การแก้ปัญหาการจัดกลุ่มข้อมูลด้วยวิธีซัพพอร์ตเวคเตอร์แมชชีน และตัวอย่างการประยุกต์ใช้งาน

Data Classification Problem Solving using Support Vector Machine Approach and Its Application

สุพจน์ จันทร์วิพัฒน์[1]

บทคัดย่อ

บทความฉบับนี้นำเสนอหลักการของซัพพอร์ตเวคเตอร์แมชชีน ซึ่งกำลังได้รับความสนใจอย่างกว้างขวาง เพื่อใช้แก้ปัญหาการจำแนกกลุ่มข้อมูลและการจดจำรูปแบบ จากนั้นจะกล่าวถึงการประยุกต์ใช้งานซัพพอร์ตเวคเตอร์แมชชีนในการแก้ปัญหาการจัดกลุ่มข้อมูลผ่านการพัฒนาโปรแกรมจดจำลายมือเขียนพยัญชนะภาษาไทย โดยเริ่มจากการจัดเตรียมข้อมูล การฝึกสอน การสร้างแบบจำลอง และการทดสอบแบบจำลองเพื่อหาประสิทธิภาพในการจำแนกกลุ่มข้อมูล ซึ่งจากการทดลองกับกลุ่มข้อมูลพยัญชนะที่เขียนโดยบุคคลต่าง ๆ พบว่า ซัพพอร์ตเวคเตอร์แมชชีนสามารถจำแนกกลุ่มข้อมูลพยัญชนะในแต่ละกลุ่มได้อย่างถูกต้องโดยมีอัตราเฉลี่ยสูงกว่า 86 %

คำสำคัญ: ซัพพอร์ตเวคเตอร์แมชชีน การจำแนกข้อมูล การจดจำรูปแบบ การเรียนรู้ของเครื่องจักร การจำแนกพยัญชนะภาษาไทย

Abstract

This paper presents the principle of Support Vector Machine (SVM) that is currently gaining a great interest for solving problems about data classification and pattern recognition. Afterward, an application of SVMs for data categorization is proposed through a Thai hand-written consonant recognition program. This part begins with a data preparation, a training method and a data modeling approach. Finally, the obtained model will be tested for evaluating the performance of data classification. The experimental results have shown that the SVMs can achieve a high accuracy rate, more than 86%, on real dataset.

Keywords: Support Vector Machines, Data Classification, Pattern Recognition, Machine Learning, Thai Consonant Recognition

[1]อาจารย์, ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์, วิทยาลัยเทคโนโลยีอุตสาหกรรม, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

รับเมื่อ 20 เมษายน 2554 ตอบรับเมื่อ 23 พฤษภาคม 2554

1. บทนำ

การเรียนรู้ของเครื่องจักร (machine learning) [1] เป็นศาสตร์แขนงหนึ่งที่กล่าวถึงการศึกษาเกี่ยวกับอัลกอริทึมของคอมพิวเตอร์ เพื่อทำให้คอมพิวเตอร์สามารถเรียนรู้และปรับเปลี่ยนพฤติกรรมในการตอบสนองต่อข้อมูลต่าง ๆ ได้อย่างอัตโนมัติ วิธีการหนึ่งที่ได้รับความนิยมก็คือ โครงข่ายประสาทเทียม (Artificial Neural Networks: ANNs) [2] ซึ่งถูกคิดค้นขึ้นครั้งแรกในปี 1958 ต่อมาในปี 1986 David E. Rumelhart ได้นำเสนออัลกอริทึมในการสอนคอมพิวเตอร์โดยใช้โครงข่ายประสาทเทียมและมีการใช้งานอย่างแพร่หลายจนถึงปัจจุบันคือ อัลกอริทึม Backpropagation จากนั้นในปี ค.ศ. 1999 Vapnik [3] ได้นำเสนอเทคนิคการเรียนรู้บนพื้นฐานทฤษฎีทางสถิติ นั่นคือ ซัพพอร์ตเวคเตอร์แมชชีน (Support Vector Machines: SVMs) การเรียนรู้ด้วยเทคนิคของ SVMs นั้นจะเป็นการเรียนรู้โดยอาศัยกลุ่มข้อมูลตัวอย่าง (training samples) หรือที่เรียกกันว่า Supervised Learning Technique กล่าวคือ SVMs จะสร้างแบบจำลอง (model) หรือฟังก์ชั่นเพื่อใช้อธิบายกลุ่มข้อมูลตัวอย่าง จากนั้นจะนำแบบจำลองหรือฟังก์ชั่นที่ได้ไปใช้เพื่อพยากรณ์ (predicting) หรือประมาณการณ์ (approximating) กลุ่มข้อมูลตัวอย่างในอนาคตต่อไป โดย Zheng และคณะได้ศึกษาและเปรียบเทียบประสิทธิภาพของซัพพอร์ตเวคเตอร์แมชชีนและโครงข่ายประสาทเทียม ในงานที่เกี่ยวกับการจำแนกกลุ่มข้อมูลไว้ดังรายละเอียดใน [4]

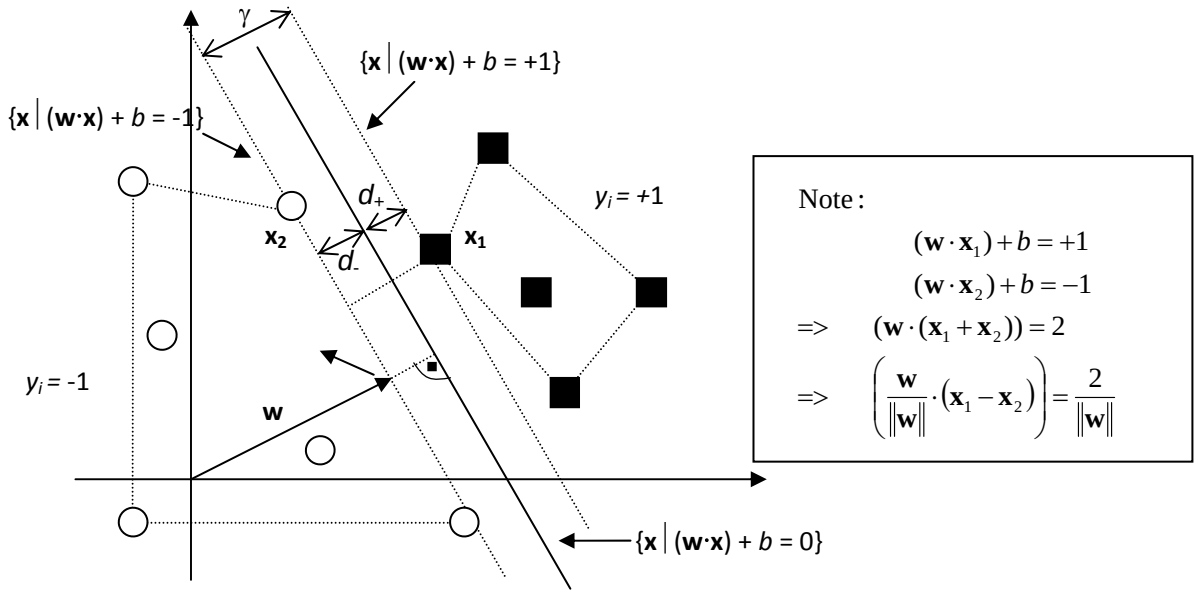
ดังนั้นในบทความนี้ จะกล่าวถึงทฤษฎีและหลักการพื้นฐานของซัพพอร์ตเวคเตอร์แมชชีนตลอดจนนำเสนอตัวอย่างและการประยุกต์ใช้งานซัพพอร์ตเวคเตอร์แมชชีนในงานที่เกี่ยวกับการจำแนกกลุ่มข้อมูลลายมือเขียนพยัญชนะภาษาไทย นอกจากนั้นจะกล่าวถึงเครื่องมือในการศึกษาและประยุกต์ใช้งาน SVMs นั่นคือ LIBSVM [5] ซึ่งเป็นไลบรารี่ของซัพพอร์ตเวคเตอร์แมชชีนที่ได้รับการยอมรับว่ามีประสิทธิภาพสูง และใช้งานง่ายตัวหนึ่งในกลุ่มของไลบรารี่ซัพพอร์ตเวคเตอร์แมชชีนที่มีอยู่ในปัจจุบัน ซึ่งรายละเอียดต่าง ๆ จะได้กล่าวถึงในส่วนต่อไป

2. ซัพพอร์ตเวคเตอร์แมชชีน (Support Vector Machines: SVMs) [3, 6]

ซัพพอร์ตเวคเตอร์แมชชีนเป็นเทคนิคหนึ่งที่ได้รับความนิยมอย่างแพร่หลายในงานที่เกี่ยวข้องกับการจดจำรูปแบบตลอดจนการแก้ปัญหาการจัดกลุ่ม (classification problem) หลักการของซัพพอร์ตเวคเตอร์แมชชีน คือ การสร้างไฮเปอร์เพลนที่เหมาะสมบนระนาบของข้อมูลตัวอย่าง (training data) เพื่อแบ่งแยกกลุ่มข้อมูลที่แตกต่างกัน ในการสร้างไฮเปอร์เพลนที่เหมาะสม เราจะนิยามระยะห่างระหว่างจุดของข้อมูลที่อยู่ใกล้กับไฮเปอร์เพลนมากที่สุดทั้งสองด้าน คือ d_+ และ d_- ระยะมาร์จิ้น (Margin) γ เกิดจากระยะ $d_+ + d_-$ ไฮเปอร์เพลนที่เหมาะสมคือไฮเปอร์เพลนที่มีค่ามาร์จิ้น γ กว้างที่สุดดังแสดงในรูปที่ 1 โดยข้อมูลตัวอย่างที่อยู่บนขอบของมาร์จิ้น γ จะถูกเรียกว่าซัพพอร์ตเวคเตอร์ (support vector)

จากรูปที่ 1 เป็นการแบ่งกลุ่มข้อมูลแบบ 2 กลุ่มกำหนดให้กลุ่มข้อมูลที่ใช้ในการฝึกสอน (training dataset) ประกอบด้วย l ตัวอย่าง (samples) ซึ่งสามารถแสดงอยู่ในรูป $\{x_k, y_k\}, k = 1,\ldots,l$ และ $x_k \in \Re^n$, $y_k \in \{-1,+1\}$ โดย x_k จะเป็นอินพุทเวคเตอร์ในขณะที่ y_k จะเป็นคลาสของข้อมูล (class label) หลักการของซัพพอร์ตเวคเตอร์แมชชีน คือ การสร้างไฮเปอร์เพลนที่เหมาะสมบนระนาบของข้อมูลตัวอย่าง ซึ่งไฮเปอร์เพลนดังกล่าวจะถูกกำหนดโดยพารามิเตอร์ $(\mathbf{w}, b)$ ดังแสดงในรูปที่ 1 โดย $\mathbf{w}$ เป็นเวคเตอร์ที่ตั้งฉากกับไฮเปอร์เพลน และ b จะเป็นค่าคงที่ซึ่งกำหนดตำแหน่งของเวคเตอร์ที่สัมพันธ์กับตำแหน่งดั้งเดิมในปริภูมิอินพุท (input space) โดยสมการของไฮเปอร์เพลนแบบเชิงเส้น จะถูกกำหนดโดยสมการ $(\mathbf{w}\cdot\mathbf{x}) + b = 0$ และเพื่อลดปัญหาในเรื่องของสเกล $\mathbf{w}$ และ b จะถูกกำหนดด้วยสมการ $|(\mathbf{w}\cdot\mathbf{x}) + b| = 1$ สำหรับจุดที่อยู่ใกล้ไฮเปอร์เพลนมากที่สุด ดังนั้นเราสามารถแสดงสมการของไฮเปอร์เพลนได้ดังสมการที่ (1)

$$y_i\left[(\mathrm{w}\cdot \mathrm{x}_i)+b\right]\geq 1 \qquad \forall i \qquad (1)$$



รูปที่ 1 การแบ่งกลุ่มข้อมูลตัวอย่าง (■ และ ○) ด้วยไฮเปอร์เพลนโดยใช้เทคนิค SVM

จากที่กล่าวมาข้างต้น จะเป็นการแบ่งกลุ่มข้อมูลด้วยไฮเปอร์เพลนแบบเชิงเส้นเท่านั้น ดังนั้นเพื่อให้อัลกอริทึมดังกล่าวสามารถแบ่งแยกกลุ่มข้อมูลที่มีลักษณะไม่เป็นเชิงเส้น (nonlinear dataset) เราจำเป็นจะต้องแปลงกลุ่มข้อมูลตัวอย่างไปสู่ปริภูมิมิติที่สูงขึ้น (higher dimensional space) ซึ่งถูกเรียกว่าปริภูมิฟีเจอร์ (feature space) โดยการแปลงดังกล่าวจะกระทำผ่านฟังก์ชั่นที่ไม่เป็นเชิงเส้น โดยสามารถแสดงสมการที่ใช้คำนวณค่าไฮเปอร์เพลนเพื่อแบ่งกลุ่มข้อมูลที่มีลักษณะไม่เป็นเชิงเส้นได้ดังสมการที่ (2) [7]

- Maximize

$$W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)$$

- Subject to (1) $\sum_{i=1}^{l}\alpha_i y_i = 0$, and

(2) $0 \le \alpha_i \le C \quad \forall i$. (2)

โดยตัวแปร $\alpha_i \ge 0$ จะถูกเรียกว่า Positive Lagrange Multipliers, $K(\mathrm{x}_i,\mathrm{x}_j)$ คือ ฟังก์ชั่นเคอร์แนล และ C จะเป็นค่าคงที่เพื่อใช้ในการปรับหรือชดเชยระหว่างค่าผิดพลาดของการฝึกสอน และความซับซ้อนของแบบจำลอง (model complexity)

จากสมการที่ (2) สามารถแสดงฟังก์ชั่นเคอร์แนลที่นิยมใช้กันโดยทั่วไปได้ดังสมการที่ (3) – (5)

โพลิโนเมียลดีกรี d (polynomial of degree d)

$$K(\mathbf{x},\mathbf{y})=(\gamma(\mathbf{x}\cdot\mathbf{y})+\beta)^d \quad (3)$$

Radial Basis Function (RBF)

$$K(\mathbf{x},\mathbf{y})=\exp(-\gamma\|\mathbf{x}-\mathbf{y}\|^2) \quad (4)$$

Sigmoid Function

$$K(\mathbf{x},\mathbf{y})=\tanh(\gamma(\mathbf{x}\cdot\mathbf{y})+\beta) \quad (5)$$

โดย γ,β และ d คือพารามิเตอร์ของเคอร์แนล (Kernel Parameters)

2.1 การประยุกต์ใช้เทคนิคซัพพอร์ตเวคเตอร์แมชชีนเพื่อการจำแนกพยัญชนะภาษาไทยจากลายมือเขียน

ในบทความนี้ได้เลือกใช้ซอฟต์แวร์ LIBSVM ซึ่งเป็นไลบรารี่ซัพพอร์ตเวคเตอร์แมชชีนที่พัฒนาโดย Chih-Jen Lin เพื่อเป็นเครื่องมือในการศึกษาการทำงานของ SVMs โดย LIBSVM จะสร้างแบบจำลอง (Model) ของข้อมูลตัวอักษรที่ใช้ฝึกสอนขึ้นมา จากนั้นผู้ใช้จะสามารถนำไฟล์แบบจำลองดังกล่าวไปใช้ในขั้นตอนการจดจำตัวอักษรต่อไป สำหรับรายละเอียดในขั้นตอนต่าง ๆ สามารถอธิบายได้ดังนี้

2.1.1 การเตรียมข้อมูลตัวอักษรเพื่อใช้ในการฝึกสอน

จากรูปที่ 2 แสดงตัวอย่างข้อมูลตัวอักษรที่ใช้ในการทดลอง โดยในการทดลองจะเก็บรูปแบบลายมือเขียนของพยัญชนะตัวละ 40 รูปแบบ รวมเป็นพยัญชนะที่ใช้ในการฝึกสอนทั้งสิ้น 1760 ตัวอักษร

รูปที่ 2 พยัญชนะที่ใช้เป็นต้นแบบในการฝึกสอน

โดยพยัญชนะแต่ละตัวจะถูกเขียนผ่านกราฟฟิกแท็บเลต (graphics tablet) จากนั้นข้อมูลของตัวอักษรทั้งหมดจะถูกนำมาลดขนาดลงเหลือ 28 X 20 พิกเซล เพื่อจะช่วยลดเวลาในการฝึกสอน โดยการลดขนาดตัวอักษรจะใช้วิธี skip pixel

2.1.2 การฝึกสอนข้อมูลลายมือเขียนพยัญชนะภาษาไทยโดยใช้ไลบรารี่ LIBSVM

ในการทดลองนี้ใช้ LIBSVM เวอร์ชั่น 2.91 ซึ่งพัฒนาขึ้นจากภาษาจาวา โดยขั้นตอนต่าง ๆ สามารถสรุปได้ดังนี้

1) การจัดรูปแบบไฟล์เพื่อใช้ฝึกสอน

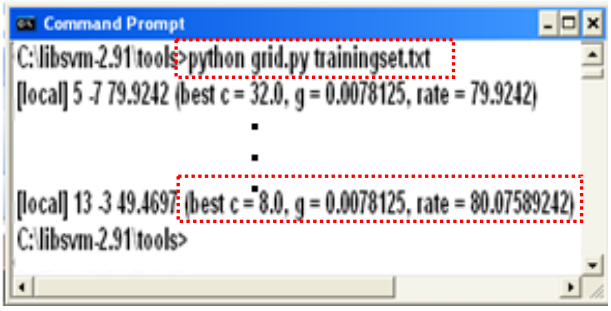
เป็นการจัดเรียงตัวอักษรในลักษณะของแถวข้อมูล [5] สามารถแสดงตัวอย่างของไฟล์ข้อมูล (trainingset.txt) ได้ดังนี้

1 1:1.0 2:1.0 3:1.0 … 559:1.0 560:1.0
1 1:1.0 2:1.0 3:1.0 … 559:1.0 560:1.0
…
44 1:1.0 2:1.0 3:1.0 … 559:1.0 560:1.0

2) การหาค่าพารามิเตอร์ C และ γ

ในการทดลองจะใช้ฟังก์ชันเคอร์แนล RBF ดังนั้นจึงมีพารามิเตอร์ 2 ตัว นั่นคือพารามิเตอร์ C และ γ (สมการที่ 2 และ 4) สามารถแสดงคำสั่งใน LIBSVM ได้ดังนี้

python grid.py trainingset.txt



รูปที่ 3 รูปแบบคำสั่งเพื่อหาค่าพารามิเตอร์ C และ γ

จากผลการทำงานของโปรแกรมในรูปที่ 3 ค่าพารามิเตอร์ C และ γ ที่ดีที่สุดคือ C = 8.0 และ γ= 0.0078125 ซึ่งจะให้ค่าความถูกต้องของข้อมูลประมาณ 80 %

3) การฝึกสอนด้วยกลุ่มข้อมูลตัวอย่าง

ในขั้นตอนต่อไปจะเป็นการฝึกสอน SVMs โดยมีรูปแบบคำสั่งที่ใช้ฝึกสอนของ LIBSVM ดังนี้

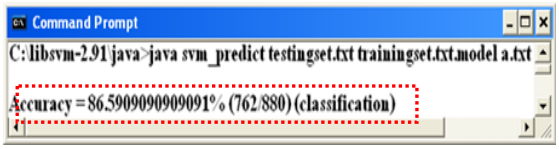
java svm_train –c 8.0 –g 0.0078125 trainingset.txt

จากนั้น โปรแกรมจะสร้างแบบจำลองของกลุ่มข้อมูลตัวอย่างในรูปของไฟล์นามสกุล .model (จากตัวอย่างจะได้ไฟล์แบบจำลองคือ trainingset.txt.model)

4) การทดสอบแบบจำลองของข้อมูล

คำสั่งที่ใช้ในการทดสอบแบบจำลองที่สร้างขึ้นสามารถแสดงได้ดังนี้

java svm_predict testingset.txt trainingset.txt.model a.txt



รูปที่ 4 คำสั่งทดสอบแบบจำลองที่สร้างขึ้น

จากรูปแบบของคำสั่ง a.txt จะเป็นชื่อไฟล์ที่ใช้สำหรับเก็บผลลัพธ์ของการทดสอบ โดยค่าความถูกต้องสามารถแสดงได้ดังรูปที่ 4 โดยผู้ใช้สามารถนำไฟล์ svm_predict นี้ไปดัดแปลงหรือใช้งานร่วมกับโปรแกรมอื่น ๆ เพื่อพัฒนาเป็นโปรแกรมประยุกต์สำหรับรู้จำตัวอักษรได้ต่อไป

3. ผลการทดลอง

ไฟล์ข้อมูลที่ใช้ในการทดสอบจะได้มาจากการเก็บข้อมูลลายมือเขียนจากกลุ่มผู้ทดสอบที่แตกต่างกัน โดยจะเก็บจำนวนตัวอักษร (พยัญชนะ) ตัวละ 20 รูปแบบ รวมจำนวนตัวพยัญชนะที่ใช้ในการทดสอบจำนวนทั้งสิ้น 880 ตัวอักษร จากรูปที่ 4 เป็นผลลัพธ์ที่ได้จากการทดสอบโดยใช้โปรแกรม svm_predict พบว่าแบบจำลองที่สร้างขึ้นสามารถจำแนกกลุ่มข้อมูลได้ถูกต้องคิดเป็นเปอร์เซ็นต์คือ 86.59 % โดยมีตัวอักษรที่จำแนกกลุ่มผิดพลาดจำนวน 118 ตัวอักษร จากจำนวนตัวอักษรทั้งหมด 880 ตัวอักษร (762/880) ซึ่งสามารถแสดงรายละเอียดการจำแนกกลุ่มพยัญชนะแต่ละกลุ่มได้ดังตารางที่ 1 จากตารางจะพบว่าพยัญชนะ “ก” “ถ” “ภ” ซึ่งมีลักษณะที่ใกล้เคียงกัน ในการทดสอบพบว่าการเขียนส่วนหัวของตัวพยัญชนะจะส่งผลต่อความถูกต้องของการจำแนกตัวอักษร ในขณะเดียวกันพยัญชนะ “ข” “บ” “ป” ส่วนของช่องว่างระหว่างส่วนหน้าและส่วนหลังของตัวอักษรในผู้เขียนแต่ละคนก็จะมีความแตกต่างกันรวมถึงส่วนหางของตัวอักษร ซึ่งปัจจัยเหล่านี้อาจจะส่งผลทำให้การจำแนกกลุ่มของพยัญชนะในกลุ่มนี้มีโอกาสที่จะผิดพลาดได้ ซึ่งแนวทางการแก้ไขอาจทำได้โดยการเพิ่มจำนวนพยัญชนะที่ใช้ในการฝึกสอนให้มากขึ้นโดยให้ครอบคลุมกรณีต่าง ๆ ที่อาจจะก่อให้เกิดความสับสนในการจำแนกตัวอักษร แต่ในขณะเดียวกันไฟล์แบบจำลองก็จะมีขนาดใหญ่ตามไปด้วย ซึ่งก็จะส่งผลให้การคำนวณต่าง ๆ ใช้เวลาในการประมวลผลเพิ่มมากขึ้น และอีกวิธีหนึ่งคือการเลือกข้อมูลที่ใช้ในการฝึกสอน หรือ features โดยการพิจารณาเฉพาะฟีเจอร์ที่มีความเด่นและมีความจำเพาะของพยัญชนะแต่ละตัว เช่น การนำข้อมูลส่วนหัวของตัวอักษรมาพิจารณา หรืออื่น ๆ แต่ก็มีส่วนที่จะต้องพิจารณาก็คือ การได้มาของฟีเจอร์ (feature extraction) จะต้องมีการคำนวณที่ไม่สลับซับซ้อนจนเกินไปและใช้เวลาในการคำนวณน้อย มิฉะนั้นก็จะส่งผลต่อประสิทธิภาพโดยรวมของการประมวลผล ซึ่งจากที่กล่าวมาข้างต้นก็จะเป็นแนวทางในการปรับปรุงประสิทธิภาพทางด้านเวลา และความถูกต้องในการจำแนกกลุ่มข้อมูล

4. บทสรุป

บทความฉบับนี้นำเสนอทฤษฎีและหลักการของซัพพอร์ตเวคเตอร์แมชชีน ในการจำแนกกลุ่มข้อมูล และการจดจำรูปแบบ นอกจากนั้นได้มีการนำเสนอตัวอย่างการประยุกต์ใช้งานซัพพอร์ตเวคเตอร์แมชชีน ในการจำแนกตัวอักษรลายมือเขียนพยัญชนะภาษาไทย โดยใช้ไลบรารี่ซัพพอร์ตเวคเตอร์แมชชีน LIBSVM เป็นเครื่องมือในการศึกษาเกี่ยวกับซัพพอร์ตเวคเตอร์แมชชีน ซึ่ง LIBSVM นับได้ว่าเป็นโปรแกรมที่มีประสิทธิภาพและใช้งานได้ง่ายโปรแกรมหนึ่ง โดยตัวโปรแกรมจะซ่อนความซับซ้อนของสมการคณิตศาสตร์ไว้และสร้างส่วนติดต่อกับผู้ใช้ที่ง่ายต่อการใช้งาน ทำให้การนำทฤษฎีซัพพอร์ตเวคเตอร์แมชชีนไปประยุกต์ใช้งานสามารถทำได้อย่างสะดวก และจากผลการทดลองได้แสดงให้เห็นว่าซัพพอร์ตเวคเตอร์แมชชีนเป็นเทคนิคหนึ่งที่มีประสิทธิภาพโดยเฉพาะในการแก้ปัญหาการจัดกลุ่มข้อมูลและการจดจำรูปแบบ ซึ่งในปัจจุบันจะพบเห็นการนำซัพพอร์ตเวคเตอร์แมชชีนไปประยุกต์ใช้งานด้าน ต่าง ๆ อย่างหลากหลาย [7, 8]

พยัญชนะทดสอบ	พยัญชนะทดสอบรูปแบบละ 20 ตัว		
	ถูกต้อง (ตัวอักษร)	ผิดพลาด (ตัวอักษร)	เปอร์เซ็นต์ความถูกต้อง
ก	16	4	80 %
ข	17	3	85 %
ฃ	18	2	90 %
ค	18	2	90 %
ฅ	17	3	85 %
ฆ	18	2	90 %
ง	19	1	95 %
จ	19	1	95 %
ฉ	18	2	90 %
ช	18	2	90 %
ซ	17	3	85 %
ฌ	17	3	85 %
ญ	18	2	90 %
ฎ	17	3	85 %
ฏ	17	3	85 %
ฐ	18	2	90 %
ฑ	18	2	90 %
ฒ	17	3	85 %
ณ	17	3	85 %
ด	16	4	80 %
ต	17	3	85 %
ถ	15	5	75 %

พยัญชนะทดสอบ	พยัญชนะทดสอบรูปแบบละ 20 ตัว		
	ถูกต้อง (ตัวอักษร)	ผิดพลาด (ตัวอักษร)	เปอร์เซ็นต์ความถูกต้อง
ท	19	1	95 %
ธ	18	2	90 %
น	17	3	85 %
บ	16	4	80 %
ป	16	4	80 %
ผ	19	1	95 %
ฝ	17	3	85 %
พ	16	4	80 %
ฟ	18	2	90 %
ภ	15	5	75 %
ม	18	2	90 %
ย	16	4	80 %
ร	18	2	90 %
ล	16	4	80 %
ว	18	2	90 %
ศ	18	2	90 %
ษ	18	2	90 %
ส	16	4	80 %
ห	17	3	85 %
ฬ	18	2	90 %
อ	18	2	90 %
ฮ	18	2	90 %

ตารางที่ 1 ผลการจำแนกกลุ่มพยัญชนะจากกลุ่มข้อมูลทดสอบจำนวน 880 ตัวอักษร

เอกสารอ้างอิง

1 T.M. Michell, “Machine Learning”, WCB/McGraw-Hill, 1997.

2 C.M. Bishop, “Neural Networks for Pattern Recognition”, Oxford University Press, 1999.

3 V. Vapnik, “An Overview of Statistical Learning Theory”, IEEE Transactions on Neural Networks, 1999, Vol. 10, no. 5, pp. 988-999.

4 E. Zheng, P. Li and Z. Song, “Performance Analysis and Comparison of Neural Networks and Support Vector Machines Classifier”, In Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, China, 2004, pp. 4232-4235.

5 C. C. Chang and C. J. Lin, “LIBSVM: a library for support vector machines”, 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

6 N. Cristianini and J Shawe-Taylor, “An Introduction to Support Vector Machines and other kernel-based learning methods”, Cambridge University Press, 2000.

7 C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, Data Mining and Knowledge Discovery, 1998, Vol. 2, no. 2, pp. 121-167.

8 C. Manop, V. Kinnares and S. Chunwiphat, “SVM-Based Stator Fault Diagnosis for Induction Motors using DQ0 Voltage Components”, International Journal of Innovative Computing, Information and Control, September 2010.