A Brief Review of Linear Support Vector Machine for Machine Learning Programming

Yong Suk Suh[*], Seung Ki Shin, Dane Baang, Sang Mun Seo, Jong Bok Lee
Kijang Research Reactor Design and Construction Project Div., Korea Atomic Energy Research Institute (KAERI), 111, Daedeok-Daero 989Beon-Gil, Yuseong-Gu, Daejeon, 34057, Korea
[*]*Corresponding author: yssuh@kaeri.re.kr*

1. Introduction

For the data classification in supervised machine learning programming, logistic regression can be used. Logistic regression was briefly reviewed in previous papers [1], [2] and [3]. Logistic regression is based on the logistic (i.e., sigmoid) function which is derived from the logit function and classifies input data into one of two states with a threshold that is given by the programmer. Logistic regression represents the data classification in the form of probabilistic scores between 0 and 1. When we do not want the probabilistic scores, support vector machine (SVM) can be used for the data classification [4]. Thus, SVM is an alternative method for the data classification in supervised machine learning programming.

When we want to apply the machine learning programming to the nuclear facilities, the historical data classification is sometimes required to evaluate new input data such as operator actions and field signals pressure and temperature and so on, into one of two states such as correct or incorrect, true or false, etc. For this, SVM can be used.

There are two types of SVMs: linear SVM and non-linear SVM. This paper briefly reviews linear SVM (LSVM) only. The LSVM is called simple or hard SVM. The non-linear SVM is called complex or soft SVM.

This paper describes theoretical background of the LSVM and provides simple test cases to show how the LSVM works. This paper also briefly mentions the shortage of the LSVM and points for improving the LSVM.

2. Linear Support Vector Machine

The SVM is used to find an optimized hyperplane which separates the given data into one of two areas (or planes) as shown in Fig. 1.

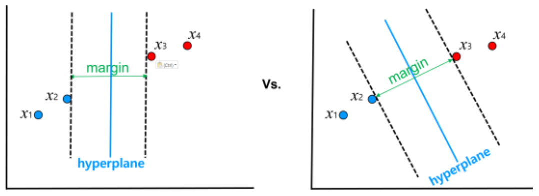

Figure 1 Two hyperplanes separating data

In Fig. 1, the hyperplane on the right side is better than that on the left side because its margin is larger. Since the hyperplane is one dimension, it is called linear support vector machine (LSVM).

In Fig. 2, the hyperplane is $\mathbf{x} \bullet \mathbf{w} + b = 0$ and we need to find the optimal $\mathbf{w}$ and b to get a maximal $\mathbf{m}$, where $x \in R^2$. When we represent the x data in the form of vectors, the $\mathbf{x}_3$ and $\mathbf{x}_2$ are called support vectors that determine the decision boundary for the classification. The margin $\mathbf{m}$ is represented in the form of

$$\begin{aligned} \mathbf{m} &= (\mathbf{x}_3 - \mathbf{x}_2) \bullet \frac{\mathbf{w}}{\|\mathbf{w}\|} \\ &= (\mathbf{x}_3 \bullet \mathbf{w} - \mathbf{x}_2 \bullet \mathbf{w}) \frac{1}{\|\mathbf{w}\|} \\ &= (1- b + 1 + b) \frac{1}{\|\mathbf{w}\|} \\ &= \frac{2}{\|\mathbf{w}\|} \qquad (1) \end{aligned}$$

The maximal of $\mathbf{m}$ is represented in the form of

$$\begin{aligned} \text{Max}(\mathbf{m}) &= \text{Max}\left(\frac{2}{\|\mathbf{w}\|}\right) \qquad (2.1) \\ &= \text{Min}\left(\frac{\|\mathbf{w}\|}{2}\right) \qquad (2.2) \\ &= \text{Min}\left(\frac{\|\mathbf{w}\|^2}{2}\right) \qquad (2.3) \\ &= \text{Min}\left(\frac{\mathbf{w} \bullet \mathbf{w}}{2}\right) \qquad (2.4) \end{aligned}$$

Although we change the maximal Eq. (2.1) to minimal Eq. (2.2), the goal of getting the maximal of $\mathbf{m}$ is not changed. We can square the Eq. (2.2) to eliminate the square root of the L2-norm in the Eq. (2.2). Although we square the Eq. (2.2), the goal of getting the maximal of $\mathbf{m}$ is not changed. The Eq. (2.2) and Eq. (2.3) are said to be the same problem. Finally, we get the Eq. (2.4) which is a quadratic optimization problem.

In order to solve the Eq. (2.4) using Lagrangian, we need to set constraints. When we set a group of data to 1 (positive plane) and the other group of data to -1 (negative plane) as shown in Fig. 2, all the data can be represented in the form of

$$y_i(\mathbf{x} \bullet \mathbf{w}+b)\text{-}1 \geq 0, \qquad (3)$$

where $y_i = 1 \quad when\ \mathbf{x} \bullet \mathbf{w}+b \geq 1$ and
$y_i = -1 \quad when\ \mathbf{x} \bullet \mathbf{w}+b \leq \text{-}1$

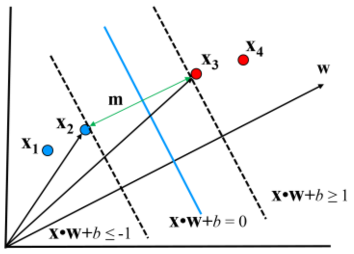
x_1
x_2
x_3
x_4
m
w
$\mathbf{x} \bullet \mathbf{w}+b \geq 1$
$\mathbf{x} \bullet \mathbf{w}+b = 0$
$\mathbf{x} \bullet \mathbf{w}+b \leq \text{-}1$

Figure 2 A hyperplane with constraints

Then Eq. (3) becomes an inequality constraint of the Eq. (2.4) to be solved using Lagrangian. We need a

Lagrangian multiplier λ that is equal to or greater than zero. Thus, we can define Lagrangian

$$\mathcal{L}(\mathbf{w},b,\lambda) = f(\mathbf{w}) - \lambda g(\mathbf{w}) = \frac{\mathbf{w}\bullet\mathbf{w}}{2} - \sum \lambda_i\,(y_i(\mathbf{x}\bullet\mathbf{w}+b)\text{-}1), \quad (4)$$

where $f(\mathbf{w})=\text{Min}\left(\frac{\mathbf{w}\bullet\mathbf{w}}{2}\right)$

subject to $\lambda_i \geq 0$ and

$g(\mathbf{w}) = y_i(\mathbf{x}\bullet\mathbf{w}+b)\text{-}1 \geq 0$, i=1..n

In order to solve the Eq. (4), we take partial derivatives

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = \mathbf{w} - \sum \lambda_i\, y_i \mathbf{x} \quad (5)$$

$$\frac{\partial\mathcal{L}}{\partial\lambda} = y_i(\mathbf{x}\bullet\mathbf{w}+b)\text{-}1 \quad (6)$$

Thus, we can see that **w** is a function of λ

$$\mathbf{w}=\sum \lambda_i\, y_i \mathbf{x} \quad (7)$$

Because of the inequality constraints in the Eq. (4), the KKT (Karush-Kuhn-Tucker) conditions are taken into account to determine λ. Based on the characteristics of the KKT conditions, λ as an Lagrange multiplier is positive for a support vector that lies on the hyperplane and is zero for a vector that lies on the positive or negative plane.

3. Simple Test Cases

We don't need to numerically program the Eq. (4) because the Sckit-learn provides an SVM library as follows:

Sklearn.svm.SVC(kernel="linear").fit(**X**, y)

A simple test case is shown in Table 1 to use the SVM library.

Table 1: Test case 1

x_1	10	20	37	33	**40**	**50**	50	65	90	100
x_2	15	25	21	22	**30**	**60**	80	57	80	100
y	-1	-1	-1	-1	**-1**	**1**	1	1	1	1

When we input the data in Table 1 to the SVM library, we can obtain the output as shown in Fig. 3.

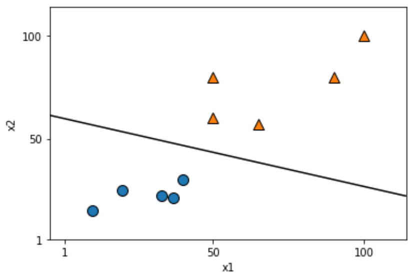

Figure 3 The result of test case 1

We can see that the SVM library classifies the given **X** data well. When we move x(65, 57) as shown in Table 2 to the negative plane, we can see that the margin shrinks as shown in Fig. 4.

Table 2: Test case 2

x_1	10	20	37	33	**40**	**50**	50	65	90	100
x_2	15	25	21	22	**30**	**60**	80	57	80	100
y	-1	-1	-1	-1	**-1**	**1**	1	**-1**	1	1

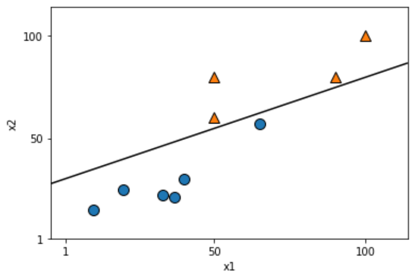

Figure 4 The result of test case 2

When we move x(100,100) as shown in Table 3 to the negative plane, we can see that the SVM library cannot classify the data as shown in Fig. 5 because x(100,100) is too far from the negative plane. This is a shortage of the LSVM. Thus, we need a non-linear SVM to overcome this shortage.

Table 3: Test case 3

x_1	10	20	37	33	**40**	**50**	50	65	90	100
x_2	15	25	21	22	**30**	**60**	80	57	80	100
y	-1	-1	-1	-1	**-1**	**1**	1	1	1	**-1**

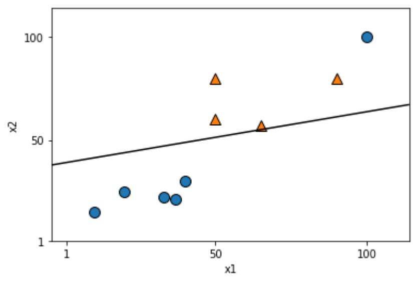

Figure 5 The result of test case 3

The machine learning programming was performed using Spyder 3.3.6 in Anaconda 3 1.9.12, Python 3.7.4, and Sckit-learn library.

4. Conclusions

This paper briefly reviews linear support vector machine (LSVM) and shows how it works with simple test cases. The LSVM is not realistic in the real world. In order to overcome a shortage of LSVM, non-linear SVM with complementary slackness conditions and a kernel trick method will be studied further.

Acknowledgement

This work was supported by Korea MSIT (Grant Code: 2020M2D5A1078133)

REFERENCES

[1] Yong Suk Suh, et al., A Brief Review of a Machine Learning Programming of Simple Logistic Regression, Transactions of the Korean Nuclear Society Autumn Meeting, Yeosu, Korea, October 25-26, 2018.
[2] Yong Suk Suh, et al., Considerations on Machine Learning Programming of Multiple Linear Regression, Transactions of the Korean Nuclear Society Spring Meeting, Jeju, Korea, May 23-24, 201.
[3] Yong Suk Suh, et al., A Performance Evaluation Method of a Machine Learning Programming of Simple Logistic Regression, Transactions of the Korean Nuclear Society Autumn Meeting, Goyang, Korea, October 24-25, 2019.
[4]http://en.m.wikipedia.org/wiki/Support_vector_machine