Support Vector Machines for Data Classification and Regression

Chih-Jen Lin

Department of Computer Science
National Taiwan University

Talk at Yang Ming University, March 29, 2002
Outline

- Support vector machines (SVM): an introduction
- Example: protein secondary structure prediction
- Example: engine misfire detection
Data Classification

- Given training data in different classes (labels known)
  Predict test data (labels unknown)
- Examples
  - Handwritten digits recognition
  - Speech recognition
  - Text classification
  - Face recognition
  - Web mining
  - Intrusion detection
  - Spam filtering
- Identification of defection solder joints
- Prediction of signal peptide and cleavage sites in human secretory proteins.

- Methods:
  - Nearest Neighbor
  - Neural Networks
  - Decision Tree

- Support vector machines: a new method
- Early development in Bell Labs from 1990 to 1995
- Becoming more and more popular
- The ambition: replacing neural networks on some applications
Support Vector Machines

- Training vectors: \( x_i, i = 1, \ldots, l \)
- Consider a simple case with two classes:
  Define a vector \( y \)
  
  \[
  y_i = \begin{cases} 
  1 & \text{if } x_i \text{ in class 1} \\
  -1 & \text{if } x_i \text{ in class 2,}
  \end{cases}
  \]
- A hyperplane which separates all data
A separating hyperplane: \( w^T x + b = 0 \)

\[
(w^T x_i) + b > 0 \quad \text{if } y_i = 1 \\
(w^T x_i) + b < 0 \quad \text{if } y_i = -1
\]

- Decision function \( f(x) = \text{sign}(w^T x + b) \), \( x \): test data

Variables: \( w \) and \( b \): Need to know coefficients of a plane

Many possible choices of \( w \) and \( b \)
• Select $w, b$ with the **maximal margin**.

**Maximal distance** between $w^T x + b = \pm 1$

Vapnik’s **statistical learning theory**.

$$
(w^T x_i) + b \geq 1 \quad \text{if } y_i = 1 \\
(w^T x_i) + b \leq -1 \quad \text{if } y_i = -1
$$

(1)

• Distance between $w^T x + b = 1$ and $-1$:

$$
2/\|w\| = 2/\sqrt{w^T w}.
$$

• $\max 2/\|w\| \equiv \min w^T w / 2$

$$
\min_{w,b} \quad \frac{1}{2} w^T w \\
y_i((w^T x_i) + b) \geq 1, \quad \text{from (1)} \\
i = 1, \ldots, l.
$$
Higher Dimensional Feature Spaces

- Earlier we tried to find a linear separating hyperplane
  Data may not be linear separable

- Non-separable case: allow training errors

\[
\begin{align*}
\min_{w, b, \xi} & \quad \frac{1}{2} w^T w + C \left( \sum_{i=1}^{l} \xi_i \right) \\
\text{s.t.} & \quad y_i \left( (w^T x_i) + b \right) \geq 1 - \xi_i, \\
& \quad \xi_i \geq 0, \ i = 1, \ldots, l
\end{align*}
\]

- \( \xi_i > 1, \ x_i \) not on the correct side of the separating plane
- \( C \): large penalty parameter, most \( \xi_i \) are zero
• **Nonlinear case:** linear separable in other spaces?

![Diagram of linear separability in higher-dimensional space](image)

• **Higher dimensional (maybe infinite) feature space**

\[
\phi(x) = (\phi_1(x), \phi_2(x), \ldots).
\]

• **Example:** \( x \in \mathbb{R}^3, \phi(x) \in \mathbb{R}^{10} \)

\[
\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)
\]
• A standard problem [Cortes and Vapnik, 1995]:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \left( \sum_{i=1}^{l} \xi_i \right)$$

$$y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \quad i = 1, \ldots, l$$
Comparisons with Other Methods

- Result from [Hsu and Lin, 2001]:

<table>
<thead>
<tr>
<th>Problem</th>
<th>#training</th>
<th>#testing</th>
<th>#class</th>
<th>statlog rate</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>dna</td>
<td>2000</td>
<td>1186</td>
<td>3</td>
<td>95.9%</td>
<td>95.447%</td>
</tr>
<tr>
<td>satimage</td>
<td>4435</td>
<td>2000</td>
<td>6</td>
<td>90.6%</td>
<td>91.3%</td>
</tr>
<tr>
<td>letter</td>
<td>15000</td>
<td>5000</td>
<td>26</td>
<td>93.6%</td>
<td>97.98%</td>
</tr>
<tr>
<td>shuttle</td>
<td>43500</td>
<td>14500</td>
<td>7</td>
<td>99.99%</td>
<td>99.924%</td>
</tr>
</tbody>
</table>

- Statlog: best rates by different methods

- SVM is very competitive
Example: Protein Secondary Structure Prediction

- Given amino acid sequences, for each amino acid, predict it as an α helices, a β strand, or a coil.
- Training data: sequences with known secondary structure.
- Earlier results:
  - Before 1988: less than 60%
  - 1988: 63%: Qian and Sejnowski; Holley and Karplus
  - 1993: 70%: Rost and Sander
  - 2000: Around 75%
  - Most use Neural Networks
• How is 63% achieved: the data encoding

Each training/testing data: a moving window of \( n \) (typically 13-21) neighboring residues

Each window: 21 attributes (20 amino acids and a null input)

Each window one and only one attribute: 1, others: 0

• Example:

\[
1 \ 21:1 \ 42:1 \ 63:1 \ 84:1 \ 105:1 \ 126:1 \ 147:1 \ 168:1 \ 184:1 \\
201:1 \ 226:1 \ 233:1 \ \ldots
\]

window size: 17

• Why not an integer between 1 to 21:

\[
1 \ 1:21 \ 2:21 \ 3:21 \ 4:21 \ 5:21 \ 6:21 \ 7:21 \ 8:21 \ 9:16 \ 10:12 \ \ldots
\]
• This is related to the **machine learning algorithm**

• How to select the size of moving windows?

• How is 70% achieved?

• **Evolutionary information used**

• For each residue the frequency of occurrence of each of the 20 amino acids at one position in the alignment is computed

• **Example**

  1  21:1  42:1  63:1  84:1  105:1  126:1  147:1  168:1  183:.50
  184: .50  201: .50  205: .50  226: .50  230: .50  233: .50
  244: .50  256: .50  259: .50  281: .50  290: .50  312: .50
  313: .50  316: .50  332: .50  345: .50  355: .50

• After the alignment of \( r \) similar sequences
$r/2$: 15th amino acid as the first

$r/2$: 16th amino acid as the first

- Using the same 130 sequences by Rost and Sander:

<table>
<thead>
<tr>
<th>Test sets</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td># Test</td>
<td>4276</td>
<td>3328</td>
<td>3449</td>
<td>3842</td>
<td>2871</td>
<td>2771</td>
<td>3850</td>
<td>24387</td>
</tr>
<tr>
<td># Training</td>
<td>20111</td>
<td>21059</td>
<td>20938</td>
<td>20545</td>
<td>21516</td>
<td>21616</td>
<td>20537</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>66.6%</td>
<td>73.4%</td>
<td>72.9%</td>
<td>69.7%</td>
<td>71.6%</td>
<td>71.3%</td>
<td>69.3%</td>
<td>70.5%</td>
</tr>
</tbody>
</table>

- About 75% results?
  Better techniques or because of larger sets?

- Without knowledge on biology, it is difficult to have significant progress

- Other topics we are working on:
Protein fold classification

Prediction of signal peptide and cleavage sites
Example: Engine Misfire Detection

- First problem of IJCNN Challenge 2001
- Given time series length \( T = 50,000 \)
- The \( k \)th data

\[
x_1(k), x_2(k), x_3(k), x_4(k), x_5(k), y(k)
\]

- Example:

\[
\begin{array}{cccccc}
0.000000 & -0.999991 & 0.169769 & 0.000000 & 1.000000 \\
0.000000 & -0.659538 & 0.169769 & 0.000292 & 1.000000 \\
0.000000 & -0.660738 & 0.169128 & -0.020372 & 1.000000 \\
1.000000 & -0.660307 & 0.169128 & 0.007305 & 1.000000 \\
0.000000 & -0.660159 & 0.169525 & 0.002519 & 1.000000 \\
0.000000 & -0.659091 & 0.169525 & 0.018198 & 1.000000 \\
0.000000 & -0.660532 & 0.169525 & -0.024526 & 1.000000 \\
0.000000 & -0.659798 & 0.169525 & 0.012458 & 1.000000 \\
\end{array}
\]
• \( y(k) = \pm 1 \): output, affected only by \( x_1(k), \ldots, x_4(k) \)

• \( x_5(k) = 1 \) considering \( y(k) \) for evaluating accuracy

• 50,000 training data, 100,000 testing data (in two sets)

• Past and future information may affect \( y(k) \)

• \( x_1(k) \): periodically nine 0s, one 1, nine 0s, one 1, and so on.
GAC: Encoding Schemes

- For SVM: each data is a vector
- \(x_1(k)\): periodically nine 0s, one 1, nine 0s, one 1, ...
  10 binary attributes
  \(x_1(k - 5), \ldots, x_1(k + 4)\) for the \(k\)th data
- \(x_4(k)\) more important
  Including \(x_4(k - 5), \ldots, x_4(k + 4)\) for the \(k\)th data
- Each training data: 22 attributes
Background: Engine Misfire Detection

- Engine misfire: a substantial fraction of a cylinder’s air-fuel mixture fails to ignite
- Frequent misfires: pollutants and costly replacement
- On-board detection: required after 1998
- Detection: engine crankshaft rational dynamics with a position sensor
  i.e. Crankshaft acceleration deficit after firing
- Training data: from some experimental environment
- More details: Feldkamp and Puskorius [1998]
GAC: Training SVM

- Selecting parameters; generate a good model for prediction
- Avoid overfitting
- 5-fold cross validation on 50,000 data
  Data randomly separated to five groups.
  Each time four as training and one as testing
- RBF kernel $K(x_i, x_j) = \phi(x_i)^T\phi(x_j) = e^{-\gamma \|x_i - x_j\|^2}$
- Two parameters: $\gamma$ and $C$
- Use $C = 2^4, \gamma = 2^2$ and train 50,000 data for the final model
Support Vector Machines

\begin{align*}
\text{lg}(C) & \quad \text{lg}(\gamma) \\
1 & \quad -2 \\
2 & \quad -1 \\
3 & \quad 0 \\
4 & \quad 1 \\
5 & \quad 2 \\
6 & \quad 3 \\
7 & \quad 4
\end{align*}

\text{d2}
• Training SVM: One \((C, \gamma)\), 50,000 data, 5-fold CV: < 1 hour on PIII-500 with 384 MB RAM

• Test set 1: 656 errors, Test set 2: 637 errors

• About 3000 support vectors of 50,000 training data
  A good case for SVM

• This is just the outline. There are other details.

• It is essential to do model selection.
Conclusions and Future Directions

- On some problems SVM is very competitive
- Integration with data mining software
- Applications: key to improve SVM
  Data mining is an empirical science
- All our research results can be found at
  http://www.csie.ntu.edu.tw/~cjlin