Transactions of the Korean Nuclear Society Virtual Autumn Meeting
October 21-22, 2021

A Brief Review of Non-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

A support vector machine (SVM) is used as a method to classify data into two or more categories in machine learning programming. A linear support vector machine (LSVM), as supervised machine learning programming, was briefly reviewed in authors' previous paper [1]. The LSVM is based on a linear kernel method, which is not realistic because of non-linear characteristics of input data that needs to be trained in the real world. Thus, we need to consider adopting a non-linear support vector machine (NLSVM) to overcome the limitations of the LSVM.

For applying the support vector machine to nuclear facilities, it is expected to classify field signal data such as pressure and temperature and so on into two or more categories. There might be the operator's actions to be classified into some categories. We sometimes want to classify historically stored data. In this case, we can implement the classification using machine learning techniques such as logistic regression and support vector machine. When the data is linearly separable, the LSVM can be used. However, in nuclear facilities, most data that needs to be classified will be non-linear. In this case, the NLSVM can be used.

This paper revisits the review of LSVM published in [1] and describes the limitations of the LSVM. We introduce a NLSVM using a polynomial kernel method and compare the results of the LSVM and NLSVM. The same sample test data in the previous paper [1] is also used in this paper for the comparison.

2. Non-linear Support Vector Machine

We briefly reviewed the linear support vector machine (LSVM) in our previous paper [1]. The LSVM finds an optimal hyperplane that maximizes the margin **m**, which maximally separates the given data into one of two categories (i.e., groups, areas or planes) as shown in Fig. 1.

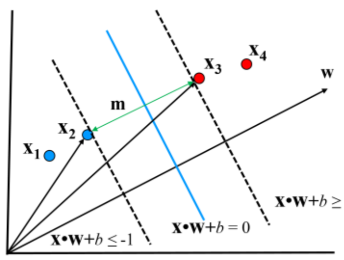

Figure 1 A hyperplane in the LSVM
(This figure is taken from [1])

The $\mathbf{x}_2$ and $\mathbf{x}_3$ in Fig. 1 are called support vectors that determine the decision boundary for the classification, which is also a boundary of the hyperplane **x•w**+b. The hyperplane is defined in [2]. The margin **m** in Fig. 1 is represented in the form of $\frac{2}{\|\mathbf{w}\|}$. In order to find the optimal **w**, we need to solve a quadratic optimization problem because the $\|\mathbf{w}\|$ represents the L2-norm of the hyperplane [1].

There exist well-known kernels for the NLSVM that are described in [3] as follows:

- Polynomial (homogeneous): $k(\vec{x_i}, \vec{x_j}) = (\vec{x_i} \cdot \vec{x_j})^d$.
- Polynomial (inhomogeneous): $k(\vec{x_i}, \vec{x_j}) = (\vec{x_i} \cdot \vec{x_j} + 1)^d$.
- Gaussian radial basis function: $k(\vec{x_i}, \vec{x_j}) = \exp(-\gamma\|\vec{x_i} - \vec{x_j}\|^2)$ for $\gamma > 0$. Sometimes parametrized using $\gamma = 1/(2\sigma^2)$.
- Hyperbolic tangent: $k(\vec{x_i}, \vec{x_j}) = \tanh(\kappa\vec{x_i} \cdot \vec{x_j} + c)$ for some (not every) $\kappa > 0$ and $c < 0$.

Among the NLSVM described above, we consider only the polynomial (homogeneous) kernel in this paper because the other kernels are so sensitive to control.

We used Sckit-learn libraries for programming the LSVM and NLSVM as shown below:

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

For the equally well classified data (i.e., x_1 and x_2 in Table 1) as two groups for example group 1 and -1 as shown in Table 1, the LSVM classifies the data as represented by the solid line in Fig. 1. The NLSVM is represented by the dotted line in Fig. 1. The LSVM has better classification because the hyperplane margin of the LSVM is bigger than that of the NLSVM.

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

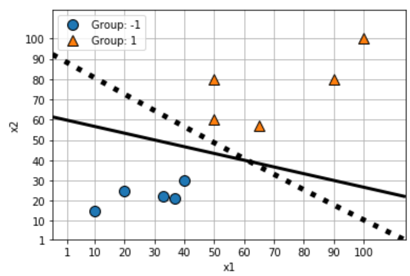

Figure 2 The comparison of LSVM and NLSVM for equally well classified data

When one data item $\boldsymbol{x}$ (65, 57) is changed (i.e., moved) from group 1 to -1 as shown in Table 2, the margins of the LSVM and NLSVM decrease as shown in Fig. 3. The NLSVM non-linearly separates the plane.

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

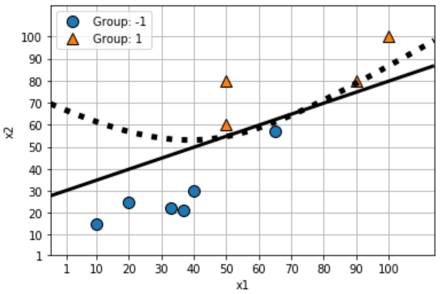

Figure 3 The comparison of the LSVM and NLSVM for short marginal classified data

When the most extreme outlier data item $\boldsymbol{x}$ (100,100) is changed (i.e., moved) from group 1 to -1 as shown in Table 3, the LSVM cannot classify the data. This is obviously a natural phenomenon because the data cannot linearly be classified as it is. However, the NLSVM can classify them. Thus, the NLSVM can overcome the limitations of the LSVM.

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**

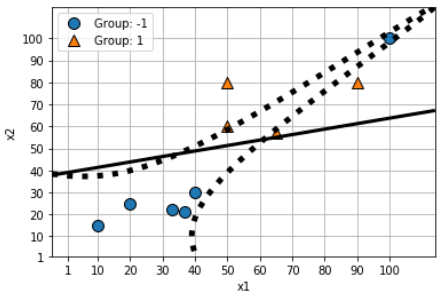

Figure 4 The comparison of the LSVM and NLSVM for not equally well classified data

Even when the data is mixed as shown in Table 4, the NLSVM can classify them as shown in Fig. 5. It classifies the most extreme outlier data that the LSVM cannot.

Table 4: Test case 4

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**

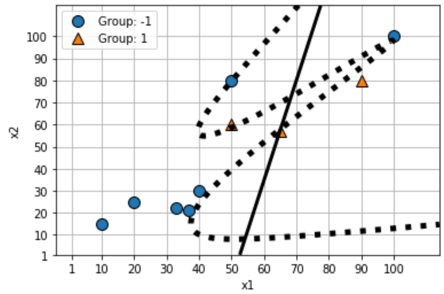

Figure 5 The comparison of the LSVM and NLSVM for mixed data

We know that the NLSVM can classify the given data where the LSVM cannot. However, the NLSVM can be overfitting the classification. The problem in using the support vector machine is that it takes too much computing time because it must perform quadratic programming for the LSVM. It takes at least O(n^2) computing time where n is the number of features to be classified. We performed the machine learning programming using Spyder 3.3.6 in Anaconda 3 1.9.12, Python 3.7.4, and Sckit-learn library.

3. Conclusions

In this paper, we revisited the linear support vector machine (LSVM) from [1] and briefly reviewed the non-linear support vector machine (NLSVM). We compared the results of the LSVM and NLSVM to show that the NLSVM overcomes the limitations of the LSVM. The kernel method adopted in the support vector machine and 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 Linear Support Vector Machine for Machine Learning Programming, Transactions of the Korean Nuclear Society Virtual Spring Meeting, May, 13-14, 2021.
[2]http://en.m.wikipedia.org/wiki/Hyperplane
[3]http://en.m.wikipedia.org/wiki/Support_vector_machine