บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้จะแบ่งเนื้อหาออกเป็น 2 ส่วนด้วยกัน โดยเนื้อหาในส่วนแรกจะกล่าวถึงทฤษฎีพื้นฐานของซัพพอร์ตเวกเตอร์แมชชีนทั้งกรณีปัญหาที่เป็นแบบเชิงเส้นและไม่เป็นเชิงเส้น และทฤษฎีสารสนเทศในส่วนที่นำมาใช้ช่วยแก้ปัญหาการจำแนกแบบหลายประเภท สำหรับเนื้อหาในส่วนที่สองจะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับปัญหาการจำแนกแบบหลายประเภทด้วยซัพพอร์ตเวกเตอร์แมชชีน

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ซัพพอร์ตเวกเตอร์แมชชีน

แนวคิดหลักในการจำแนกข้อมูล ของซัพพอร์ตเวกเตอร์แมชชีนก็คือ การสร้างระนาบหลายมิติที่ใช้ในการแบ่งข้อมูลออกเป็นสองประเภท

2.1.1.1 ซัพพอร์ตเวกเตอร์แมชชีนแบบเชิงเส้น (Linear support vector machine)

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

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

ระนาบหลายมิติในปริภูมิอันดับ n ถูกกำหนดโดย $(\mathbf{w},b)$ เมื่อ $\mathbf{w}$ คือเวกเตอร์ในปริภูมิอันดับ n ที่ตั้งฉากกับระนาบหลายมิติและ b คือค่าคงที่ระนาบหลายมิติ $(\mathbf{w}.\mathbf{x}) + b$ จะแบ่งข้อมูลได้ก็ต่อเมื่อ

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

ถ้าเราต้องการค่า $\mathbf{w}$ และ b ที่ทำให้จุดที่อยู่ใกล้ระนาบหลายมิติมากที่สุดมีระยะห่าง $1/|\mathbf{w}|$ แล้วจะได้

$$\begin{aligned} (\mathbf{w}\cdot\mathbf{x}_i)+b\ge 1 \quad &\text{if} \quad y_i=+1 \\ (\mathbf{w}\cdot\mathbf{x}_i)+b\le -1 \quad &\text{if} \quad y_i=-1 \end{aligned} \tag{3}$$

ซึ่งเท่ากับ

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

ในการค้นหาระนาบหลายมิติที่ใช้แบ่งข้อมูลดีที่สุดจะต้องค้นหาระนาบหลายมิติที่มีระยะห่างระหว่างข้อมูลที่ใช้สอนกับระนาบหลายมิติที่น้อยที่สุดมีค่ามากที่สุด (ดูรูปที่ 1) ระยะห่างระหว่างข้อมูลตัวอย่างสองตัวจากประเภทที่แตกต่างกันมีค่าเท่ากับ

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

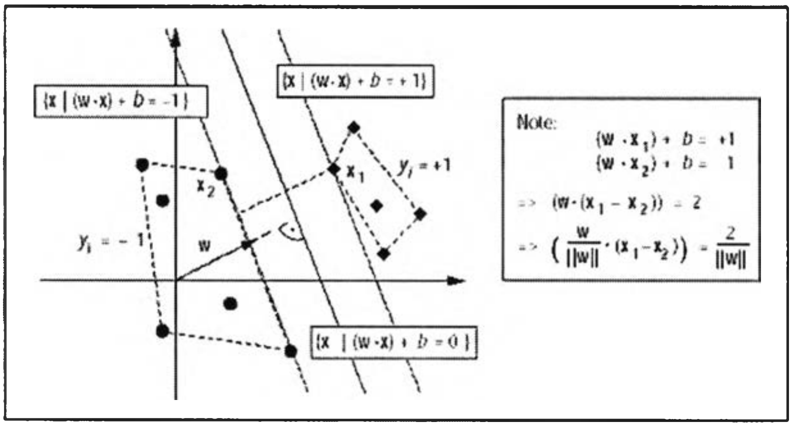
จากสมการที่ (3) ค่าที่น้อยที่สุดและมากที่สุดที่เหมาะสมคือ ±1 ดังนั้นจำเป็นต้องเพิ่มค่าของฟังก์ชัน

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

ให้สูงที่สุด ดังนั้นปัญหานี้จึงเท่ากับ

- ลดค่าของ $|w|^2/2$ ให้ต่ำที่สุดโดยขึ้นกับเงื่อนไขต่อไปนี้

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



รูปที่ 1 ตัวอย่างของการจำแนกข้อมูลสองประเภทด้วยซัพพอร์ตเวกเตอร์แมชชีน

สำหรับกรณีที่ไม่สามารถแบ่งข้อมูลได้โดยเงื่อนไขข้างต้นจะต้องปรับเงื่อนไขใหม่โดยเพิ่มพจน์ค่าปรับซึ่งประกอบด้วยผลรวมของความคลาดเคลื่อน ξ_i จากขอบเขตเข้าไปในเงื่อนไข ดังนั้นปัญหาตอนนี้คือ

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

  (1) $y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b] \ge 1-\xi$

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

พจน์ค่าปรับสำหรับตัวอย่างข้อมูลสอนที่ทำนายผิดพลาดถูกเพิ่มน้ำหนักโดยค่าคงที่ 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 ตัวอย่างข้อมูลสอนแต่ละตัวจะมีตัวคูณลากรองหนึ่งตัวอย่างที่มีค่า $\alpha_i > 0$ เรียกว่า ซัพพอร์ตเวกเตอร์ 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 และข้อมูลสอนโดยเฉพาะอย่างยิ่งในเทอมของซัพพอร์ตเวกเตอร์ได้ดังนี้

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

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

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

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

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

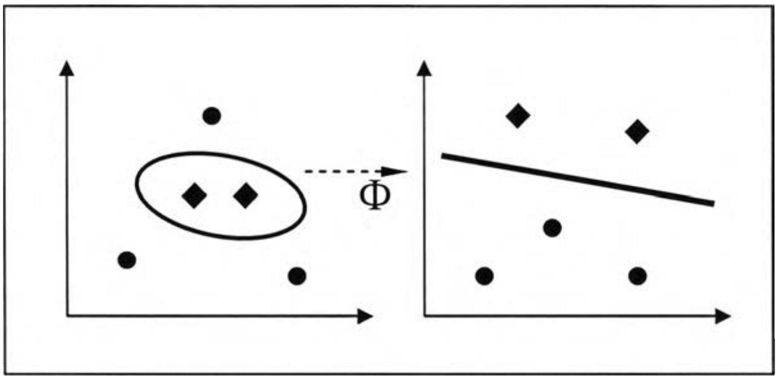
2.1.1.2 **ซัพพอร์ตเวกเตอร์แมชชีนแบบไม่เชิงเส้น** (Non-linear Support vector machine)

อัลกอริทึมที่ได้กล่าวมาแล้วใช้ได้กับข้อมูลที่สามารถแบ่งด้วยระนาบหลายมิติแบบเชิงเส้นเท่านั้น แต่กับข้อมูลที่ไม่อาจแบ่งแบบเชิงเส้นได้ ซัพพอร์ตเวกเตอร์แมชชีนแก้ปัญหานี้โดยแมปข้อมูลตัวอย่าง ไปยังปริภูมิอันดับสูง โดยเลือกใช้ฟังก์ชันการแมปที่ไม่เป็นเชิงเส้น นั่นคือเราเลือกฟังก์ชันในการแมป $\Phi : \Re^n \mapsto \mathrm{H}$ เมื่อปริภูมิของ H มีอันดับสูงกว่าปริภูมิอันดับ n เราสามารถค้นหาระนาบที่แบ่งแยกดีสุดในปริภูมิอันดับสูงที่เทียบเท่ากับการแยกที่ไม่เป็นเชิงเส้นใน $\Re^n$ (ดูรูปที่ 2 ประกอบ)

ตัวอย่างข้อมูลสอนที่ใช้ในสมการที่ (7) อยู่ในรูปของผลคูณเชิงสเกลาร์ระหว่างเวกเตอร์ ดังนั้นในปริภูมิอันดับสูงเราสามารถจัดการกับข้อมูลในรูปของ $\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$. ถ้าปริภูมิของ H มีอันดับสูงจะทำให้ยุ่งยาก หรือใช้การคำนวณมาก อย่างไรก็ตามถ้าเรามีฟังก์ชันเคอร์เนลที่สามารถคำนวณ $k(\mathbf{x}_i,\mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$ แล้วเราสามารถใช้ฟังก์ชันนี้แทนที่ $\mathbf{x}_i\cdot\mathbf{x}_j$ ทุกๆ ที่ในการคำนวณและไม่ จำเป็นต้องรู้ฟังก์ชันที่ใช้แมป Φ จริงๆ ฟังก์ชันเคอร์เนลที่นิยมใช้ได้แก่

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

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



รูปที่ 2 แนวคิดการแมปข้อมูลแบบเป็นไม่เชิงเส้นไปสู่ปริภูมิอันดับสูง

2.1.2 ทฤษฎีสารสนเทศ

ค่าสารสนเทศของข้อมูลขึ้นอยู่กับความน่าจะเป็นของข้อมูล [10] ซึ่งสามารถวัดอยู่ในรูปของบิตจากสูตร

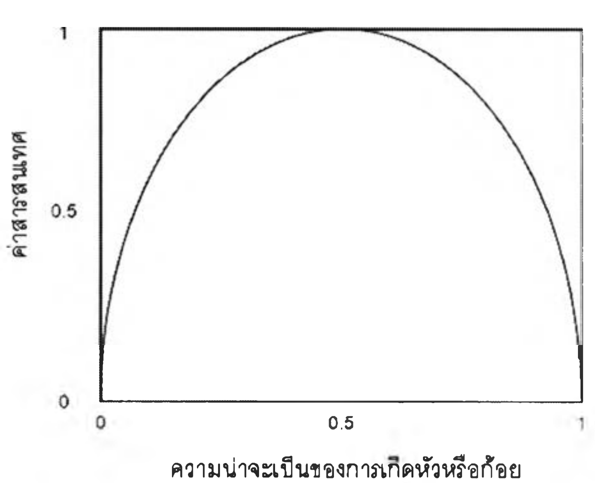
$$\text{ค่าสารสนเทศของข้อมูล} = -\log_2 \text{ความน่าจะเป็นของข้อมูล} \tag{13}$$

ถ้าให้ชุดของข้อมูล M ประกอบด้วยค่าที่เป็นไปได้คือ $\{m_1, m_2, ..., m_n\}$ และให้ความน่าจะเป็นที่จะเกิดค่า m_i มีค่าเท่ากับ $P(m_i)$ จะได้ว่าค่าสารสนเทศของ M หรือค่าเอนโทรปีของ M เขียนแทนด้วย $I(M)$ คำนวณได้จากสูตร

$$I(M) = \sum_{i=1}^{n} -P(m_i)\log_2 P(m_i) \tag{14}$$

ความหมายของค่าสารสนเทศกรณีมีค่าน้อย หมายถึงข้อมูลในชุดนั้นมีความแตกต่างกันน้อย หรือเกือบจะเป็นประเภทเดียวกัน แต่กรณีค่าสารสนเทศสูงจะบ่งบอกว่าข้อมูลในชุดนั้นมีความแตกต่างกันมากหรือประกอบด้วยตัวอย่างหลายประเภทที่มีจำนวนใกล้เคียงกัน ตัวอย่างในการโยนหัวโยนก้อย ชุดข้อมูล M จะประกอบด้วยค่าที่เป็นไปได้ {หัว, ก้อย} จะได้ว่าค่าสารสนเทศของการโยนหัวก้อยเท่ากับ $-P(\text{หัว})\log_2(P(\text{หัว})) - P(\text{ก้อย})\log_2(P(\text{ก้อย}))$

เมื่อพิจารณากรณีที่โยนออกหัวหมดหรือก้อยหมด ค่าสารสนเทศจะเป็น 0 และค่าสารสนเทศจะค่อยๆ เพิ่มขึ้นจนสูงที่สุด เมื่อความน่าจะเป็นของการเกิดหัวเท่ากับความน่าจะเป็นของการเกิดก้อยหรือสามารถแสดงได้ดังรูปที่ 3



รูปที่ 3 ความสัมพันธ์ระหว่างความน่าจะเป็นและค่าสารสนเทศ

2.2 งานวิจัยที่เกี่ยวข้อง

จากที่กล่าวข้างต้นการนำซัพพอร์ตเวกเตอร์แมชชีนมาแก้ปัญหาการจำแนกแบบหลายประเภทมีผู้เสนอวิธีการหลายวิธีด้วยกัน โดยส่วนใหญ่มักเป็นการนำฟังก์ชันที่จำแนกแบบสองประเภทหลายฟังก์ชันมาใช้ร่วมกันดังงานวิจัยที่เกี่ยวข้องต่อไปนี้

2.2.1 วิธีจำแนกแบบหนึ่งต่อที่เหลือ (One-against-the rest)

เป็นการสร้างระนาบเพื่อจำแนกข้อมูลสองส่วนโดยเปรียบเทียบข้อมูลประเภทหนึ่งกับประเภทอื่นที่เหลือทั้งหมดพร้อมกัน ดังนั้นจึงต้องสร้างระนาบหรือตัวจำแนกทั้งสิ้น k ตัว สำหรับปัญหา k ประเภท เมื่อ i คือประเภทข้อมูลใด ๆ ตัวจำแนกสำหรับประเภท ข้อมูล i เป็นไปตามเงื่อนไขต่อไปนี้

$$\begin{aligned}
&\min_{w^i,b^i,\xi^i} \quad \frac{1}{2}(\mathbf{w}^i)^T\mathbf{w}^i + C\sum_{j=1}^{l}\xi_j^i(\mathbf{w}^i)^T \\
&\qquad (\mathbf{w}^i)^T\Phi(\mathbf{x}_j)+b^i \geq 1-\xi_j^i, \quad \text{if } y_j = i \\
&\qquad (\mathbf{w}^i)^T\Phi(\mathbf{x}_j)+b^i \leq -1+\xi_j^i, \quad \text{if } y_j \neq i \\
&\xi_j^i \geq 0, \quad j=1,\dots,l.
\end{aligned} \tag{15}$$

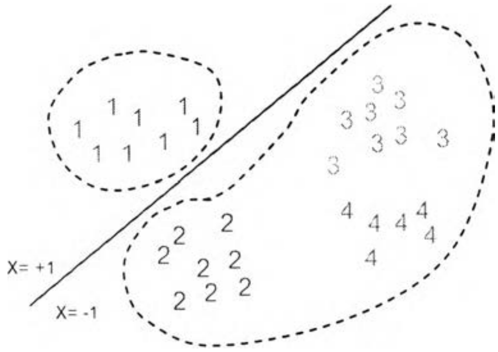
หลังจากหาผลเฉลยของสมการ (15) จะได้ฟังก์ชันที่จำแนกสำหรับ k ประเภท ดังนี้

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

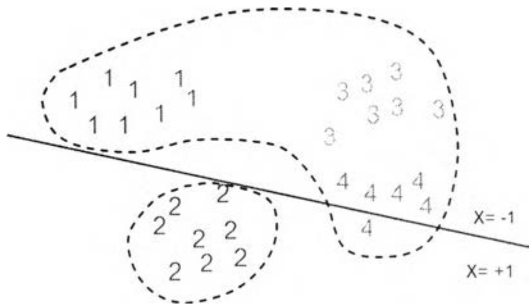
การพิจารณาคำตอบของการจำแนกจะเลือกจากตัวจำแนกที่ให้ค่าของฟังก์ชันตัวจำแนกสูงสุดจากตัวจำแนกทั้ง k ตัว

$$\text{class of } \mathbf{x} \equiv \arg\max_{i=1,\dots,k}((\mathbf{w}^i)^T\Phi(\mathbf{x})+b^i) \ . \tag{16}$$

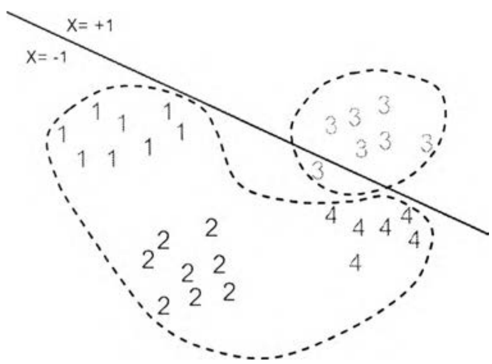
พิจารณาตัวอย่างการสร้างระนาบเพื่อจำแนกข้อมูลของวิธีจำแนกแบบหนึ่งต่อที่เหลือกรณีปัญหา 4 ประเภทดังรูปที่ 4



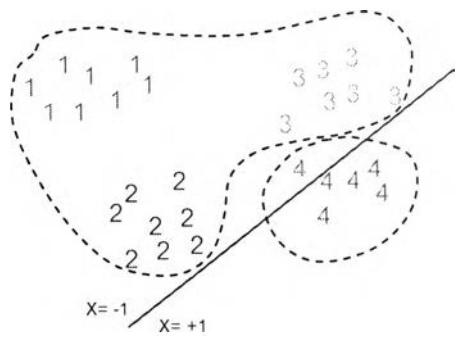
(ก) ตัวอย่างระนาบที่แบ่งข้อมูล 2 ส่วน ออกจากกัน โดยส่วนแรกคือ ประเภทที่ 1 และส่วนที่สองคือ ประเภทที่ 2, 3 และ 4



(ข) ตัวอย่างระนาบที่แบ่งข้อมูล 2 ส่วน ออกจากกัน โดยส่วนแรกคือ ประเภทที่ 2 และส่วนที่สองคือ ประเภทที่ 1, 3 และ 4



(ค) ตัวอย่างระนาบที่แบ่งข้อมูล 2 ส่วน ออกจากกัน โดยส่วนแรกคือ ประเภทที่ 3 และส่วนที่สองคือ ประเภทที่ 1, 2 และ 4



(ง) ตัวอย่างระนาบที่แบ่งข้อมูล 2 ส่วน ออกจากกัน โดยส่วนแรกคือ ประเภทที่ 4 และส่วนที่สองคือ ประเภทที่ 1, 2 และ 3

รูปที่ 4 ตัวอย่างการสร้างระนาบของการจำแนกแบบหนึ่งต่อที่เหลือ สำหรับปัญหา 4 ประเภท

ตัวอย่างการสร้างระนาบของการจำแนกแบบหนึ่งต่อที่เหลือ กรณีปัญหา 4 ประเภท ซึ่งจะสร้างระนาบทั้งสิ้น 4 ตัว ได้แก่ระนาบที่แบ่งส่วนข้อมูลดังต่อไปนี้

1. ระนาบแบ่งระหว่างส่วนที่เป็นประเภทที่ 1 และส่วนที่เป็นประเภทที่ 2, 3 และ 4
2. ระนาบแบ่งระหว่างส่วนที่เป็นประเภทที่ 2 และส่วนที่เป็นประเภทที่ 1, 3 และ 4
3. ระนาบแบ่งระหว่างส่วนที่เป็นประเภทที่ 3 และส่วนที่เป็นประเภทที่ 1, 2 และ 4
4. ระนาบแบ่งระหว่างส่วนที่เป็นประเภทที่ 4 และส่วนที่เป็นประเภทที่ 1, 2 และ 3

จากรูปที่ 4(ข-ง) จะเห็นว่าการหาระนาบที่แบ่งข้อมูลออกเป็นสองส่วน โดยตัวอย่างทุกตัวตกอยู่ในด้านของระนาบที่ถูกต้องนั้นเป็นไปได้ยาก โดยไม่ว่าจะสร้างระนาบอย่างไรก็จะมีตัวอย่างบางตัวตกอยู่ในด้านที่ไม่ถูกต้องเสมอ เมื่อนำระนาบที่ได้ไปใช้ในการจำแนกข้อมูลย่อมส่งผลต่อความถูกต้องของการจำแนกด้วย ซึ่งเป็นข้อด้อยของวิธีนี้ที่นอกจากใช้เวลาในการสร้างระนาบนานแล้วอาจไม่ได้ระนาบที่ดีในการจำแนกข้อมูลอีกด้วย

2.2.2 วิธีแมกซ์วิน (Max Wins)

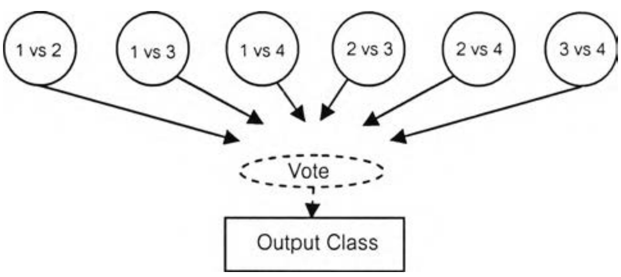
อัลกอริทึมแมกซ์วินจะทำงานโดยอาศัยฟังก์ชันพื้นฐานของการจำแนกแบบหนึ่งต่อหนึ่ง สร้างตัวจำแนกข้อมูลทั้งหมด จำนวน $k(k-1)/2$ ตัว โดยที่ k คือจำนวนประเภทข้อมูลทั้งหมด ตัวจำแนกแต่ละตัวจะถูกสอนโดยข้อมูลจาก 2 ประเภท i และ j ใดๆ ตามเงื่อนไขดังนี้

$$\min_{w^{ij},b^{ij},\xi^{ij}} \quad \frac{1}{2}(\mathbf{w}^{ij})^T\mathbf{w}^{ij} + C\sum_t \xi_t^{ij}(\mathbf{w}^{ij})^T$$

$$(\mathbf{w}^{ij})^T\Phi(\mathbf{x}_t) + b^{ij} \ge 1-\xi_t^{ij}, \quad \text{if } y_t = i$$

$$(\mathbf{w}^{ij})^T\Phi(\mathbf{x}_t) + b^{ij} \le -1+\xi_t^{ij}, \quad \text{if } y_t = j \;\; \xi_t^{ij} \ge 0. \qquad (17)$$

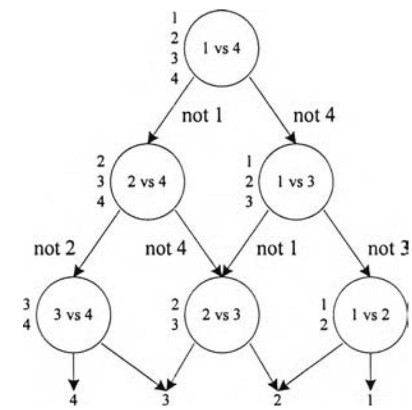
แล้วใช้ฟังก์ชัน $sign((\mathbf{w}^{ij})^T\Phi(\mathbf{x}) + b^{ij})$ จำแนกให้คำตอบว่าตัวอย่าง x ใดๆ อยู่ในประเภท i หรือ j หลังจากได้คำตอบทั้งหมดแล้ว วิธีแมกซ์วินจะนำคำตอบที่ได้จากตัวจำแนกทั้ง $k(k-1)/2$ ตัว มาโหวตรวมกันโดยประเภทที่ได้ผลโหวตสูงที่สุด ก็จะเป็นคำตอบของแมกซ์วิน (ดังแสดงในรูปที่ 5) ส่วนในกรณีที่มีผลโหวตสูงสุดมากกว่า 1 ประเภท ก็จะทำการสุ่มคำตอบจากประเภทเหล่านั้น



รูปที่ 5 วิธีแมกซ์วินสำหรับปัญหา 4 ประเภท

2.2.3 วิธีดีดีเอจี (Decision Directed Acyclic Graph (DDAG))

Platt และคณะ [3] ได้เสนอวิธีดีดีเอจี โดยใช้โทโปโลยีของกราฟมาวิเคราะห์ ซึ่งนำตัวจำแนกแบบสองประเภทหลายตัวมาใช้ร่วมกันเป็นตัวจำแนกแบบหลายประเภทหนึ่งตัว สำหรับปัญหาที่มี k ประเภท ในขั้นตอนการเรียนรู้จะสร้างตัวจำแนกแบบสองประเภทจำนวน $k(k-1)/2$ ตัว เช่นเดียวกับวิธีแมกซ์วิน แต่ในขั้นตอนการจำแนกจะใช้กราฟไม่มีวงแบบมีทิศทาง แต่ละโนดจะมีฟังก์ชันแบบสองประเภท ซึ่งมีทั้งหมด $k(k-1)/2$ โนดและมี k ใบ (ดูรูปที่ 6) แต่ละโนดคือตัวจำแนกข้อมูลของประเภทที่ i และ j เมื่อต้องการจำแนกข้อมูล x จะเริ่มจากโนดรากแล้วฟังก์ชันแบบสองประเภทที่โนดจะถูกนำมาคำนวณ ถ้าได้ค่าของฟังก์ชันน้อยกว่าศูนย์ก็ออกที่กิ่งซ้าย ถ้าได้มากกว่าหรือเท่ากับศูนย์ก็ออกที่กิ่งขวา หลังจากนั้นฟังก์ชันที่โนดต่อมาจะถูกคำนวณจนถึงโนดใบซึ่งก็คือคำตอบของดีดีเอจี

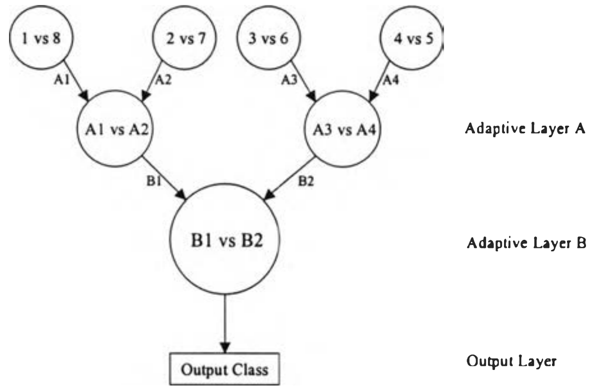


รูปที่ 6 โครงสร้างของดีดีเอจีสำหรับปัญหา 4 ประเภท

ใน [4] ได้แสดงให้เห็นถึงข้อเสียของดีดีเอจี อย่างแรกก็คือ ดีดีเอจีจะให้ผลลัพธ์ที่มีการแจกแจงความน่าจะเป็นไม่สม่ำเสมอ อันเนื่องมาจากการขึ้นอยู่กับลำดับของตัวจำแนกในกราฟ ซึ่งมีผลต่อความน่าเชื่อถือของอัลกอริทึม นอกจากนี้ประเภทที่ถูกต้องที่วางอยู่ใกล้โนดราก จะมีข้อเสียเมื่อเปรียบเทียบประเภทที่อยู่ใกล้โนดใบ เนื่องจากจะมีโอกาสที่จะถูกจำแนกมากครั้งกว่าทำให้โอกาสที่จะถูกจำแนกผิดมีมากขึ้น อย่างที่สองคือ จำนวนครั้งที่ประเภทที่ถูกต้องถูกจำแนกสูงทำให้มีความผิดพลาดสะสมสูง และส่งผลให้จำแนกผิดได้ เส้นทางการจำแนกของดีดีเอจีมีค่าความลึกเท่ากับ k-1 นั่นคือจำนวนครั้งที่ประเภทที่ถูกต้องถูกจำแนกกับประเภทอื่นจะแปรผันตามจำนวนประเภทข้อมูลทั้งหมด

2.2.4 วิธีเอดีเอจี (Adaptive Directed Acyclic Graph (ADAG))

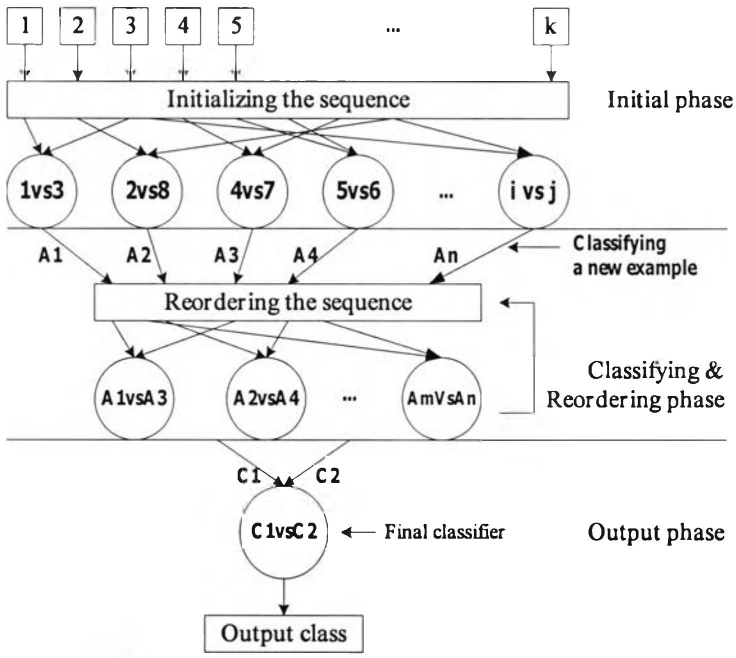
Ussivakul และ Kijsirikul [4] เสนอวิธีการที่ลดข้อเสียของวิธีดีดีเอจีเรียกว่า วิธีเอดีเอจี เป็นกราฟไม่มีวงแบบมีทิศทาง ที่ปรับโครงสร้างเป็นรูปคล้ายสามเหลี่ยมคว่ำ โดยจะเลือกตัวจำแนกแบบสองประเภท มาใช้จำแนกเพียง k-1 ตัวในปัญหา k ประเภท ในขั้นตอนการจำแนกจะมีการแบ่งเป็นชั้น โดยในแต่ละชั้นจะเลือกตัวจำแนกแบบสองประเภทมาจำนวน k/2 ตัวเพื่อจับคู่ให้ทุกประเภทข้อมูลถูกจำแนก 1 ครั้งในแต่ละชั้น และเมื่อจำแนกแล้วจำนวนประเภทข้อมูลจะลดลงเหลือครึ่ง หนึ่งในชั้นถัดมา จนเหลือประเภทเดียวในระดับล่างสุด (ดูรูปที่ 7)



รูปที่ 7 โครงสร้างของเอดีเอจีสำหรับปัญหา 8 ประเภท

จากโครงสร้างของวิธีเอดีเอจีทำให้ทุกประเภทข้อมูลมีโอกาสของจำนวนครั้งที่จะถูกจำแนกเท่ากัน จึงลดจำนวนครั้งที่ประเภทที่ถูกต้องถูกจำแนกกับประเภทอื่นจึงลดความผิดพลาดสะสมลงได้ อย่างไรก็ตามความถูกต้องของวิธีนี้ยังคงขึ้นกับลำดับการจับคู่เพื่อจำแนกของประเภทข้อมูลในแต่ละชั้น ในวิธีอาร์เอดีเอจี [5] จะเสนอวิธีการเลือกลำดับที่เหมาะสมในการจับคู่จำแนกข้อมูลในแต่ละชั้น เพื่อให้มีโอกาสจำแนกข้อมูลผิดในแต่ละชั้นน้อยที่สุด

2.2.5 **วิธีอาร์เอดีเอจี** (Reordering Adaptive Directed Acyclic Graph (RADAG))



รูปที่ 8 โครงสร้างการจำแนกข้อมูลของอาร์เอดีเอจี

Phetkaew, et al. [5] นำเสนอวิธีการใหม่ เรียกว่าอาร์เอดีเอจี เพื่อปรับปรุงความถูกต้องของวิธีเอดีเอจีเดิม [4] โดยในโครงสร้างของการจำแนกนั้นจะเป็นรูปคล้ายสามเหลี่ยมคว่ำเหมือนวิธีเอดีเอจี แต่จะมีการเลือกลำดับที่ทำให้เกิดความผิดพลาดน้อยที่สุดของการจำแนกในแต่ละชั้นโดยวิธีการเลือกจะใช้อัลกอริทึมของการจับคู่สมบูรณ์แบบน้ำหนักน้อยสุด (Minimum-Weight Perfect Matching) [6] เพื่อจับคู่ให้ค่าขอบเขตความผิดพลาดรวมในแต่ละชั้นมีค่าน้อยที่สุด โดยค่าขอบเขตความผิดพลาดของฟังก์ชันแบบสองประเภทแต่ละตัวหาได้จากค่าประสิทธิภาพโดยนัยทั่วไปของซัพพอร์ตเวกเตอร์แมชชีน (Generalization Performance of Support Vector Machine) [11] และเมื่อมีการจำแนกผ่านไปสู่ชั้นใหม่ก็จะมีการเรียงลำดับการจำแนกใหม่ (Reordering) อีกจนเหลือประเภทเดียวในชั้นสุดท้ายซึ่งเป็นคำตอบของอาร์เอดีเอจี

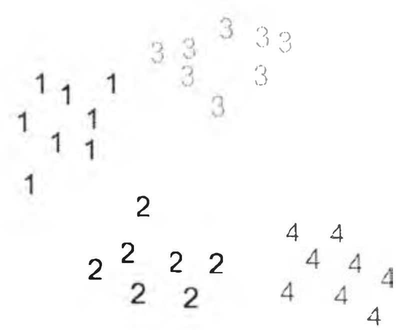
2.2.6 วิธีจำแนกข้อมูลแบบแตกครึ่งแบบสมดุล (Balanced Dichotomization Classification)

Kijsirikul, et al. [7] เสนอวิธีจำแนกข้อมูลแบบแตกครึ่งแบบสมดุล (Balanced Dichotomization) ซึ่งมีแนวคิดในการจำแนกข้อมูลโดยนำตัวจำแนกแบบสองประเภทมาสร้างเป็นต้นไม้ไบนารีสำหรับการจำแนกแบบหลายประเภท หลักเกณฑ์ในการเลือกตัวจำแนกมาสร้างโนดของต้นไม้จะพิจารณาเลือกตัวจำแนกที่แบ่งประเภทข้อมูลได้สมดุลที่สุด เมื่อกล่าวถึงตัวจำแนกแบบสองประเภทซึ่งอีกนัยหนึ่งคือ ระนาบหลายมิติที่แบ่งประเภทข้อมูลสองประเภทออกจากกัน การหาระนาบที่แบ่งข้อมูลได้สมดุลที่สุดคือ แบ่งแล้วให้จำนวนประเภทที่ตกอยู่แต่ละด้านของระนาบมีค่าใกล้เคียงกันมากที่สุด เทคนิคนี้ได้ใช้ฟังก์ชัน *UnBalancedScore* เป็นตัววัดค่าความสมดุลของระนาบจากสูตรต่อไปนี้

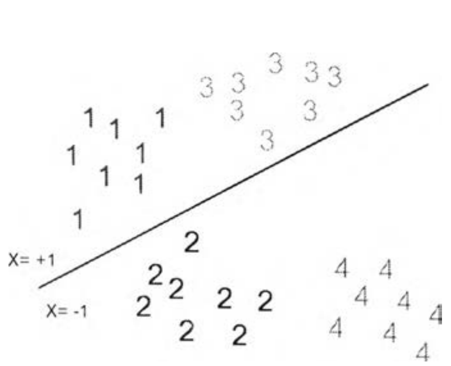
$$UnBalancedScore = CountLeft + CountRight + (0.5)\,|CountLeft - CountRight| \qquad (18)$$

โดยที่ *CountLeft* คือจำนวนประเภทข้อมูลที่มีตำแหน่งอยู่ในฝั่งซ้ายของระนาบและ *CountRight* คือจำนวนประเภทข้อมูลที่มีตำแหน่งอยู่ในฝั่งขวาของระนาบ ซึ่งระนาบที่ได้ค่าของฟังก์ชัน *UnBalancedScore* น้อยที่สุดก็จะถูกเลือกให้เป็นระนาบที่แตกครึ่งแบบสมดุลที่สุด

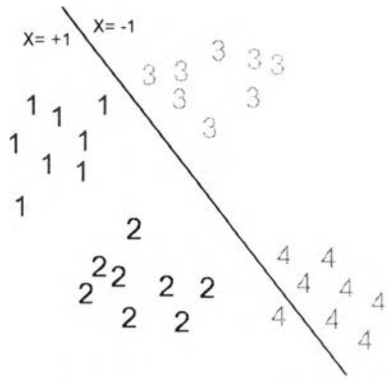
ตัวอย่างหากตำแหน่งของข้อมูลกรณีปัญหาการจำแนกจำนวน 4 ประเภทเป็น ดังรูปที่ 9 เมื่อใช้วิธีการแตกครึ่งแบบสมดุล ในแต่ละรอบต้องสร้างตัวจำแนกแบบสองประเภททุกแบบที่เป็นไปได้จากนั้นจึงคำนวณค่า *UnBalancedScore* ของตัวจำแนกแต่ละตัวเพื่อใช้ในการพิจารณาเลือกมาเป็นโนดของต้นไม้สำหรับการจำแนกแบบหลายประเภท (ดูรูปที่ 10 ประกอบ)



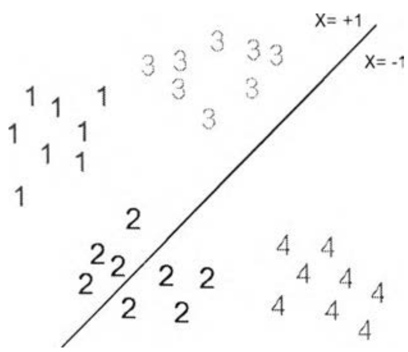
รูปที่ 9 ตัวอย่างตำแหน่งของข้อมูลกรณีปัญหา 4 ประเภท



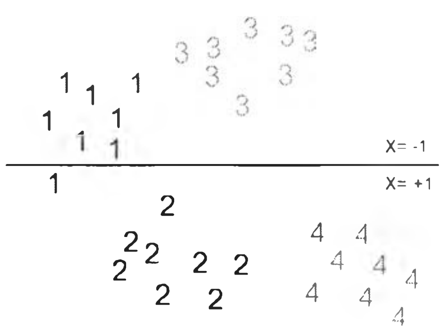
(ก) ตำแหน่งของข้อมูลประเภทอื่นๆ
เมื่อเทียบกับตัวจำแนกประเภท 1-2



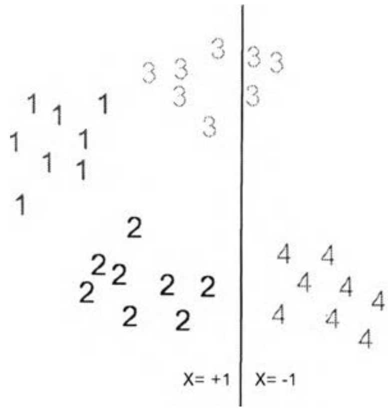
(ข) ตำแหน่งของข้อมูลประเภทอื่นๆ
เมื่อเทียบกับตัวจำแนกประเภท 1-3



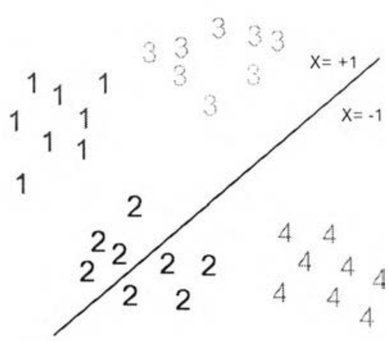
(ค) ตำแหน่งของข้อมูลประเภทอื่นๆ
เมื่อเทียบกับตัวจำแนกประเภท 1-4



(ง) ตำแหน่งของข้อมูลประเภทอื่นๆ
เมื่อเทียบกับตัวจำแนกประเภท 2-3



(จ) ตำแหน่งของข้อมูลประเภทอื่นๆ
เมื่อเทียบกับตัวจำแนกประเภท 2-4



(ฉ) ตำแหน่งของข้อมูลประเภทอื่นๆ
เมื่อเทียบกับตัวจำแนกประเภท 3-4

รูปที่ 10 การแบ่งข้อมูลด้วยตัวจำแนกแบบสองประเภททุกแบบที่เป็นไปได้ กรณีปัญหา 4 ประเภท

ค่า *UnBalancedScore* ของตัวจำแนก 1-2 = 2 + 2 - 0.5 | 2-2 | = 4

ค่า *UnBalancedScore* ของตัวจำแนก 1-3 = 3 + 2 - 0.5 | 3-2 | = 4.5

ค่า *UnBalancedScore* ของตัวจำแนก 1-4 = 3 + 2 - 0.5 | 3-2 | = 4.5

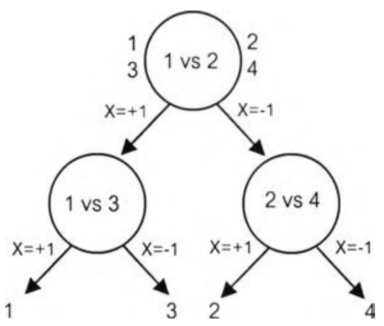
ค่า *UnBalancedScore* ของตัวจำแนก 2-3 = 3 + 2 - 0.5 | 3-2 | = 4.5

ค่า *UnBalancedScore* ของตัวจำแนก 2-4 = 3 + 2 - 0.5 | 3-2 | = 4.5

ค่า *UnBalancedScore* ของตัวจำแนก 3-4 = 3 + 2 - 0.5 | 3-2 | = 4.5

ดังนั้นตัวจำแนกที่ให้ค่า *UnBalancedScore* ต่ำสุดคือ 1-2 จึงได้ว่าโนดแรกหรือโนดรากของต้นไม้สำหรับการจำแนกแบบหลายประเภทของปัญหานี้ คือ ตัวจำแนกแบบสองประเภท1-2 นั่นเอง

ในรอบต่อๆ มาก็ใช้วิธีการเดียวกันโดยต้นไม้สำหรับการจำแนกแบบหลายประเภทที่สร้างจากค่า *UnBalancedScore* ของข้อมูลชุดดังกล่าว กรณีที่ดีที่สุดแสดงดังรูปที่ 11



รูปที่ 11 โครงสร้างการจำแนกข้อมูลของการแตกครึ่งแบบสมดุล กรณีปัญหา 4 ประเภท

เมื่อพิจารณารูปที่ 10(ค) และ 10(ง) พบว่าตัวจำแนก 1-4 และ 2-3 แบ่งประเภทข้อมูลด้านบวกมีจำนวนประเภทเป็น 3 และด้านลบมีจำนวนประเภทเป็น 2 ทั้งคู่ โดยเมื่อไปคำนวณค่า *UnBalancedScore* จึงให้ค่าที่เท่ากัน หากพิจารณาด้วยสายตาจะพบว่าตัวจำแนก 2-3 มีแนวโน้มที่ดีกว่าเนื่องจากแบ่งข้อมูลประเภทที่ 2 แล้วมีเพียง 1 ใน 8 ที่ตกอยู่ในด้านบวกขณะที่ข้อมูลประเภทเดียวกันที่เหลือตกอยู่ในด้านลบทั้งหมด แต่ตัวจำแนก 1-4 เมื่อใช้จำแนกข้อมูลประเภทที่ 2 พบว่าข้อมูลตกอยู่ทั้งด้านบวกและด้านลบอย่างละครึ่ง ตรงจุดนี้จะเห็นว่าฟังก์ชัน *UnBalancedScore* ไม่สามารถแยกได้ว่าตัวจำแนก 2 ตัวนี้ ตัวไหนที่มีแนวโน้มในการจำแนกประเภทได้ดีกว่ากันซึ่งนับเป็นจุดอ่อนอย่างหนึ่งของเทคนิคการแตกครึ่งแบบสมดุล

สำหรับวิธีนี้หากได้ระนาบที่แบ่งประเภทข้อมูลแล้วให้จำนวนประเภทที่ตกอยู่แต่ละด้านของระนาบมีค่าเท่ากันนั่นคือ ต้นไม้ที่ได้จะเป็นต้นไม้แบบสมดุลที่สุดและต้นไม้ต้นนั้นย่อมมีแนวโน้มที่จะให้ค่าเฉลี่ยของจำนวนครั้งในการจำแนกที่ต่ำด้วย วิธีนี้จะใช้ลดจำนวนครั้งในการจำแนกได้ดีในกรณีที่ความน่าจะเป็นในการเกิดข้อมูลแต่ละประเภทมีค่าใกล้เคียงกันซึ่งในทางปฏิบัติหาได้เป็นเช่นนั้นไม่ หากยังคงสร้างต้นไม้โดยเป็นต้นไม้ที่สมดุลอยู่ เมื่อนำต้นไม้นี้ไปใช้ก็จะให้จำนวนครั้งเฉลี่ยในการจำแนกประเภทข้อมูลสูงโดยไม่จำเป็น จึงเป็นที่มาของงานวิจัยชิ้นนี้ซึ่งจะได้อธิบายรายละเอียดของแนวคิดใหม่ในบทถัดไป