SUPPORT VECTOR MACHINE IN MATLAB

Weicheng Guo
We classify red points (in a circle with radius of 1) as 1 and green points (in a ring with radius from 1 to 2) as -1 in this figure.

We want to find the hyperplane separating these two classes. Our decision boundary will have equation:

$$w^T x + b = 0$$
Let \( \{x_1, \ldots, x_n\} \) be the training data set and let \( y_i \in \{1, -1\} \) be the class label of \( x_i \).

The decision boundary should classify all points correctly with

\[
y_i(w^T x_i + b) \geq 1 - \xi_i
\]

The decision boundary can be found by solving the following constrained optimization problem

\[
\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \\
subject to y_i(w^T x_i + b) \geq 1 - \xi_i
\]
We can introduce Lagrange multipliers to represent the condition and thus have the following formulation:

\[
\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]

subject to \(0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0\)
Since it is a nonlinear classification in this case, we can use kernel functions that operates on the lower dimension vectors $x_i$ and $x_j$ to produce a value equivalent to the dot product of the higher-dimensional vectors. Therefore, the data can be linearly separated in that space and the formulation is transformed to:

$$
\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)
$$

subject to $0 \leq \alpha_i \leq C$, $\sum_{i=1}^{n} \alpha_i y_i = 0$
Kernel Functions

- Linear kernel
  \[ K(x_i, x_j) = x_i^T x_j \]

- Polynomial kernel with \( \gamma \), \( d \) and \( r \)
  \[ K(x_i, x_j) = (\gamma x_i^T x_j + r)^d \]

- Radial basis function kernel with \( \gamma \)
  \[ K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2) \]

- Sigmoid kernel with \( \gamma \) and \( r \)
  \[ K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \]
LIBSVM\textsuperscript{[1]} is an open source machine learning library developed at the National Taiwan University and written in C++ though with a C API. LIBSVM implements the SMO algorithm for kernelized support vector machines (SVMs), supporting classification and regression.

<table>
<thead>
<tr>
<th>Differences between LIBSVM and SVM function in Matlab</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBSVM</td>
</tr>
<tr>
<td>Classification or regression</td>
</tr>
<tr>
<td>Classification type</td>
</tr>
<tr>
<td>SVM formulations</td>
</tr>
</tbody>
</table>

\textsuperscript{[1]} https://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html
LIBSVM -- A LIBRARY FOR SUPPORT VECTOR PROCEDURE

1. Transform data to the format of an SVM package

2. Conduct simple scaling on the data

3. Consider the RBF kernel \( K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \)

4. Find the best parameter \( C \) and \( \gamma \)

5. Use the best parameter \( C \) and \( \gamma \) to train the whole training data

6. Test
Number of Support Vectors is 106, Ratio of all samples 53% (106/200)

---Accuracy---

---Overall Accuracy of Classification--- 93.5% (187/200)

Accuracy of Classification $1 = 100\%$ (100/100)

Accuracy of Classification $-1 = 87\%$ (87/100)
Number of Support Vectors is 54, Ratio of all samples 27% (54/200)

---Accuracy---

---Overall Accuracy of Classification--- 100% (200/200)
Accuracy of Classification 1 = 100% (100/100)
Accuracy of Classification -1 = 100% (100/100)
Number of Support Vectors is 167, Ratio of all samples 83.5% (167/200)

Tips: ratio is too large (>=80%), refine the parameters

--- Accuracy ---

--- Overall Accuracy of Classification --- 100% (200/200)
Accuracy of Classification 1 = 100% (100/100)
Accuracy of Classification -1 = 100% (100/100)
TESTING RESULT

**GAMMA = 3**

result =

1  -1
2  -1
3  -1
4  -1
5  -1
6   1
7   1
8  -1
9   1
10 -1

The visualization of classification

- + class 1
- o class -1
- * Test Points
- □ Support Vectors
TESTING RESULT

GAMMA = 3

result =

1  -1
2  -1
3   1
4   1
5   1
6   1
7   1
8  -1
9  -1
10 -1

The visualization of classification

 PURDUE UNIVERSITY
CONCLUSION

- SVM is a new method of machine learning based on statistics theory. In contrast to ‘black box’ learning approaches (artificial neural network), SVM is supported by certain mathematical models.

- The training of SVM is relatively easy. Unlike in neural network, SVM can get global optimum and the training time does not depend on dimensionality of feature space any more by using the kernel trick.

- The key to training SVM is to select a kernel function and its parameters, which cannot be conducted by a principled manner.