	1.8	1	1.8	0	1.8	0	1.8
0.004	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
0.005	0	1.3	0	1.3	0	1.3	0	1.3	0	1.3
0.01	1	1.8	1	1.8	1	1.8	1	1.8	1	1.8
0.02	0	1.8	0	1.8	0	1.8	0	1.8	0	1.8
0.03	1	1.3	1	1.3	1	1.3	1	1.3	1	1.3

ประวัติผู้เขียนวิทยานิพนธ์

นาย ณรงค์ บุญสิริสัมพันธ์ เกิดเมื่อวันที่ 9 กุมภาพันธ์ พ.ศ.2524 ที่กรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ จากภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ เมื่อปีการศึกษา 2543 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2545 ปัจจุบันเป็นนักวิจัยอิสระ