Evaluation of Proper Hyperparameters in Machine Learning Algorithms for Fuel Cycle Related Documents Classification

Byoungchan Han*, Byeonghyeok Ha, Tongkyu Park
[a]*FNC Technology Co., Ltd., 32F, 13 Heungdeok 1-ro, Giheung-gu, Yongin-si, Gyeonggi-do, Korea*
**Corresponding author: bchan007@fnctech.com*

1. Introduction

In 1997, IAEA adopted an Additional Protocol to the former safeguard agreement. This protocol contains the contents of NPT (Treaty on the Non-Proliferation of Nuclear Weapons) members to report nuclear and nuclear related activities to IAEA for preventing nuclear proliferation. For effective compliance on the agreement, a three-year research project was held by the Korea Institute of Nonproliferation and Control (KINAC) to develop collection and analysis system of nuclear fuel cycle related R&D projects and activities in 2018. [1]

In this paper, we introduce the optimization through sensitivity analysis of supervised learning algorithms applied to collection and analysis system.

2. Applied Machine Learning Algorithms

The collection and analysis system contains three machine learning algorithms; Naïve Bayesian, SVM (Support Vector Machine) and XGBoost (Extreme Gradient Boosting) to analyze the contents of collected documents. This section introduces the mechanism and hyperparameters of each algorithm for optimization via sensitivity analysis.

2.1 Naïve Bayesian

Naïve Bayesian is one of the classic and simple machine learning algorithms used in classification [2]. It determines the category of data by calculating probability distribution over a set of classes with independence assumptions across the features. Because of its simplicity, Naïve Bayesian can classify data easier and faster, especially in multi-group classification.

Consider the problem classifying documents to various categories. According to Bayes' Theorem, the conditional probability that document d will exist in category c_k can be expressed as Eq. 1.

$$p(c_k \mid d) = \frac{p(c_k)p(d \mid c_k)}{p(d)} \qquad (1)$$

As classification is solely interested in the fraction of probabilities, the denominator $p(d)$ is out of our concern. Assume that document d is consisted of words $w_1 \sim w_n$. According to the independence assumptions, $p(c_k \mid d)$ can be expressed as Eq. 2.

$$p(c_k \mid d) \propto p(c_k)p(w_1|c_k)p(w_2|c_k) \dots p(w_n|c_k) \qquad (2)$$

In other word, Naïve Bayesian algorithm classifies data by comparing the number of appearances of words in each category. Also, Eq. 2 informs that external parameter does not exist that we can handle, which means optimization is unnecessary.

2.2 Support Vector Machine

SVM generates a hyperplane in the n-dimensional vector space to group data points. SVM aims to maximize the distance from the hyperplane to the nearest data point [3]. Such hyperplane is expressed as Eq. 3, where $\vec{w}$ is the normal vector to the hyperplane.

$$\vec{w} \cdot \vec{x} + b = y \qquad (3)$$

Assume that dataset is given as $(\vec{x_1}, y_1), \dots, (\vec{x_n}, y_n)$. $\{y_k\}$ are either 1 or -1, which represents the category where data belong. Then, we can find two more hyperplanes that are parallel to the existing hyperplane that contain the very data points that are closest to the original hyperplane. These planes are called support vectors, and the distance between them is called margin. Geometrically, margin can be expressed as $\frac{2}{\|\vec{w}\|}$.

However, finding boundary hyperplane is not always possible. Therefore, SVM algorithm modifies constraint that some points can cross the boundary, or defines a curved boundary. The former method is called C-SVM, or soft margin SVM. Let ξ_i as the distance between margin plane and data, ξ_i acts as a penalty to the margin. New objective function to minimize is given in Eq. 4.

$$Obj = min\left(\frac{1}{2}\|w\|^2\right) + C\sum_i \xi_i \qquad (4)$$

C is the external parameter that can be handled. As the impact of ξ_i on objective function gets more influential when C increases, C tends to restrain data from crossing the margin surface, and therefore the distance between two margin surfaces decreases. On the other hand, smaller C weakens the influence of ξ_i and in consequence, margin increases.

The latter method is called kernel-SVM. Even C-SVM is often impossible to generate the hyperplane to classify data (e.g. one group of data surrounding the other), a method to extend dimension was developed. However, the amount of calculation is extremely increased when mapping data vectors and creating boundary surface. Therefore, it uses a shortcut approach called "kernel trick", which calculates the inner product

of two mapping function $\phi(\vec{x_i})$ and $\phi(\vec{x_j})$ for each pair of data point $\vec{x_i}$ and $\vec{x_j}$. The result of this inner product is called "kernel", be capable of replace mapping function when computing boundary surface.

RBF (Radial Basis Function) kernel-SVM, one of the most effective and popular methods in SVM, uses Gaussian RBF kernel expressed in Eq. 5.

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \qquad (5)$$

Eq. 5 indicates the influence of Euclidean distance between two data points on the kernel increases with the hyperparameter γ. It means that each datum gets more weighted when calculating the boundary surface, leads curvature of the hyperplane to increase.

2.3 Extreme Gradient Boosting

XGBoost is an ensemble algorithm combining several decision trees [4]. Each decision tree has nodes to classify features of data, so that points in the vector space can be divided into smaller groups. At the tree's deepest level, nodes are paired with categories which finally classify data to their fitted groups.

There are several hyperparameters to handle for finding a proper XGBoost model. First of all, learning rate η refers to the shrinkage of gradient of targeted learning values made in every time step. If learning rate is low, internal parameters would be changed slowly, and therefore computing time to reach optimum value increases. Besides, this phenomenon makes XGBoost model fits more accurately on the current training data set. However, low learning rate can also leads to data overfitting and falling into local minima.

Second, XGBoost limits the minimum number of observations each node can have. If a certain node has less data than the limit, XGBoost does not divide such node further. Therefore, model performance can be improved as the fluctuation caused by singular value is ignored.

Third, XGBoost can adjust the ratio of training data and their features to be learned in every step. Through this adjustment, it can create trees that are slightly different with each other, prevents data overfitting in outcome.

3. Analysis and Results

In our previous study [5][6], we presented the performance of machine learning algorithms in classifying documents about nuclear engineering, and nuclear fuel cycle. In this section, we will feature about the result of sensitivity analysis and tuned hyperparameters.

For the experiment, we set ten groups of documents to evaluate the performance. Detailed description of each case is presented in Table I.

Table I: Experiment Case Description

Case	Fuel Cycle Docs.	Non-Fuel Cycle Docs. (Nuc.)	Non-Nuclear Docs.	Total
I	300	0	300	600
II	300	300	0	600
III	300	150	150	600
IV	900	0	900	1,800
V	900	900	0	1,800
VI	900	450	450	1,800
VII	1,500	0	1,500	3,000
VIII	1,500	1,500	0	3,000
IX	1,500	750	750	3,000

For cross-validation, a five-fold method was used to divide the data set into five and use four of them as training group, and the other as test group. Therefore, five combinations of training and test groups were generated in such method. Moreover, five independent tests were performed on the test cases to obtain the mean and standard deviation of the F1 scores.

3.1 Naïve Bayesian

As mentioned in the previous section, Naïve Bayesian method is unnecessary to manage as there is no hyperparameter for tuning. Therefore, we presented solely the performance of the algorithm in Table II to compare with other algorithms.

Table II: Performance of Naïve Bayesian Classifier

Case	F1 Score			
	Test Set		Training Set	
	Mean	SD	Mean	SD
I	0.989	0.011	0.997	0.001
II	0.915	0.031	0.944	0.006
III	0.895	0.024	0.965	0.002
IV	0.982	0.006	0.993	0.001
V	0.952	0.006	0.978	0.002
VI	0.960	0.011	0.980	0.002
VII	0.978	0.004	0.988	0.001
VIII	0.779	0.013	0.837	0.004
IX	0.838	0.009	0.865	0.002

First of all, result proves that Naïve Bayesian classifier is very effective in separating fuel cycle documents from non-nuclear documents as F1 scores of test set in Case I, IV, VII are above 0.97. Secondly, experiment cases II, V, VIII were intended to separate fuel cycle documents from nuclear documents. As fuel cycle documents are more similar to nuclear documents than non-nuclear documents, F1 scores of test set in cases II, V, VIII were found to be lower than that in

cases I, IV and VII. F1 scores of case III, VI, IX showed insignificant difference with case II, V and VIII.

3.2 Support Vector Machine

There are two hyperparameters that needs to be adjusted in RBF SVM: C and γ. F1 score evaluation was done by changing C in the range of 10^{-3} to 10^{5} and γ in the range of 10^{-3} to 10^{3}. For each experiment case, we presented the first and second best-scored combination of C and γ, in Table III.

Table III: Performance and Hyperparameters of SVM

Case	F1 Score		Hyperparameter	
	Mean	SD	C	γ
I	0.993	0.003	**5**	**0.1**
	0.993	0.003	100	0.01
II	0.901	0.041	**5**	**0.1**
	0.893	0.034	10	0.1
III	0.956	0.006	**10**	**0.1**
	0.955	0.012	5	0.1
IV	0.995	0.003	**10**	**0.1**
	0.995	0.003	100	0.01
V	0.959	0.005	**5**	**1**
	0.959	0.005	10	1
VI	0.971	0.006	**100**	**0.01**
	0.970	0.006	10	0.1
VII	0.997	0.002	**10**	**0.1**
	0.997	0.002	100	0.01
VIII	0.783	0.011	**10**	**0.1**
	0.782	0.016	10	1
IX	0.863	0.015	**10**	**0.1**
	0.863	0.015	10	0.1

According to the result, RBF SVM shows sufficient performance in classifying nuclear fuel cycle documents in C values between 5 and 100, with γ in the range of 0.01 to 0.1.

3.3 Extreme Gradient Boosting

This part shows the trend of performances of XGBoost differ with parameters. Learning rate, minimum child weight, and tree depth were tuned to get the performance of XGBoost. Hyperparameters that showed the first and second best F1 score of XGBoost model are shown in Table IV. For the comparison, the worst XGBoost models are also shown in Table V.

Table IV: Best Performance and Hyperparameters of XGB

Case	F1 Score	Hyperparameter		
		Learning rate	min child weight	depth
I	0.997	**0.2**	**1**	**4**
	0.997	0.2	1	6
II	0.956	**0.05**	**0.75**	**4**
	0.954	0.1	0.5	8
III	0.997	**0.05**	**0.5**	**4**
	0.997	0.05	0.75	4
IV	0.998	**0.1**	**1**	**4**
	0.998	0.1	1	6
V	0.956	**0. 05**	**0.75**	**4**
	0.954	0.1	0.5	8
VI	0.967	**0.2**	**1**	**4**
	0.967	0.2	1	8
VII	0.999	**0.1**	**1**	**4**
	0.998	0.1	0.5	4
VIII	0.818	**0.1**	**0.5**	**8**
	0.815	0.1	1	8
IX	0.922	**0.2**	**1**	**8**
	0.922	0.2	0.5	6

Table V: Worst Performance and Hyperparameters of XGB

Case	F1 Score	Hyperparameter		
		Learning rate	min child weight	depth
I	0.993	0.1	0.75	6
II	0.925	0.05	0.5	8
III	0.993	0.1	0.75	4
IV	0.994	0.05	0.5	4
V	0.925	0.05	0.5	8
VI	0.953	0.05	1	8
VII	0.992	0.05	1	4
VIII	0.788	0.2	1	8
IX	0.907	0.2	1	8

Based on the results presented in Table III and IV, XGBoost showed superior performance than RBF SVM in every test case. However, tendency of hyperparameters to optimize the XGBoost model was hard to find. Fortunately, XGBoost showed difference between the best f1 score and the worst of less than 0.03 in all cases. It implies that XGBoost is still effective without adjusting hyperparameters.

3. Conclusions

This study presented the hyperparmeters to optimize machine learning algorithms for classifying documents related to nuclear fuel cycles. According to the result, the performance of SVM varied largely depending on hyperparameters. Depending on the result, combination of in C values between 5 and 100, and γ in the range of 0.01 to 0.1 is recommended for document classification.

On the other hand, the performance of XGBoost showed insignificant differences with the hyperparameters. Learning rate in the range of 0.05 to 0.2, minimum child weight around 1, and tree depth in the range of 4 to 8 may show sufficient result in document classification.

Acknowledgement

This work was supported by the Nuclear Safety Research Program through the Korean Foundation Of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of Korea (No. 1803021).

REFERENCES

[1] Sung-ho Yoon and Dong-hoon Shin, "A Conceptual Design of the Information Analysis System for Searching Nuclear Fuel Cycle Related R&D Project", Proc. of the KRS 2018 Autumn Conference, 16(2), October 31 – November 2, 2018, Jeju, Korea.
[2] Hand, David J. and Keming Yu, "Idiot's Bayes-not so stupid after all?", International Statistical Review, 69(3), pp.385-398, 2001.
[3] C. Cortes and V. Vapnik, "Support Vector Networks", Machine Learning 20, p.273, 1995.
[4] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System", Proc. of the 22nd Conference on Knowledge Discovery and Data Mining, p.785, 2016, San Francisco.
[5] Tongkyu Park, Yunpil Jeong, Byoungchan Han, Sang Jun Lee, Chan Seo Lee, and Dong-hoon Shin, "Two Classification Algorithms for Nuclear Fuel Cycle Related Documents", Proc. of the KRS 2019 Spring Conference, 17(1), May 8 – May 10, 2019, Busan, Korea.
[6] Byoungchan Han, Kibeom Park, Yunpil Jeong, Tongkyu Park, "Classification of Nuclear Fuel Cycle Related Documents by Supervised and Unsupervised Learning Algorithms", Transactions of the Korean Nuclear Society Autumn Meeting, October 24 – October 25, 2019, Goyang, Korea.