

### บทที่ 3

#### ซัพพอร์ตเวกเตอร์แมชีน

ซัพพอร์ตเวกเตอร์แมชีนเป็นเครื่องมือที่ช่วยในการเรียนรู้ มีการนำมาประยุกต์ใช้ในงาน 2 ประเภทหลัก คือ การแบ่งกลุ่ม (Classification) และ การวิเคราะห์การถดถอย (Regression) [11,12] โดยที่ผ่านมามีการนำซัพพอร์ตเวกเตอร์แมชีนทั้งสองประเภทไปใช้กับงานทางด้านวิศวกรรมมากมาย สำหรับงานวิจัยนี้ได้นำซัพพอร์ตเวกเตอร์แมชีนมาประยุกต์ใช้เพื่อการประมาณค่าฟังก์ชัน โดยเป็นการประมาณหรือทำนายค่ากำลังด้านย้อนกลับของระบบโทรศัพท์เคลื่อนที่ซีดีเอ็มเอ ซึ่งเป็นขั้นตอนที่สำคัญอย่างมากสำหรับการควบคุมกำลังของระบบโทรศัพท์เคลื่อนที่ ดังนั้นเนื้อหาในบทนี้จะกล่าวถึง การนำซัพพอร์ตเวกเตอร์แมชีนเพื่อวิเคราะห์การถดถอยเป็นหลัก ส่วนทฤษฎีของซัพพอร์ตเวกเตอร์สำหรับการแบ่งกลุ่มจะกล่าวไว้ในส่วนแรกพอสังเขป

#### 3.1 พื้นฐานเกี่ยวกับซัพพอร์ตเวกเตอร์แมชีน

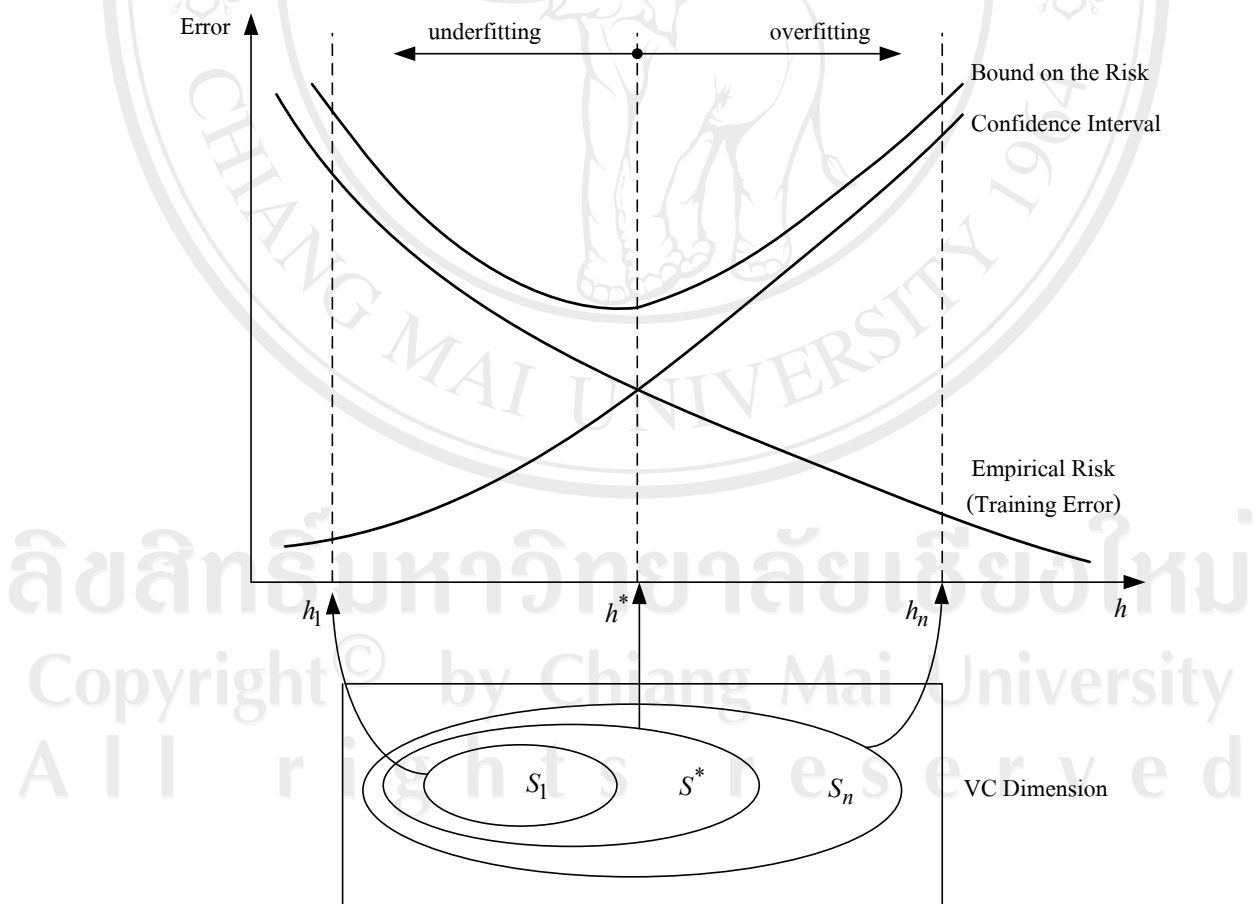
ซัพพอร์ตเวกเตอร์แมชีนเป็นเครื่องช่วยเรียนรู้ (Machine Learning Tasks) ที่ได้ถูกพัฒนาต่อเนื่องมาจากโครงข่ายประสาทเทียม (Artificial Neural Network) โดยใช้ความรู้พื้นฐานจากทฤษฎีการเรียนรู้ทางสถิติ (Statistical Learning Theory) [11-12,21-23] โครงสร้างของซัพพอร์ตเวกเตอร์แมชีนจะใช้หลักการลดความเสี่ยงเชิงโครงสร้างให้ต่ำสุด (Structural Risk Minimization : SRM) ต่างจากโครงข่ายประสาทเทียมที่ใช้หลักการลดความเสี่ยงทดลองให้ต่ำสุด (Empirical Risk Minimization : ERM) จากการออกแบบโครงสร้างอย่างมีหลักการนี้ทำให้ซัพพอร์ตเวกเตอร์มีประสิทธิภาพที่ดีกว่าโครงข่ายประสาทเทียม และถูกนำมาประยุกต์ใช้งานหลากหลายแบบ ทั้งด้านการแบ่งกลุ่มข้อมูล [13,20] เช่น การแยกลักษณะตัวอักษร การจดจำตัวอักษรที่เขียนด้วยมือ การแยกแยะภาพใบหน้านมนุษย์ การตัดสินใจที่จะเพิ่มหรือลดกำลังส่งของเครื่องลูกข่าย การแบ่งกลุ่มประเภทของสินค้า เป็นต้น ด้านการประมาณค่าฟังก์ชันหรือการทำนายค่า [14-17,23-25] เช่น การทำนายรูปแบบของยีน การพยากรณ์ความต้องการใช้ไฟฟ้า การพยากรณ์อัตราการแลกเปลี่ยนเงิน การค้นหาตำแหน่งผู้ประสบภัยทางทะเล เป็นต้น

### 3.1.1 หลักการลดความเสี่ยงเชิงโครงสร้างให้ต่ำสุด

หลักการลดความเสี่ยงเชิงโครงสร้างให้ต่ำสุด (Structure Risk Minimization: SRM) [11,12] ได้ถูกนำมาใช้ในการพัฒนาซอฟต์แวร์แมชชีน ซึ่งต่างจากหลักการเดิมคือ หลักการลดความเสี่ยงเชิงการทดลองให้ต่ำสุด (Empirical Risk Minimization: ERM) ที่ใช้กับโครงข่ายประสาท โดยมีหลักการพอสังเขปดังนี้

การวัดประสิทธิภาพของระบบสามารถวัดได้ในรูปของความเสี่ยง (Risk) โดยเมื่อทำการสอนระบบใดๆ แล้ว จะมีการหาความเสี่ยงที่เรียกว่า ความเสี่ยงจากการทดลอง (Empirical Risk:  $R_{emp}(\alpha)$ ) ซึ่งหาจากการนำข้อมูลชุดสอนมาทำการทดสอบ ดังสมการ 3.1 เมื่อ  $\mathbf{x}$  และ  $y$  คือ ข้อมูลขาเข้าและข้อมูลขาออกของชุดสอนซึ่งมีจำนวน  $\ell$  ค่า และ  $\alpha$  คือพารามิเตอร์ของฟังก์ชัน

$$R_{emp}(\alpha) = \frac{1}{2\ell} \sum_{i=1}^{\ell} |y_i - f(\mathbf{x}_i, \alpha)| \quad (3.1)$$



รูปที่ 3.1 ความสัมพันธ์ระหว่างความซับซ้อน (Complexity) ของระบบ

กับค่าความผิดพลาด (Error) [12]

เมื่อนำฟังก์ชันเชิงเส้นที่ได้มาทดสอบกับข้อมูลอีกชุดหนึ่ง ก็จะได้ข้อมูลขาออกซึ่งจะนำข้อมูลที่ได้อามาเปรียบเทียบกับค่าจริงเพื่อหาความผิดพลาดที่เกิดขึ้น โดยจะเรียกค่าความเสี่ยงที่ได้ว่า ความเสี่ยงที่แท้จริง (True Risk หรือ Actual Risk) ดังสมการ 3.2 เมื่อ  $R(\alpha)$  คือค่าความเสี่ยงที่แท้จริงและ  $P(\mathbf{x}, y)$  คือค่าความน่าจะเป็นของข้อมูลขาออก  $y$  เมื่อมีข้อมูลขาเข้าเป็น  $\mathbf{x}$

$$R(\alpha) = \int \frac{1}{2} |y - f(\mathbf{x}, \alpha)| dP(\mathbf{x}, y) \quad (3.2)$$

ซึ่งในการหาค่า  $P(\mathbf{x}, y)$  เป็นไปได้ยาก จึงมีการหาค่าความเสี่ยงที่แท้จริงโดยสามารถหาได้จากผลรวมของค่าความเสี่ยงจากการทดลองกับช่วงความเชื่อถือได้ (Confidence Interval) ซึ่งก็คือพจน์หลังของสมการ 3.3 โดยที่ค่าความเสี่ยงที่แท้จริงจะมีค่าไม่เกินผลรวมของทั้งสองพจน์นั้น เมื่อ  $h$  คือขนาดของปริภูมิ VC (Vapnik-Chervonenkis: VC Dimension) และ  $\eta$  คือความน่าจะเป็นหรือระดับความน่าเชื่อถือจะมีค่าในช่วง  $0 \leq \eta \leq 1$

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(\log(2\ell/h) + 1) - \log(\eta/4)}{\ell}} \quad (3.3)$$

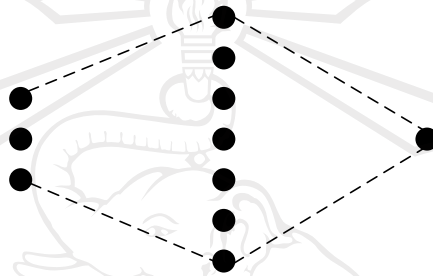
จากรูปที่ 3.1 เป็นกราฟที่แสดงถึงความสัมพันธ์ของปริภูมิ VC ความเสี่ยงจากการทดลอง (Empirical Risk) ช่วงความเชื่อถือได้ของระบบ (Confidence Interval) และขอบเขตของความเสี่ยง (Bound on the Risk) [12] ซึ่งจากกราฟพบว่า การลดความเสี่ยงเชิงทดลองให้ต่ำสุด (Empirical Risk Minimization) หรือค่าความผิดพลาดจากชุดสอน (Training Error) จะมีค่าน้อยที่ตำแหน่งปริภูมิ VC ที่ค่าสูง ๆ หรืออีกนัยหนึ่งก็คือ ถ้าข้อมูลชุดสอนมีขนาดใหญ่จะส่งผลให้ค่าความเสี่ยงเชิงทดลองต่ำลงแต่ในขณะเดียวกันค่าความเชื่อถือได้ (Confidence) ของระบบจะลดลง ซึ่งอาจเกิดเหตุการณ์ที่เรียกว่า การโอเวอร์ฟิต (Overfitting)

สำหรับซัพพอร์ตเวกเตอร์แมชีน จะใช้ความเสี่ยงเชิงโครงสร้าง ไม่ใช่การทดลองซึ่งเป็นการหาคำตอบโดยการแก้สมการและอสมการ โดยในการหาคำตอบซัพพอร์ตเวกเตอร์แมชีนจะพยายามลดความเสี่ยงเชิงโครงสร้างให้ต่ำที่สุด (Structural Risk Minimization: SRM)

$$\min_{S_h} R_{emp}(\alpha) + \sqrt{\frac{h(\log(2\ell/h) + 1) - \log(\eta/4)}{\ell}} \quad (3.4)$$

### 3.1.2 ฟังก์ชันเคอร์เนล (Kernel Function)

สำหรับการแก้ปัญหของข้อมูลที่มีลักษณะไม่เป็นเชิงเส้น (Nonlinear) สามารถทำการแมป (Mapping) ข้อมูลเหล่านี้โดยใช้ฟังก์ชันเคอร์เนล ทำให้ข้อมูลมีปริภูมิ (Dimension หรือ Space) ที่สูงขึ้นและได้ข้อมูลที่มีลักษณะเชิงเส้น (Linear) ดังรูป 3.2 เป็นการแมปข้อมูลจากปริภูมิอินพุต (Input Space) ผ่านฟังก์ชันเคอร์เนล จะได้ปริภูมิลักษณะเด่น (Feature Space) แล้วจึงทำการหาข้อมูลในปริภูมิเอาต์พุต (Output Space) ต่อไป



รูปที่ 3.2 การแมปข้อมูลโดยฟังก์ชันเคอร์เนล

ฟังก์ชันเคอร์เนลมีอยู่หลายแบบ แต่ละแบบก็จะเหมาะกับงานที่มีลักษณะแตกต่างกันไปในที่นี้ขอยกตัวอย่างดังตารางที่ 3.1

ตารางที่ 3.1 รูปแบบของฟังก์ชันเคอร์เนล [22]

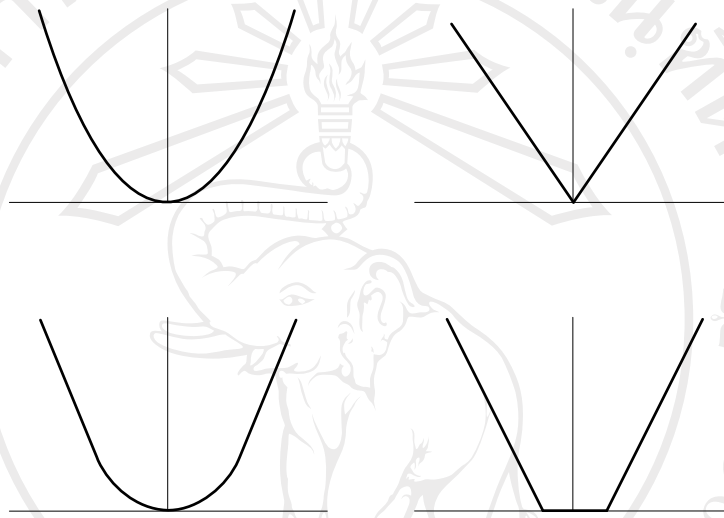
| รูปแบบของฟังก์ชันเคอร์เนล            | รูปแบบสมการ                                                                                       |
|--------------------------------------|---------------------------------------------------------------------------------------------------|
| เชิงเส้น                             | $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$                                           |
| โพลิโนเมียล                          | $K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x} \square \mathbf{x}' \rangle^d$                   |
| ฟังก์ชันเรเดียลเบสแบบเกาส์เซียน      | $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\ \mathbf{x} - \mathbf{x}'\ ^2}{2\sigma^2}\right)$ |
| ฟังก์ชันเรเดียลเบสแบบเอกซ์โพเนนเชียล | $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\ \mathbf{x} - \mathbf{x}'\ }{2\sigma^2}\right)$   |

ปริภูมิอินพุต

ปริภูมิ

### 3.1.3 ฟังก์ชันการสูญเสีย (Loss Function)

สำหรับซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอยที่มีหน้าที่ในการประมาณค่าฟังก์ชัน ซึ่งจะต้องมีการกำหนดส่วนเพื่อหรือค่าความคลาดเคลื่อนที่ยอมรับได้(Error Insensitive) ซึ่งอยู่ในรูปฟังก์ชันการสูญเสีย (Loss Function) จากรูป 3.3 [11] เป็นตัวอย่างของฟังก์ชันการสูญเสียและตารางที่ 3.2 เป็นรูปแบบสมการของฟังก์ชันการสูญเสีย



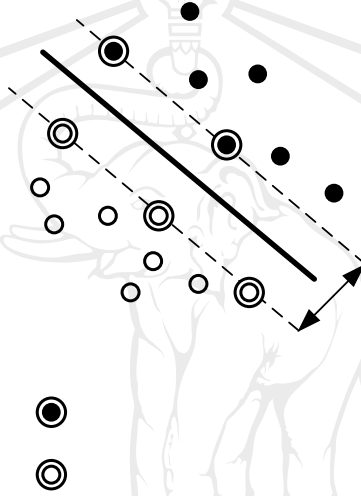
รูปที่ 3.3 รูปแบบฟังก์ชันการสูญเสีย (ก) Quadratic (ข) Laplace (ค) Huber (ง)  $\varepsilon$ -Insensitive

ตารางที่ 3.2 ฟังก์ชันการสูญเสีย

| ชื่อ                       | Density Model                                                                                                                                                      |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Quadratic                  | $L_{quad}(f(x) - y) = (f(x) - y)^2$                                                                                                                                |
| Laplace                    | $L_{Laplace}(f(x) - y) =  f(x) - y $                                                                                                                               |
| Huber                      | $L_{huber}(f(x) - y) = \begin{cases} \frac{1}{2}(f(x) - y)^2 & \text{for }  f(x) - y  < \mu \\ \mu f(x) - y  - \frac{\mu^2}{2} & \text{otherwise} \end{cases}$ (ก) |
| $\varepsilon$ -Insensitive | $L_{\varepsilon}(y) = \begin{cases} 0 & \text{for }  f(x) - y  < \varepsilon \\  f(x) - y  - \varepsilon & \text{otherwise} \end{cases}$                           |

### 3.2 ซัพพอร์ตเวกเตอร์แมชชีนสำหรับการแบ่งกลุ่ม (Support Vector Classification: SVC)

ซัพพอร์ตเวกเตอร์แมชชีนในยุคแรกถูกออกแบบมาเพื่อแก้ปัญหาการแบ่งกลุ่มของข้อมูล โดยเริ่มจากการแบ่งข้อมูลออกเป็น 2 กลุ่ม แล้วพัฒนาแบ่งข้อมูลเป็นหลายกลุ่มมากขึ้น สำหรับการแบ่งข้อมูล 2 กลุ่ม ซัพพอร์ตเวกเตอร์แมชชีนจะพยายามหาระนาบเกินที่เหมาะสมที่สุดซึ่งจะแยกข้อมูลทั้งสองกลุ่มให้ห่างกันมากที่สุด (Maximum Margin) ดังรูปที่ 3.4 ข้อมูลที่อยู่ตำแหน่งขอบของระนาบจะถูกเรียกว่า ซัพพอร์ตเวกเตอร์ โดยระนาบเกินที่ได้จะพยายามไม่ให้มีข้อมูลเกินเข้ามาในระหว่างระนาบทั้งสอง ซัพพอร์ตเวกเตอร์ที่ได้จะถูกใช้เป็นตัวแทนในแต่ละกลุ่ม



รูปที่ 3.4 ซัพพอร์ตเวกเตอร์แมชชีนสำหรับการแบ่งกลุ่ม

พิจารณาข้อมูลอินพุต  $\mathbf{x}_i$  โดยที่  $i = 1, 2, \dots, \ell$  เพื่อให้ได้ข้อมูลเอาต์พุตที่ถูกแบ่งออกเป็น 2 กลุ่ม  $y_i = \pm 1$  จะใช้ฟังก์ชันซิกนัม (Signum function:  $\text{sgn}$ ) ในการแบ่งกลุ่มตามสมการ 3.5 เมื่อ  $\mathbf{w}$  คือเวกเตอร์น้ำหนัก  $b$  คือค่าไบอัส และ  $\langle \cdot \rangle$  คือ การทำผลคูณแบบจุด โดยถ้า  $f(\mathbf{x}) > 0$  จะทำให้  $y_i = +1$  และ  $f(\mathbf{x}) < 0$  จะทำให้  $y_i = -1$

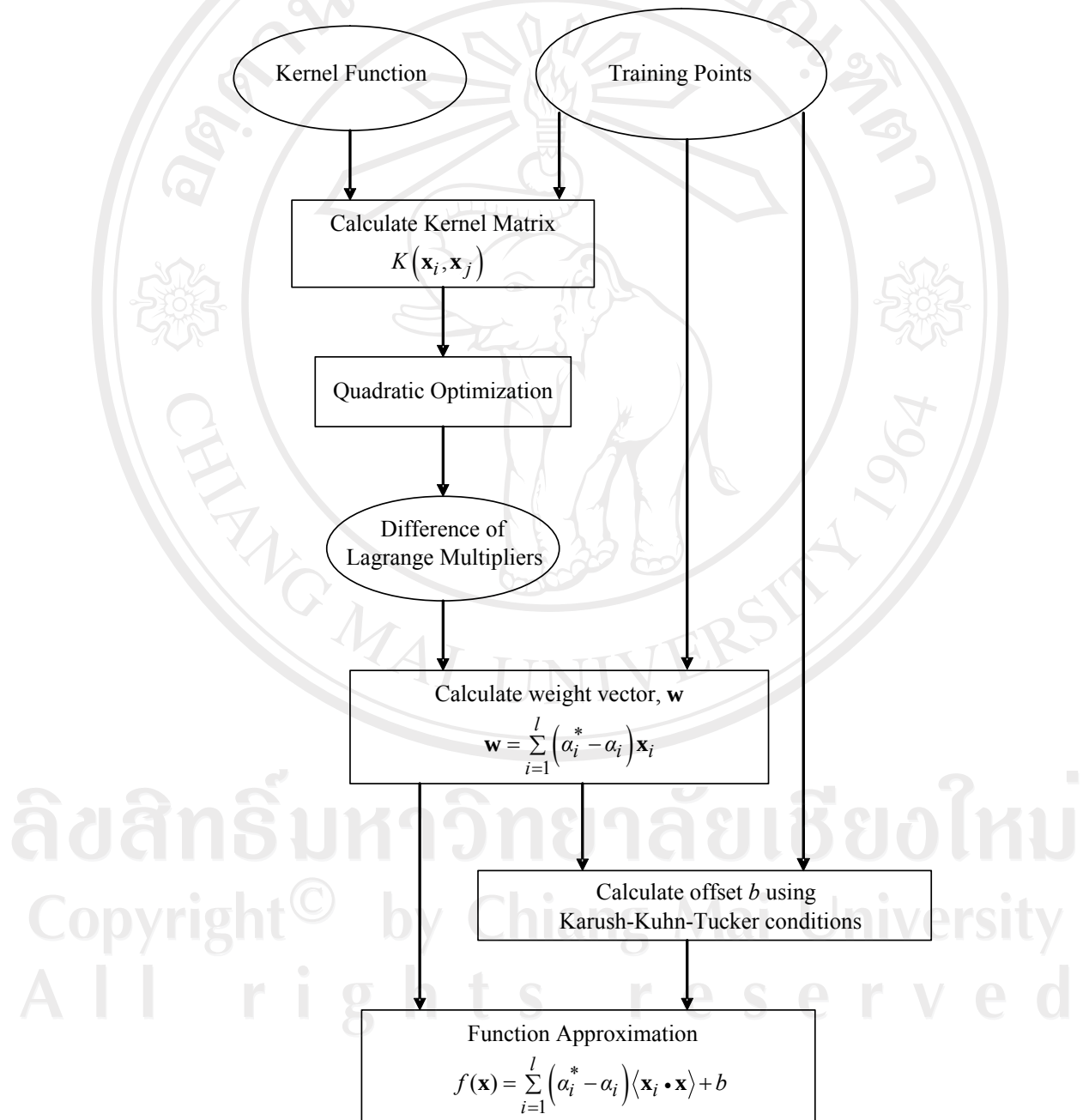
$$f(\mathbf{x}) = \text{sgn}(\langle \mathbf{w} | \mathbf{x} \rangle + b) \quad (3.5)$$

สำหรับกรณีที่ข้อมูลอินพุตไม่เป็นเชิงเส้น (Nonlinear) จะต้องมีการแมปข้อมูลให้ไปอยู่ในปริภูมิที่สูงขึ้นโดยใช้ฟังก์ชันเคอร์เนล เพื่อให้ข้อมูลเหล่านั้นกลายเป็นข้อมูลเชิงเส้นและสามารถวิเคราะห์ต่อไปได้ ดังสมการ 3.6 โดยที่  $\mathbf{x}_i$  คือซัพพอร์ตเวกเตอร์ และ  $K(\mathbf{x}_i, \mathbf{x})$  คือการแมปข้อมูลด้วยฟังก์ชันเคอร์เนล

$$f(\mathbf{x}) = \text{sgn} \left( \sum_{i=1}^{N_s} a_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right) \quad (3.6)$$

### 3.3 ซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอย (Support Vector Regression: SVR)

ซัพพอร์ตเวกเตอร์สำหรับการถดถอยได้ถูกนำมาในงานด้านวิศวกรรมหลายประเภท และในหลายๆ งานก็สามารถให้ผลลัพธ์ที่ดีกว่าวิธีการเดิม เช่น โครงข่ายประสาท เป็นต้น ซัพพอร์ตเวกเตอร์สำหรับการถดถอยแบ่งออกเป็น 2 แบบตามลักษณะข้อมูลขาเข้า คือ ซัพพอร์ตเวกเตอร์สำหรับการถดถอยเชิงเส้น และซัพพอร์ตเวกเตอร์สำหรับการถดถอยไม่เชิงเส้น



รูปที่ 3.5 ขั้นตอนของซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอย [22]



โดยลักษณะโครงสร้างของทั้งสองแบบเหมือนกัน แต่ต่างกันที่สำหรับซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอยไม่เชิงเส้น จะต้องมีการแมปข้อมูลให้ไปอยู่ในปริภูมิที่สูงกว่า และได้ข้อมูลที่เป็นลักษณะเชิงเส้น หลังจากนั้นก็สามารถทำตามขั้นตอนต่าง ๆ เหมือนกับซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอยเชิงเส้น รูปที่ 3.5 เป็นขั้นตอนการประมาณค่าฟังก์ชันของซัพพอร์ตเวกเตอร์แมชชีน โดยเป็นซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอยไม่เชิงเส้น

### 3.3.1 ซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอยเชิงเส้น (Linear Regression)

จุดประสงค์หลักของซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอยคือการหาฟังก์ชัน  $f(\mathbf{x})$  ซึ่งเป็นฟังก์ชันเชิงเส้นที่ใช้แทนกลุ่มข้อมูลที่นำมาฝึกสอนทั้งหมด โดยเริ่มจากการสอนซัพพอร์ตเวกเตอร์แมชชีนด้วยเซตข้อมูล  $\{\mathbf{x}_i, y_i\}_{i=1}^l, \mathbf{x}_i \in \mathcal{R}^n, y_i \in \mathcal{R}$  เมื่อ  $\mathbf{x}_i$  คือเวกเตอร์ข้อมูลอินพุต  $y_i$  คือข้อมูลเอาต์พุต และ  $l$  คือขนาดของข้อมูล จะได้ฟังก์ชันสำหรับการประมาณดังสมการ 3.7

$$f(\mathbf{x}) = \langle \mathbf{w} \square \mathbf{x} \rangle + b \quad (3.7)$$

โดยที่  $\mathbf{w}$  คือ เวกเตอร์น้ำหนัก  $b$  คือค่าไบอัส  $\langle \square \rangle$  คือ การทำผลคูณแบบจุด

สำหรับซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอย จะเป็นการหาตำแหน่งของข้อมูลที่เราเรียกว่า ซัพพอร์ตเวกเตอร์ ซึ่งจะอยู่ห่างจากระนาบเกินมากที่สุด เพื่อใช้เป็นตัวแทนของกลุ่มข้อมูลที่ฝึกสอนทั้งหมด หลังจากนั้นจะพยายามสร้างระนาบเกินให้อยู่ภายในระนาบทั้งสอง โดยพิจารณาจากซัพพอร์ตเวกเตอร์ที่ได้ ซึ่งจะทำให้ระนาบเกินที่ได้ ใช้เป็นฟังก์ชันประมาณค่าของกลุ่มข้อมูลทั้งหมด การหา  $\mathbf{w}$  ที่เหมาะสมที่สุด จะหาได้จากการหानอร์ม (Norm) ที่น้อยที่สุดของ  $\mathbf{w}$  โดยมีเงื่อนไขตามสมการ 3.9

$$\text{Minimize } \frac{1}{2} \|\mathbf{w}\|^2 \quad (3.8)$$

$$\left. \begin{aligned} y_i - \langle \mathbf{w} \square \mathbf{x}_i \rangle - b &\leq \varepsilon \\ \langle \mathbf{w} \square \mathbf{x}_i \rangle + b - y_i &\leq \varepsilon \end{aligned} \right\} \quad (3.9)$$

ในการสร้างระนาบเกินที่ประมาณค่าได้อย่างแม่นยำ เกิดจากการกำหนดความกว้างของระนาบที่เหมาะสม สามารถพิจารณาได้จากค่าความคลาดเคลื่อนที่ยอมรับได้ (Error Insensitive) ซึ่งอยู่ในรูปฟังก์ชันการสูญเสีย (Loss Function)



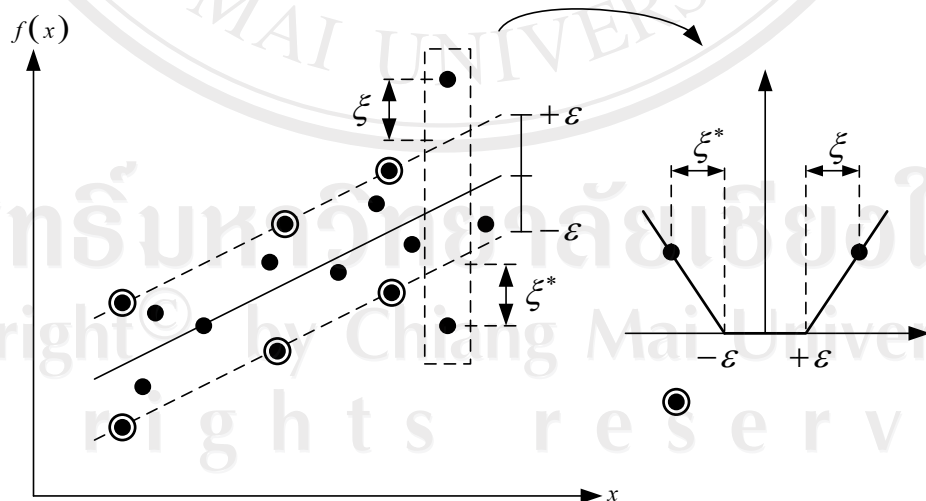
สำหรับการประมาณค่าฟังก์ชันของซัพพอร์ตเวกเตอร์แมชชีนมักอ้างอิงถึง ฟังก์ชันการสูญเสียแบบ  $\varepsilon$ -Insensitive ดังสมการ 3.10

$$L(y_i, f(\mathbf{x})) = \begin{cases} 0 & ; |y - f(\mathbf{x})| \leq \varepsilon \\ |y_i - f(\mathbf{x})| - \varepsilon & ; |y - f(\mathbf{x})| > \varepsilon \end{cases} \quad (3.10)$$

จะเห็นว่าฟังก์ชันการสูญเสียแบบ  $\varepsilon$ -Insensitive จะต้องมีการพิจารณาในส่วนของตัวแปร  $\zeta$  (Slack) ซึ่งเป็นค่าความคลาดเคลื่อนของข้อมูลที่อยู่นอกเหนือจากขอบระนาบทั้งสอง ทำให้สมการ 3.8 และ 3.9 มีรูปที่เปลี่ยนไป ตามสมการ 3.11 โดยมีเงื่อนไขตามสมการ 3.12 เมื่อ  $C$  คือค่าคงที่สำหรับควบคุมค่าคลาดเคลื่อน (Regularization Parameter) และ  $\zeta$ ,  $\zeta^*$  คือค่าคลาดเคลื่อนของข้อมูลจากขอบระนาบบนและขอบระนาบล่างตามลำดับ

$$\text{Minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\zeta_i + \zeta_i^*) \quad (3.11)$$

$$\left. \begin{aligned} y_i - \langle \mathbf{w} \square \mathbf{x}_i \rangle - b &\leq \varepsilon + \zeta_i \\ \langle \mathbf{w} \square \mathbf{x}_i \rangle + b - y_i &\leq \varepsilon + \zeta_i^* \\ \zeta_i, \zeta_i^* &\geq 0 \end{aligned} \right\} \quad (3.12)$$



รูปที่ 3.6 การหาระนาบเกินที่เหมาะสมที่สุด

ในการหาคำตอบจากสมการ 3.11 ภายใต้เงื่อนไขของสมการ 3.12 จะใช้ฟังก์ชันลากรางจ์ (Lagrange Function) เข้าช่วยโดยจะเพิ่มตัวคูณลากรางจ์ (Lagrange Multipliers) จึงทำให้ได้สมการ 3.13 เมื่อ  $L$  คือ Lagrangian และ  $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$  คือตัวคูณลากรางจ์

$$L = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) - \sum_{i=1}^{\ell} \alpha_i (\varepsilon + \xi_i - y_i + \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - \sum_{i=1}^{\ell} \alpha_i^* (\varepsilon + \xi_i^* - y_i + \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - \sum_{i=1}^{\ell} (\eta_i \xi_i + \eta_i^* \xi_i^*) \quad (3.13)$$

จากนั้นทำการแก้สมการที่ได้ด้วยวิธีโปรแกรมกำลังสอง (Quadratic Programming) โดยการหาอนุพันธ์ย่อย (Partial Derivatives) เทียบกับตัวแปรที่ต้องการหาทั้งสี่ตัวแล้วให้เท่ากับศูนย์ จะได้คำตอบดังสมการ 3.14

$$\left. \begin{aligned} \frac{\partial L}{\partial b} &= \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) = 0 \\ \frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) \mathbf{x}_i = 0 \\ \frac{\partial L}{\partial \xi_i} &= C - \alpha_i - \eta_i = 0 \\ \frac{\partial L}{\partial \xi_i^*} &= C - \alpha_i^* - \eta_i^* = 0 \end{aligned} \right\} \quad (3.14)$$

นำสมการ 3.14 ไปแทนในฟังก์ชันลากรางจ์ จะได้คำตอบที่เหมาะสมดังสมการ 3.15

$$\text{Maximize } -\frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle - \varepsilon \sum_{i=1}^{\ell} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{\ell} y_i (\alpha_i - \alpha_i^*) \quad (3.15)$$

ซึ่งต้องกระทำภายใต้เงื่อนไข 3.16

$$\sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) = 0, \quad (\alpha_i - \alpha_i^*) \in [0, C] \quad (3.16)$$

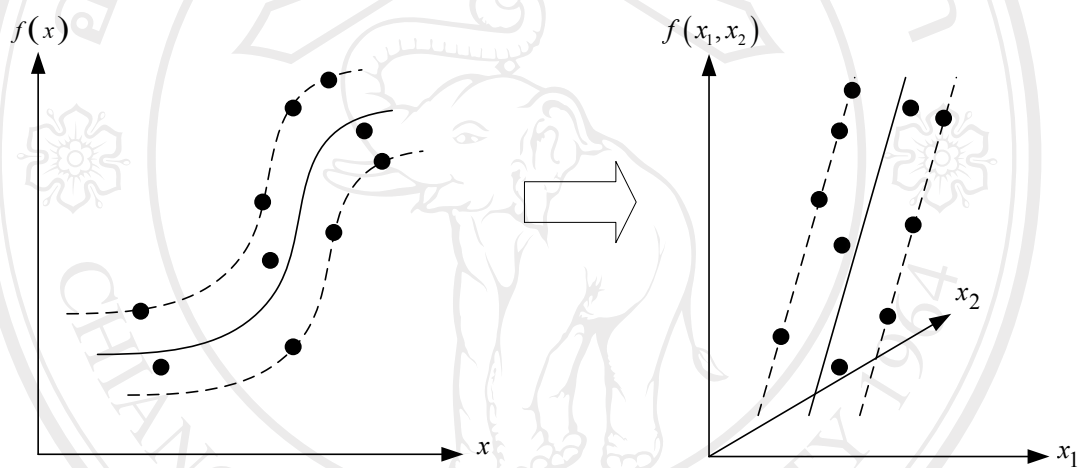
และจากสมการ 3.14 ซึ่ง  $\mathbf{w} = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) \mathbf{x}_i$  จะทำให้ได้สมการระนาบเกินใหม่เป็น

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b \quad (3.17)$$

### 3.3.2 ซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอยไม่เชิงเส้น (Nonlinear Regression)

ในกรณีที่ชุดข้อมูลเป็นลักษณะไม่เชิงเส้น จำเป็นต้องมีการใช้ฟังก์ชันเคอร์เนล (Kernel Function) ทำการส่งผ่านหรือแมป (Mapping) ข้อมูลที่ไม่เชิงเส้นให้มีปริภูมิ (Dimension หรือ Space) ที่สูงขึ้นเพื่อให้ข้อมูลมีลักษณะเป็นเชิงเส้น [11,21-22,26] แล้วสามารถใช้วิธีการต่างๆ เช่นเดียวกับข้อมูลเชิงเส้นดังที่กล่าวผ่านมา เมื่อ  $\mathbf{x}$  คือเวกเตอร์อินพุต  $\mathbf{x}_i$  คือซัพพอร์ตเวกเตอร์ และ  $\Phi$  คือฟังก์ชันการแมป จะได้ฟังก์ชันเคอร์เนลดังสมการ 3.18

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i) | \Phi(\mathbf{x}_j) \rangle \quad (3.18)$$



รูปที่ 3.7 การแมปข้อมูลที่ไม่เป็นเชิงเส้นในปริภูมิอินพุตไปเป็นข้อมูลเชิงเส้นในปริภูมิลักษณะเด่น

จากการแมปข้อมูลด้วยฟังก์ชันเคอร์เนล จะทำให้ได้สมการน้ำหนักใหม่คือ

$$\mathbf{w} = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) K(\mathbf{x}_i) \quad (3.19)$$

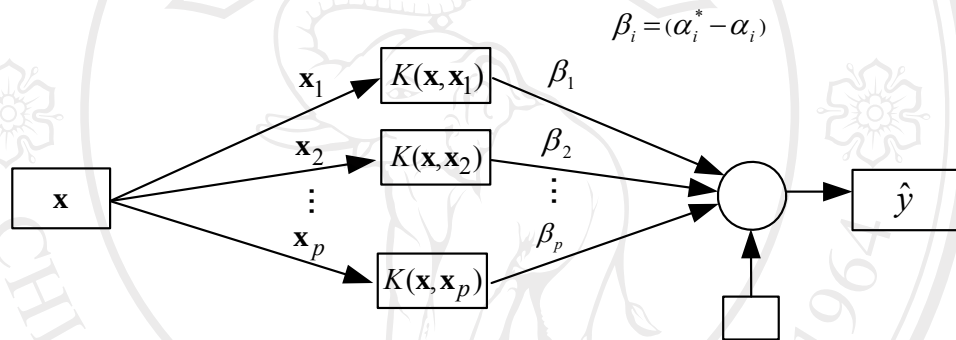
และเมื่อนำ  $\mathbf{w}$  ไปแทนค่าจะได้สมการระนาบเกินที่เหมาะสมที่สุดในรูปแบบใหม่คือ

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) K(\mathbf{x}, \mathbf{x}_i) + b \quad (3.20)$$

การหาค่าไบอัส  $b$  ที่เหมาะสมจะใช้หลักการของ KKT (Karush-Kuhn-Tucker) เพื่อปรับค่าที่อยู่ตรงกลางระหว่างขอบบนและขอบล่างให้มีความเหมาะสม ซึ่งแสดงดังสมการ 3.21 เมื่อ  $\mathbf{x}_r$  และ  $\mathbf{x}_s$  คือซัพพอร์ตเวกเตอร์ที่อยู่บนขอบบนและขอบล่างตามลำดับ

$$b = -\frac{1}{2} \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) (K(\mathbf{x}_i, \mathbf{x}_r) + K(\mathbf{x}_i, \mathbf{x}_s)) \quad (3.21)$$

สามารถสรุปภาพรวมของสถาปัตยกรรมของซัพพอร์ตเวกเตอร์แมชีนสำหรับการถดถอยได้ดังรูปที่ 3.8 เมื่อ  $\hat{y}$  คือผลจากการประมาณค่าโดยมี  $\mathbf{x}$  เป็นข้อมูลขาเข้า  $\mathbf{x}_i$  คือซัพพอร์ตเวกเตอร์  $\beta_i$  คือค่าน้ำหนัก และ  $b$  คือค่าไบอัส



รูปที่ 3.8 สถาปัตยกรรมของซัพพอร์ตเวกเตอร์แมชีนสำหรับการถดถอย