บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงรายละเอียดของทฤษฎีที่เกี่ยวข้องในงานวิจัยสำหรับการสร้างกลับจุดที่หายไปด้วยฟัซซีซัพพอร์ตเวกเตอร์แมชีนแบบถดถอย และซัพพอร์ตเวกเตอร์แมชีนแบบถดถอย ซึ่งใช้สำหรับการจำลองวัตถุ 3 มิติในคอมพิวเตอร์ด้วยเครื่องสแกน 3 มิติแบบเลเซอร์

2.1 ซัพพอร์ตเวคเตอร์แมชีนแบบถดถอย (Support Vector Regression: SVR)

ซัพพอร์ตเวคเตอร์แบบถดถอย (SVR) เป็นสถาปัตยกรรมที่ใช้ซัพพอร์ตเวคเตอร์ (Support vector) เพื่อเป็นตัวแทนของกลุ่มข้อมูลที่ใช้สำหรับเรียนรู้ โดย SVR จะเป็นการนำข้อมูลปัจจุบันและข้อมูลในอดีตจำนวนหนึ่ง มาทำการฝึกให้ระบบเรียนรู้รูปแบบ (Pattern) เพื่อการคาดการณ์ผลของค่าที่จะเกิดขึ้นในอนาคต ภายใต้ความเชื่อที่ว่าลักษณะที่เคยเกิดขึ้นในอดีตอาจเกิดขึ้นได้อีกในอนาคต ดังนั้นการคำนวณหาจุดที่หายไปได้นำหลักการข้างต้นมาใช้ โดยทำการเรียนรู้ระบบด้วยลักษณะของข้อมูลของจุดที่มีอยู่แล้วไปทำนายจุดที่หายไป

2.2 หลักการหาระนาบเชิงเส้น

กำหนดให้ $\{x_i, y_i\}_{i=1}^{l}$ โดยที่ $x_i \in \Re^n, y_i \in \Re$ เมื่อ x_i คือเวคเตอร์ข้อมูลอินพุตและ y_i คือข้อมูลเอาต์พุต เพื่อหาฟังก์ชันเชิงเส้น $f(x)$ ที่ใช้แทนสมการระนาบเกิน (Hyper plane) ของกลุ่มข้อมูลที่นำมาฝึกสอนทั้งหมดดังสมการ (1)

$$f(x) = (w \bullet x) + b \tag{2.1}$$

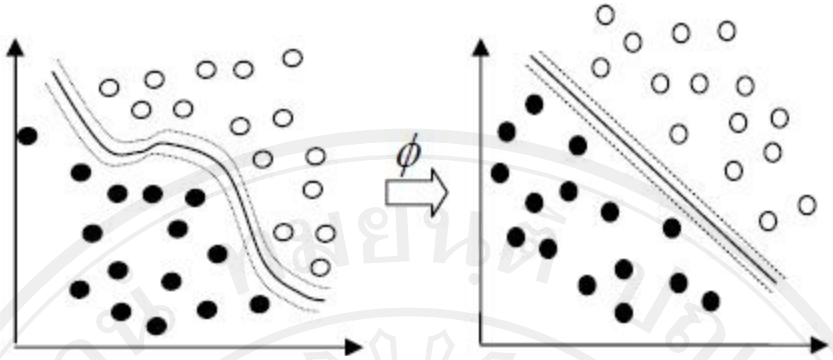
เมื่อ w คือ เวคเตอร์น้ำหนัก

b คือ ค่าไบอัส

$\bullet$ คือ การหาผลคูณแบบจุดของเวคเตอร์ (Dot Product)

x คือ เวคเตอร์ข้อมูลอินพุตในรูปของเวคเตอร์หลัก (Column Vector)

ดังนั้นเมื่อมีเวคเตอร์ x ใด ๆ ก็จะสามารถประมาณค่า y ได้โดยใช้สมการที่ (2.1) นั่นคือ $f(x) = y'$



รูปที่ 2.1 การวิเคราะห์การถดถอยแบบไม่เป็นเชิงเส้นในปริภูมิอินพุต
และการวิเคราะห์การถดถอยเชิงเส้นในปริภูมิลักษณะเด่น

ก่อนการสร้างระนาบเกิน (Hyper Plane) สถาปัตยกรรมของซัพพอร์ตเวคเตอร์แมชชีนได้มีการออกแบบให้มีการแมปข้อมูล (Data Mapping) จากปริภูมิอินพุตไปอยู่ในปริภูมิที่สูงขึ้นเรียกว่าปริภูมิลักษณะเด่นด้วยฟังก์ชันเคอร์เนล (Kernel Function) เพื่อให้สามารถหาฟังก์ชันเชิงเส้นที่เหมาะสมสำหรับใช้แทนสมการระนาบเกินได้ดังรูปที่ 2.1 ซึ่งแสดงถึงความแตกต่างในการวิเคราะห์การถดถอยแบบไม่เป็นเชิงเส้นในปริภูมิอินพุตและการวิเคราะห์การถดถอยเชิงเส้นในปริภูมิลักษณะเด่น ทำให้เวคเตอร์ข้อมูลอินพุตเปลี่ยนไปเป็นข้อมูลที่มีมิติสูงขึ้นในปริภูมิลักษณะ-เด่นเป็นต้น

2.3 ฟังก์ชันเคอร์เนล (Kernel Function)

การแมปข้อมูล (Data Mapping) ด้วยฟังก์ชันเคอร์เนล (Kernel Function) แสดงดังสมการที่ (2.2) เมื่อ ϕ คือ ฟังก์ชันการแมป (Mapping Function) นั่นคือ $x_i \rightarrow K(x_i, x_j)$ ทำให้สมการที่ (2.1) สามารถเขียนใหม่ได้เป็นดังสมการที่ (2.3)

$$K(x_i, x_j) = (\phi(x_i), \phi(x_j)) \tag{2.2}$$

$$f(x) = (w \bullet K(x_i, x_j)) + b \tag{2.3}$$

การสร้างระนาบเกิน $f(x)$ เพื่อประมาณค่าให้ใกล้เคียงกับ y ที่สุดจะต้องหาค่า w และ b ที่เหมาะสมที่สุด ฟังก์ชันเคอร์เนลที่ใช้ในการเปรียบเทียบผลในงานวิจัยนี้ได้แก่

<u>ฟังก์ชันเชิงเส้น (Linear)</u>

$$K(x_i, x_j) = x_i^T x_j \tag{2.4}$$

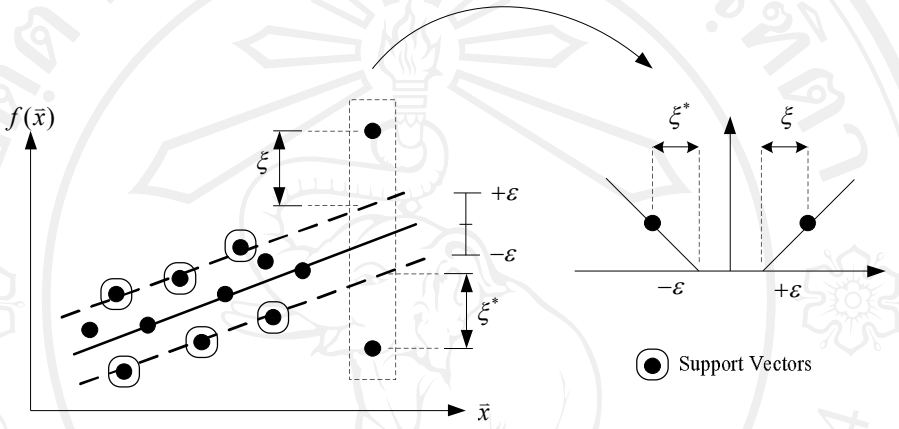
ฟังก์ชันโพลิโนเมียล (Polynomial)

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \tag{2.5}$$

ฟังก์ชันเกาส์เซียนเรเดียลเบสิส (Gaussian Radial basis function: RBF)

$$K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2), \sigma > 0 \tag{2.6}$$

2.4 ฟังก์ชันการสูญเสีย (Loss Function)



รูปที่ 2.2 การหาระนาบเกินที่เหมาะสมที่สุดสำหรับใช้แทนกลุ่มข้อมูล
และการกำหนดแนวขอบระนาบเกินด้วยฟังก์ชัน ε - insensitive

การสร้างระนาบเกินสำหรับกรณีของ SVR จะพิจารณาฟังก์ชันการสูญเสีย (Loss Function) ซึ่งเป็นค่าความคลาดเคลื่อนที่ยอมรับได้ สำหรับงานวิจัยนี้ได้ศึกษาการใช้ ε -Insensitive ซึ่งเป็นฟังก์ชันการสูญเสียที่นิยมใช้กันมากดังรูปที่ 2.2 มีความสัมพันธ์ดังสมการที่ (2.7)

$$L(y_i, f(x)) = \begin{cases} 0 & for |y_i - f(x)| \le \varepsilon \\ |y_i - f(x)| - \varepsilon & otherwise \end{cases} \tag{2.7}$$

เมื่อ $L(y_i, f(x))$ คือ ความคลาดเคลื่อนที่เกิดขึ้น ซึ่งเป็นความแตกต่างระหว่าง y_i และ $f(x)$ โดยความหมายของสมการ (2.7) คือ ค่าความคลาดเคลื่อนที่น้อยกว่าหรือเท่ากับ ε ให้ถือว่าเป็นศูนย์และเป็นค่าความคลาดเคลื่อนที่ผู้ใช้งานยินยอมให้เกิดขึ้นได้

ตัวอย่างเช่น $\varepsilon = 0.001$ หมายความว่า ผู้ทำการทดลองยอมรับผลการทำนายในสองหลักแรก และถือว่าการทำนายในหลักที่ 3 ไม่ค่อยมีความสำคัญและสามารถคลาดเคลื่อนได้เท่ากับ 0.001 นั่นเอง

2.5 FSVR (Fuzzy Support Vector Regression: FSVR)

เช่นเดียวกับ SVR ที่มีการประยุกต์ใช้กับงาน 2 รูปแบบ คือ การแบ่งกลุ่ม (Classification) และการประมาณค่าฟังก์ชัน (Regression) FSVR นั้นถูกพัฒนาขึ้นมาจาก SVR โดยแตกต่างกันตรงที่มีการกำหนดค่าความเป็นสมาชิกให้กับข้อมูลที่ใช้สำหรับการเรียนรู้ดังที่กล่าวไปแล้วข้างต้น ในงานวิจัยนี้ได้ศึกษาเกี่ยวกับการประมาณค่าฟังก์ชันโดยใช้ FSVR หรือที่เรียกว่า ฟัซซีซัพพอร์ตเวคเตอร์แมชีน สำหรับการถดถอย (Fuzzy Support Vector Regression: FSVR) นั่นเอง

กำหนดให้ $\{x_i, y_i, s_i\}_{i=1}^{l}$ โดยที่ $x_i \in \Re^n, y_i \in \Re$ เมื่อ x_i คือเวคเตอร์ข้อมูลอินพุต y_i คือข้อมูลเอาต์พุตและ s_i คือ ค่าความเป็นสมาชิกของ x_i แต่ละตัว โดย s_i จะอยู่ในรูป $\{\lambda \le s_i \le 1\}_{i=1}^{l}$ และ $\lambda > 0$ ดังนั้นในการสร้างฟังก์ชันเชิงเส้น $f(x)$ จะต้องทำการหาเวคเตอร์น้ำหนัก w และค่าไบอัส b เพื่อการประมาณค่า $f(x)$ ที่ใกล้เคียงที่สุดโดยพิจารณาค่าความคลาดเคลื่อนที่ยอมรับได้จาก ε-Insensitive ซึ่งแสดงไว้ในสมการที่ (2.7) การหาเวคเตอร์น้ำหนัก w ที่เหมาะสมที่สุดสามารถเขียนแทนด้วยสมการที่ (2.8)

$$\min_{w,b,\xi,\xi^*} C\sum_{i=1}^{n} s_i(\xi_i + \xi_i^*) + \frac{1}{2}w^T w$$

โดยมีเงื่อนไขคือ

(2.8)

$$\left\{\begin{array}{c} y_i - w^T\phi(x_i) - b_i \le \varepsilon + \xi_i, \quad \xi_i \ge 0 \\ w^T\phi(x_i) + b_i - y_i \le \varepsilon + \xi_i^*, \quad \xi_i^* \ge 0 \\ i = 1,\ldots,n \end{array}\right\}$$

โดยที่ w^T คือ การทรานสโพสเวคเตอร์น้ำหนัก w

C คือ ค่าคงที่สำหรับคุมค่าความคลาดเคลื่อนในการสร้าง $f(x)$

ξ_i, ξ_i^* คือ ค่าความคลาดเคลื่อนของข้อมูลจากขอบระนาบบนและล่างตามลำดับ

สำหรับข้อมูลที่ไม่เป็นเชิงเส้น จะต้องทำการแมป (Mapping) ข้อมูลอินพุตไปยังปริภูมิลักษณะเด่น (Feature Space) โดยการเลือกใช้ฟังก์ชันเคอร์เนลในหัวข้อที่ 2.3 แล้วทำการแก้สมการที่ (2.8) โดยใช้ขั้นตอนวิธีการโปรแกมกำลังสอง (Quadratic Programming: QP) จะสามารถแก้ปัญหาดังนี้

$$\max \begin{matrix} -\varepsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)y_i \\ -\frac{1}{2}\sum_{i,j}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j) \end{matrix} \tag{2.9}$$

โดยมีเงื่อนไขคือ

$$\left\{ \begin{matrix} \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0 \\ \alpha_i,\alpha_i^* \in [0,s_i,C] \end{matrix} \right\}$$

โดยที่ $K(x_i,x_j)$ คือ ฟังก์ชันเคอร์เนลใด ๆ

α_i,α_i^* คือ ตัวคูณลากรานจ์ (Lagrange Multipliers)

ตามเงื่อนไขข้างต้นเมื่อทำการแก้สมการโดยใช้วิธีโปรแกรมกำลังสองจะสามารถหาเวคเตอร์น้ำหนัก w ได้ดังนี้

$$w=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)K(x_i,x_j) \tag{2.10}$$

ดังนั้นสามารถเขียนสมการที่ (2.1) ในรูปสมการระนาบเกินได้เป็น

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)K(x_i,x_j)+b \tag{2.11}$$

ซึ่ง α_i,α_i^* และ b ได้จากการสอนด้วยชุดฝึกสอน (Training data)

2.6 ฟังก์ชันความเป็นสมาชิก (Membership Function)

ฟังก์ชันความเป็นสมาชิกโดยทั่วไปจะมีค่าระหว่าง 0 ถึง 1 การเลือกฟังก์ชันความเป็นสมาชิก จะต้องเลือกตามความเหมาะสมและความครอบคลุมของข้อมูลที่จะรับเข้ามา ซึ่งสามารถมีความเป็นสมาชิกหลายค่าได้ ในงานวิจัยนี้ได้ศึกษาฟังก์ชันความเป็นสมาชิกที่ขึ้นกับเวลา โดยกำหนดให้ t_i เป็นฟังก์ชันของเวลา ซึ่งสามารถเขียนให้อยู่ในรูปของ $s_i=f(t_i)$ และ $t_1 \le ... \le t_n$ คือ เวลาที่เข้ามาในระบบ โดยกำหนดให้ข้อมูลที่ x_n มีความสำคัญมากที่สุด และให้ x_1 มีความสำคัญน้อยที่สุดซึ่งจะสามารถแสดงเป็นสมการได้ดังนี้

$$s_n = f(t_n) = 1$$
$$s_1 = f(t_1) = \lambda \tag{2.12}$$

กำหนดให้ λ เป็นขอบเขตล่างของฟังก์ชันความเป็นสมาชิก

$$s_i = f(t_i) = \frac{1-\lambda}{t_n - t_1} t_i + \frac{t_n \lambda - t_1}{t_n - t_1} \tag{2.13}$$

$$s_i = f(t_i) = (1-\lambda)(\frac{t_i - t_1}{t_n - t_1})^2 + \lambda \tag{2.14}$$

สำหรับฟังก์ชันความเป็นสมาชิกแบบฟังก์ชันของเวลานั้น สามารถแบ่งออกอีกหลายแบบ ในที่นี้ได้ทำการศึกษา 2 แบบคือ ฟังก์ชันความเป็นสมาชิกของฟัซซีแบบเชิงเส้น (Linear) ดังสมการที่ (2.13) และฟังก์ชันความเป็นสมาชิกของฟัซซีแบบกำลังสอง (Quadratic) ดังสมการที่ (2.14) เป็นต้น

อย่างไรก็ตามฟังก์ชันความเป็นสมาชิกยังสามารถเปลี่ยนแปลงแก้ไขให้เหมาะกับงานที่กำลังปฏิบัติหรือตามความต้องการได้

2.7 k-Fold Cross-validation

k-Fold Cross-validation เป็นวิธีการแบ่งข้อมูลออกเป็นกลุ่มจำนวน k กลุ่ม (k-Fold) ในตอนแรกเลือกข้อมูลกลุ่มที่ 1 เป็นข้อมูลชุดทดสอบ และข้อมูลชุดที่เหลือจะเป็นข้อมูลชุดฝึกสอน แล้วนำข้อมูลไปประมวลผล จากนั้นจะสลับข้อมูลกลุ่มที่ 2 มาเป็นชุดทดสอบและข้อมูลกลุ่มอื่นๆที่เหลือเป็นชุดฝึกสอน สลับอย่างนี้ไปเรื่อย ๆ จนครบ k กลุ่ม ในขั้นตอนสุดท้ายจะหาค่าความคลาดเคลื่อนในแต่ละกลุ่ม แล้วเลือกเอาชุดที่ให้ค่าความคลาดเคลื่อนน้อยที่สุดไปใช้ในการทดสอบระบบ วิธีการนี้ข้อมูลทุกตัวอย่างจะได้เป็นทั้งชุดทดสอบและชุดฝึกสอนในระบบอีกด้วย

2.8 การหาค่าความถูกต้อง

การหาค่าความถูกต้องในงานวิจัยนี้จะใช้การคำนวณหาค่าความคลาดเคลื่อนสัมบูรณ์ (Mean Absolute Error: MAE) โดยพิจารณ์ผลที่ได้จากสมการที่ (2.15) ยิ่งผลที่ได้มีค่าน้อย ๆ แสดงว่าความถูกต้องของระบบมีมากตามไปด้วยนั่นเอง

$$e = \left(\frac{\sum_{i=1}^{n} |D_i - y_i|}{n} \right) \tag{2.15}$$

เมื่อ n คือ จำนวนข้อมูลทั้งหมดที่เป็นอินพุต

y_i คือ ค่าของเอาท์พุตจริง

D_i คือ ค่าที่ได้จากการทำนาย