Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 19-20, 2022

A Brief Review of a Kernel Method in Machine Learning Programming

Yong Suk Suh*, Seung Ki Shin
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

We presented brief reviews of logistic regression and support vector machine methods as data classification methods in previous papers [1], [2], and [3]. Data classification methods are used in many applications in supervised machine learning programming for nuclear facilities. For field process signals such as pressure and temperature signals, when we want to classify the signals with user defined appropriate thresholds into two states, we can apply logistic regression and support vector machine methods for the classification.

Logistic regression has a limitation in classifying non-linear data and can classify data into only one of two states with a predefined threshold. The support vector machine (SVM) [4] is an alternative method for data classification in supervised machine learning programming. There are two types of SVMs: the linear SVM and the non-linear SVM. The linear SVM is called a simple or hard SVM. The non-linear SVM is called a complex or soft SVM. The SVM is based on the kernel method.

This paper briefly reviews and describes the kernel method with simple test data.

2. Kernel Method

The kernel method for the SVM is different from the definition in linear algebra. The kernel in linear algebra (i.e., linear transformation) is simply defined as a vector space in a domain that is mapped to the zero (i.e., null) vector space in an image or co-domain. The kernel method for the SVM is defined as follows [5]:

Definition 2.8 [Kernel function] A *kernel* is a function κ that for all $\mathbf{x}, \mathbf{z} \in X$ satisfies

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle ,$$

where ϕ is a mapping from X to an (inner product) feature space F

$$\phi : \mathbf{x} \longmapsto \phi(\mathbf{x}) \in F.$$

The kernel is used to transform (or map) data in a vector space into another space (or plane) as shown in Fig. 1 [5].

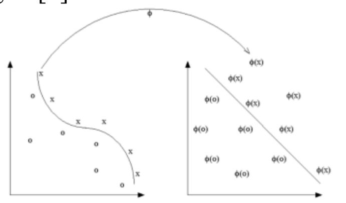

Fig. 2.1. The function ϕ embeds the data into a feature space where the nonlinear pattern now appears linear. The kernel computes inner products in the feature space directly from the inputs.

Figure 1 Data transformation by a kernel

The kernel method can be used to linearly classify data by transforming the data into another vector space. It usually transforms the data into a higher vector space.

3. Simple Test Case

We describe data classification based on the kernel method by using arbitral test data as shown in Table 1. $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are data to be transformed and classified. $\boldsymbol{y}$ is a predefined class where the data is classified to. For example, a class 0 contains the data that $\boldsymbol{y}$ is 0 and a class 1 contains the data that $\boldsymbol{y}$ is 1.

Table 1: Arbitral test data before being transformed

$\boldsymbol{x}_1$	-10	-9	-8	-4	-3	3	4	8	9	10
$\boldsymbol{x}_2$	-10	-8	-7	-1	-2	2	0	9	8	10
$\boldsymbol{y}$	0	0	0	1	1	1	1	0	0	0

The data in Table 1 is plotted as shown in Fig. 2. The plot in Fig. 2 is generated using the matplotlib library.

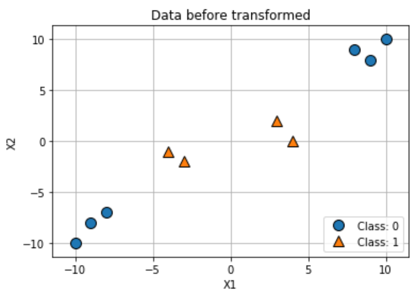

Figure 2 Plotted data in Table 1

The data in Table 1 cannot be linearly classified as shown in Fig. 3. It is generated using the SVM Sckit-learn library.

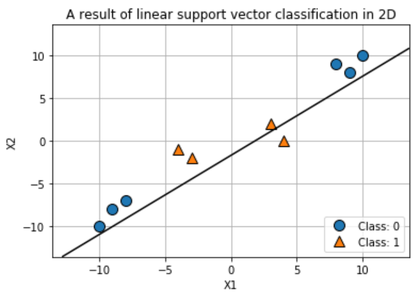

Figure 3 Linear classification of data in Table 1

We define a kernel function that transforms the data in Table 1 into a three dimensional data as follows:

$$\phi: \mathbf{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2) \longmapsto \phi(\mathbf{x}) = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_2^2) \in F = R^3 \quad (1)$$

In Eq.(1), the third element $\boldsymbol{x}_3$ is generated by squaring the second element $\boldsymbol{x}_2$. The data in Table 1 is transformed into the data in Table 2.

Table 2: Arbitral test data after being transformed

$\boldsymbol{x}_1$	-10	-9	-8	-4	-3	3	4	8	9	10
$\boldsymbol{x}_2$	-10	-8	-7	-1	-2	2	0	9	8	10
$\boldsymbol{x}_3$	100	64	49	1	4	4	0	81	64	100
$\boldsymbol{y}$	0	0	0	1	1	1	1	0	0	0

The data in Table 2 is plotted in three dimensional space as shown in Fig. 4. The plot in Fig. 4 is generated using the mpl_toolkits.mplot3d library.

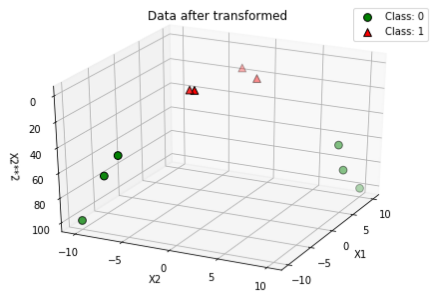

Figure 4 Plotted data in Table 2

After the transformation of data, we can linearly classify the data as shown in Fig. 5. It is generated using the SVM Sckit-learn library.

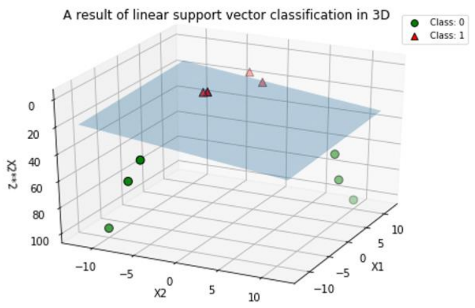

Figure 5 Linear classification of data in Table 2

We can see that the data in Table 2 is linearly well classified after the transformation. When we draw contour lines as shown in Fig. 6, the data is well classified.

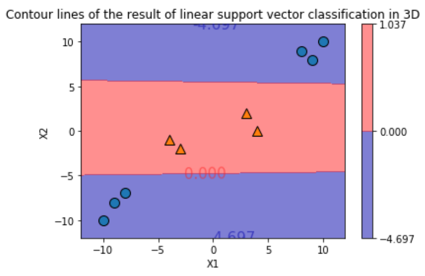

Figure 6 Contour lines of Fig. 5

The machine learning programming was performed using Spyder 3.3.6 in Anaconda 3 1.9.12, Python 3.7.4, and matplotlib and Sckit-learn libraries. The data classification is programmed using the support vector machine function in Sckit-learn library.

4. Conclusions

This paper briefly reviewed a kernel method used for data classification in supervised machine learning programming. This paper showed that non-linear data can be linearly classified after transforming them into higher vector space using the kernel function.

Additional study is needed to find the optimal kernel function depending on the data.

Acknowledgement

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

REFERENCES

[1] 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.
[2] 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.
[3] Yong Suk Suh, et al., A Brief Review of Non-linear Support Vector Machine for Machine Learning Programming, Transactions of the Korean Nuclear Society Virtual Autumn Meeting, October, 21-22, 2021.
[4]http://en.m.wikipedia.org/wiki/Support_vector_machine
[5] John Shawe-Taylor, Nello Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.