Journal of information and
communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 17(1): 41-48, Mar. 2019 Regular paper

An Improved Text Classification Method for Sentiment Classification

Guangxing Wang[1] **and Seong Yoon Shin**[2*], ***Member*, *KIICE***

[1]Department of Information Technology Center, Jiujiang University, Jiujiang 332005, China
[2]School of Computer Information & Communication Engineering, Kunsan National University, Gunsan 54150, Korea

Abstract

In recent years, sentiment analysis research has become popular. The research results of sentiment analysis have achieved remarkable results in practical applications, such as in Amazon's book recommendation system and the North American movie box office evaluation system. Analyzing big data based on user preferences and evaluations and recommending hot-selling books and hot-rated movies to users in a targeted manner greatly improve book sales and attendance rate in movies [1, 2]. However, traditional machine learning-based sentiment analysis methods such as the Classification and Regression Tree (CART), Support Vector Machine (SVM), and k-nearest neighbor classification (kNN) had performed poorly in accuracy. In this paper, an improved kNN classification method is proposed. Through the improved method and normalizing of data, the purpose of improving accuracy is achieved. Subsequently, the three classification algorithms and the improved algorithm were compared based on experimental data. Experiments show that the improved method performs best in the kNN classification method, with an accuracy rate of 11.5% and a precision rate of 20.3%.

Index Terms: Sentiment Analysis, Machine Learning, Text Classification, k-Nearest Neighbor Method

I. INTRODUCTION

Sentiment analysis or opinion mining refers to the analysis and research of people's opinions, emotions, and evaluations on attitudes such as goods, services, and even organizations, and extracts valuable data from these big data [3], such as people's preference for the product, people's evaluation of the service, etc., for suppliers to improve products and service quality. The development and rapid start of the field benefited from social media platforms on the Web, such as product reviews, forum discussions, Weibo, and the rapid development of WeChat, because this is the first time where such a huge digital form record was registered [4]. In recent years, the machine learning-based sentiment classification method has achieved certain results, such as Amazon's book recommendation system, North American movie box office evaluation system, analysis of big data based on user preferences and evaluation, and targeted recommendation to users for hot sales. Books and hot review movies have greatly increased book sales and movie box office attendance [1, 2].

According to the granularity of text, sentiment analysis can be divided into three levels: chapter level, sentence level, and word level [5, 6]. The chapter-level sentiment analysis is a pointer to analyze the content of an article, pointing out its overall emotional direction or polarity (positive or negative). The chapter-level emotional classification is a binary classification task, which can also be a regression task. Sentence-level sentiment analysis is generally divided into knowledge-based analysis methods, network-based analysis methods, and corpus-based analysis methods [7]. Word-

Received 06 January 2019, **Revised** 13 March 2019, **Accepted** 13 March 2019
***Corresponding Author** Seong-Yoon Shin (E-mail:s3397220@kunsan.ac.kr, Tel: +82-63-469-4860, Fax. +82-63-469-4560)
School of Computer Information & Communication Engineering, Kunsan National University, Kunsan 54150, Korea.

Open Access https://doi.org/10.6109/jicce.2019.17.1.41 print ISSN: 2234-8255 online ISSN: 2234-8883

level sentiment analysis refers to the analysis of the emotions of words. The emotions of words are the main constituent elements of sentences and chapter emotions, and also the basis of sentence-level and chapter-level sentiment analyses. Word-level sentiment analysis mainly includes dictionary-based analysis methods, network-based analysis methods, and corpus-based analysis methods. A corpus-based analysis method uses machine learning-related techniques to classify the emotions of words. Machine learning methods usually require the classification model to learn the rules in the training data, and then use the trained model to predict the test data.

The sentiment analysis algorithms based on machine learning methods mainly include a decision tree algorithm (DT), support vector machine (SVM), and k-nearest neighbor algorithm (kNN). However, the traditional machine learning-based sentiment analysis method still has a certain gap in the accuracy of text classification, such as the classification regression tree algorithm (CART), SVM, kNN classification method, etc. In this paper, an improved kNN method is proposed. This method can effectively improve the accuracy of text classification. Compared with the CART, SVM, and kNN algorithms, the optimized method is the best in classification accuracy and precision.

This paper makes the following contributions:

1. The accuracy of three text classification algorithms were compared (CART, SVM, and kNN)

2. An improved kNN method was proposed. Experiments were carried out based on the sampling data set of THUCNews [8], while the excellent performance of the improved kNN method in improving the classification accuracy was verified.

II. SENTIMENT CLASSIFICATION METHODS

In this section, the methods, algorithmic processes, and advantages and disadvantages of several commonly used classification algorithms in sentiment analysis methods will be briefly described. The model of the CART and SVM lays the foundation for the next step of the kNN method.

A. CART Model

The CART algorithm is an implementation form of the decision tree. Usually there are three main implementations of decision trees, namely ID3 algorithm, CART algorithm, and C4.5 algorithm [9, 10]. The CART algorithm is a binary recursive segmentation technique. It divides the current sample into two sub-samples, so that each non-leaf node generated has two branches. Therefore, the decision tree generated by the CART algorithm is a simple binary tree. Since the CART algorithm constitutes a binary tree, there can only be "yes" or "no" in the decision of each step. Even if a feature has multiple values, the data are divided into two parts. There are two main steps in the CART algorithm. The first step is to recursively divide the sample into a tree building process. The second step is to prune with the verification data. The following is a brief introduction to the principle of CART.

Let x_1, x_2, …, x_n represent the attributes of a single sample, indicating the category they belong to. The CART algorithm divides the space of a dimension into non-overlapping rectangles in a recursive manner. The division steps are as follows:

1. An independent variable, x_i, is selected, and then the value v_i of x_i is selected, with v_i dividing the n-dimensional space into two parts, all of the points satisfying $x_i \leq v_i$ and all points of the other parts satisfying $x_i > v_i$ for non-continuous variables. In other words, the value of the attribute has only two values; that is, equal to or not equal to the value.

2. Recursive processing is performed. The two parts obtained above are re-selected according to step 1 and continue to be divided until the entire n-dimensional space is divided.

Criteria for partitioning: For a variable attribute, its dividing point is the midpoint of a pair of continuous variable attribute values. Assuming that a set of m samples has m consecutive values, then there will be m-1 split points, each split point being the mean of two consecutive values. The division of each attribute is sorted according to the amount of impurities that can be reduced, and the amount of reduction of impurities is defined as the sum of the impurity before division minus the ratio of the impurity division of each node after division. The *Gini* indicator is commonly used for impurity measurement methods. Assuming that a sample has a G class, the *Gini* impurity of a node can be defined as the formula, as shown in (1).

$$Gini(A) = 1 - \textstyle\sum_{i=1}^{C} p_i^2. \tag{1}$$

Here, p_i represents the probability of belonging to class i. When $Gini(A) = 0$, all samples belong to the same class. When all classes appear with equal probability in the node, $Gini(A)$ is maximized, with the result being $C(C$-1)/2.

According to the theoretical basis above, the actual recursive partitioning process is as follows. If all samples of the current node do not belong to the same class or only one sample remains, then this node is a non-leaf node. Therefore, it will try each attribute of the sample and the split point corresponding to each attribute, trying to find the largest division of the impurity variable. The subtree of the attribute division is the optimal branch. It can be seen that the accuracy of the CART method depends on each branch node. However, when the binary tree is established, the branch is trimmed, so that the branch data abnormality occurs during the entire classification process, resulting in low CART classification accuracy.

B. SVM Model

SVM is a common method of discrimination. In the field of machine learning, it is a supervised learning model that is commonly used for pattern recognition, classification, and regression analysis. Vapnik et al. proposed another design best criterion for linear classifiers based on years of research on statistical learning theory [11]. The principle is also linear from a point of view, and then extended to the case of linear indivisibility. Even extended to use nonlinear functions, this classifier is called Support Vector Machine (SVM). The main idea of SVM can be summarized in two points:

1. Linear case analysis. For linear indivisible cases, high-dimensional features are obtained by transforming linearly indivisible samples of low-dimensional input space into high-dimensional feature spaces using nonlinear mapping algorithms. It is possible to linearly analyze the nonlinear characteristics of samples using a linear algorithm.
2. It builds an optimal hyperplane in the feature space based on the structural risk minimization theory, so that the learner is globally optimized, and the expectation of the entire sample space satisfies a certain upper bound with a certain probability.

The SVM method maps the sample space into a feature space of high-dimensional or even infinite dimension (Hilbert space) through a nonlinear mapping p, so that the problem of nonlinear separability in the original sample space is transformed into the feature space. Meanwhile, a linearly separable problem simply explained is the ascending dimension and linearization. Ascending dimension is to map the sample to high-dimensional space. In general, this will increase the computational complexity, and even cause “dimensionality disaster,” so it is rarely used. However, as a classification, and with regression and other issues, it is very likely that the sample set that cannot be linearly processed in the low-dimensional sample space can be linearly divided (or regression) by a linear hyperplane in the high-dimensional feature space. In general, the ascending dimension will bring about the complexity of the calculation. The SVM method subtly solves this problem: applying the expansion theorem of the kernel function, you do not need to know the explicit expression of the nonlinear mapping. Because the linear learning machine is built in the high-dimensional feature space, compared with the linear model, it not only increases the computational complexity, but also avoids the “dimensional disaster” to some extent. However, the SVM method can only be used for binary classification. For multidimensional classification, the SVM method does not perform well.

III. PROPOSED IMPROVED kNN METHOD

In this section, the kNN method is briefly introduced and the improved method of the kNN model is assessed in detail.

A. kNN Model

k-Nearst Neighbor algorithm was proposed by Cover and Hart in 1968. The nearest neighbor of K is the meaning of k nearest neighbors, saying that each sample can be represented by its nearest k neighbors. kNN classification algorithm is a theoretically mature method and one of the simplest machine learning algorithms. The kNN algorithm flow is as shown in (2).

Input: Training Dataset

$$T = (x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N) \;.. \tag{2}$$

Here, $x_i \in X \subseteq R^n$ is the instance feature vector, $y_i \in Y = \{c_1, c_2, \ldots, c_k\}$ is the category of the instance, $i = 1, 2, \ldots$, and N. x is the instance feature vector.

Output: class y, which is the class to which instance x belongs. It works as follows:

1. According to the distance metric of the given point, the k points nearest to x in the training set T is found, covering the field of k points, denoted as $N_k(x)$.
2. The category y of x is determined in $N_k(x)$ according to the classification decision rule (such as majority vote).

y formula is as shown in (3).

$$y = \arg max_{c_j} \textstyle\sum_{x_i \in N_{k(x)}} I(y_i = c_j),\ i = 1, 2, \ldots, N. \tag{3}$$

In the formula above, I is an indication function; that is, when $y_i = c_j$, I is 1; otherwise, I is 0. The special case of kNN is the case of $k = 1$, which is called the nearest neighbor algorithm. For the input instance point (feature vector) x, the nearest neighbor algorithm will classify the training data set with the x nearest neighbor as the class of x. In the kNN algorithm, there are three commonly used distances, namely Manhattan distance, Euclidean distance, and Minkowski distance.

Let the feature space X be an n-dimensional real vector space R^n, x_i, $x_j \in X$, $x_i = (x_i(1), x_i(2), \ldots, x_i(n))^T$, $x_j = (x_j(1)$, and $x_j(2), \ldots, x_j(n))^T$, with the L_p distance between x_i and x_j defined as in (4):

$$L_p(x_i, x_j) = (\textstyle\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p)^{1/p}, p \geq 1. \tag{4}$$

When $p = 1$, it is called Manhattan distance. The formula is as shown in (5).

$$L_1(x_i, x_j) = \textstyle\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|. \tag{5}$$

When $p = 2$, it is called Euclidean distance, and the formula is as shown in (6).

Table 1. kNN algorithm process pseudo code description

```
INPUT: Dataset(TR, TE) //TR:Train Dataset, TE:Test Dataset
OUTPUT: Classification Report (accuracy, precision, etc.)
Begin:
    For(0<i≤len(TE, TR_i)) // Calculate the distance between the test data and each training datum
        $L_i=Lp(x_i,y_i)$;
        i++;
    return $L_i$
    Sort($L_i$) // Sort by increasing distance
    $K=L_{imix}(L_i)$ // Select the K points with the smallest distance
    $Frequent_K=K$ // Get the frequency of occurrence of the category of the first K points
    $Classification_p= Frequent_K$
    Return $Classification_p$ // Returns the category
End
```

$$L_2(x_i, x_j) = (\sum_{l=1}^{n}|(x_i^{(l)} - x_j^{(l)}|^2)^{1/2}. \quad (6)$$

When $p = \infty$, it is the maximum value of each coordinate distance, and the calculation formula is as shown in (7).

$$L_\infty(x_i, x_j) = max_l|(x_i^{(l)} - x_j^{(l)}|. \quad (7)$$

The summary of the idea of the kNN algorithm is that when the data and tags in the training set are known, the test data are input, and the features of the test data are compared with the features corresponding to the training set. With the top *k* data most similar to the training set found, the category corresponding to the test data is the one with the most occurrences among the *k* data. The description of the kNN algorithm is shown in Table 1.

B. Inadequacies of the kNN

The kNN algorithm is one of the simplest and most efficient classification algorithms in classification algorithms. It has the advantages of simplicity, ease of understanding, and exclusion to estimate parameters. In particular, kNN is suitable for the classification of rare events, and the kNN algorithm performs much better than the SVM algorithm even for multi-classification problems (objects with multiple category labels). However, the kNN algorithm also has some drawbacks.

First, when the sample is unbalanced and a new sample is input, the sample of the large-capacity class among the *K* neighbors of the sample is dominant if the sample size of one class is large and the sample size of other classes is very small. The algorithm only calculates the "nearest" neighbor samples. The number of samples in a class can be extremely large, or the samples either not close to the target sample or exceptionally close to the target sample.

Second, the amount of calculation is large. Because each of the texts to be classified must calculate its distance to all known samples, it can find its *K* nearest neighbors, which consumes a large amount of memory on the computer. When the data set is large, the calculation time increases. When the sample size is unbalanced, especially for the data set of sentiment analysis, the kNN algorithm may produce problems such as low prediction accuracy.

C. An Improved Method for kNN

Research and improvement methods for the kNN method have continued, such as the cluster-based CLKNN improvement algorithm proposed by Lijuan et al. [12], and the weight-based kNN improvement algorithm proposed by Halil Yigit et al. [13]. Due to the use of kNN algorithm for classification, it is necessary to calculate the similarity between the test text and each training text, which undoubtedly greatly increases the calculation amount of the classification, with the classification speed not being improved. Therefore, in the case of more training texts, how to reduce the amount of calculation and improve the classification accuracy are key issues. Therefore, the data are processed again during the process of using the kNN algorithm. The normalized processing method for data aims to prevent the numerical value of a certain dimension from affecting the distance calculation. There are two normalization methods: Linear Function Normalization (Min-Max scaling) and Z-score standardization. For the kNN algorithm, the Z-score standardization method to process the preprocessed data is used, and the raw data mean μ and standard deviation σ are given to standardize the data. The processed data conform to the standard normal distribution with a mean of 0 and a standard deviation of 1. The conversion function is shown in (8).

$$X^* = (X - \mu)/\sigma. \quad (8)$$

In this equation, μ is the mean of all sample data and σ is the standard deviation of all sample data. The variance normalization anti-interference ability is strong, and it is related to all data. To find the standard deviation requires the intervention of all values. If there is an outlier, it will be suppressed. For the kNN algorithm, Z-score standardization performs better than the PCA technology when distances are used to measure similar properties. In the paper, the pre-processed data set is *X*. After calculating the sample mean μ and variance σ in the data set, normalizing by 0-means to obtain a new sample space is done, and then it is processed using the kNN method. The improved kNN algorithm flow pseudo code is shown in Table 2.

IV. EXPERIMENTS

Our experiment is divided into two steps. First, the data

Table 2. Improved kNN algorithm flow pseudo code description

```
INPUT: Dataset(TR, TE) //TR:Train Dataset, TE:Test Dataset
OUTPUT: Classification Report (accuracy, precision, etc.)
Get to $\mu(TR,TE)$, $\sigma(TR,TE)$
$Dataset^* = (Dataset - \mu)/\sigma$
Begin:
    For($0<i\leq len(TE^*, TR^*_i)$) // Recalculate the distance between the test data and each training data
        $L_i = L_p^* (x_i, y_i)$;
        $i$++;
    return $L_i$
    Sort($L_i$) // Sort by increasing distance
    $K = L_{imix}(L_i)$ // Select the K points with the smallest distance
    $Frequent_K = K$ // Get the frequency of occurrence of the category of the first K points
    $Classification_p = Frequent_K$
    Return $Classification_p$ // Returns the category
End
```

set is prepared and then preprocessed. The pre-processed data sets are imported into the CART, SVM, and kNN models for prediction. The accuracy and precision rate of the three models in the classification of sentiment analysis texts are then compared. In the second step, the improved method is used for prediction and comparison.

A. Datasets

In the paper, the dataset of THUCNews is used [8]. The THUCNews dataset is generated according to the historical data filtering of the Sina News RSS subscription channel from 2005 to 2011. It contains 74 million news documents (2.19 GB), all in UTF-8 text format. Because the data set is large, only 8 classified texts were selected as experimental data sets in order to speed up the operation and the experimental processes. For the sampled data set, the word segmentation was used for each text segmentation. Each word was treated as a feature, the binary word string was used to construct more features, the stop words were removed, and the features with too many occurrences were removed. 19,630 features were received. 1998 samples were selected for training and 509 were used for testing. Finally, the method based on the bag of words model converted each text into a vector, and the training and test sets were converted to matrices and saved in npy format. The test sample classification was shown in Table 3.

B. Comparison of Classification Effects of Models

The initial data were imported into the model for training, comparing the classification effects of several models. On parameter selection, CART and SVM all use default parameters. The K value selected in kNN is 20.

Table 3. The classification of test sample

Classification No.	Classification Name	Number of Samples
1	Sports News	41
2	Entertainment News	38
3	Domestic News	62
4	Real Estate News	55
5	Education News	68
6	Fashion News	158
7	Politics News	27
8	Game News	60
Total		509

Table 4. Comparison of the three models of classification prediction

Model Name	Evaluating Indicators				
	Classification No.	Precision	Recall	F1-score	Support
CART	1	0.70	0.63	0.67	41
	2	0.59	0.63	0.61	38
	3	0.72	0.50	0.59	62
	4	0.62	0.71	0.66	55
	5	0.91	0.90	0.90	68
	6	0.69	0.75	0.72	158
	7	0.60	0.67	0.63	27
	8	0.67	0.65	0.66	60
	Avlg/Total	0.70	0.63	0.67	509
	Accuracy: **0.997997997997998** Running Time Total: 8.3s				
SVM	1	0.93	0.32	0.47	41
	2	0.00	0.00	0.00	38
	3	0.00	0.00	0.00	62
	4	1.00	0.13	0.23	55
	5	1.00	0.43	0.60	68
	6	0.35	1.00	0.52	158
	7	1.00	0.30	0.46	27
	8	0.00	0.00	0.00	60
	Avlg/Total	0.48	0.42	0.33	509
	Accuracy: **0.4194194194194194** Running Time Total: 192.5s				
kNN	1	1.00	0.37	0.54	41
	2	0.43	0.24	0.31	38
	3	0.45	0.29	0.35	62
	4	0.82	0.33	0.47	55
	5	1.00	0.60	0.75	68
	6	0.46	0.85	0.60	158
	7	1.00	0.41	0.58	27
	8	0.43	0.48	0.45	60
	Avlg/Total	0.64	0.54	0.53	509
	Accuracy: **0.6051051051051051** Running Time Total: 229.1s				

The comparison of classification accuracy before and after kNN method improvement is shown in Fig. 1.

Table 5. Comparison of classification prediction after improvement

Model Name	Evaluating Indicator				
	Classification No.	Precision	Recall	F1-score	Support
CART	1	0.81	0.63	0.71	41
	2	0.41	0.45	0.43	38
	3	0.58	0.53	0.55	62
	4	0.49	0.62	0.62	55
	5	0.89	0.48	0.84	68
	6	0.70	0.69	0.69	158
	7	0.61	0.63	0.63	27
	8	0.58	0.60	0.60	60
	Avlg/Total	0.66	0.65	0.65	509
	Accuracy: **0.997997997997998** Running Time Total: 8.7s				
SVM	1	0.00	0.00	0.00	41
	2	0.00	0.00	0.00	38
	3	0.00	0.00	0.00	62
	4	0.00	0.00	0.00	55
	5	0.00	0.00	0.00	68
	6	0.31	1.00	0.47	158
	7	0.00	0.00	0.00	27
	8	0.00	0.00	0.00	60
	Avlg/Total	0.10	0.31	0.15	509
	Accuracy: **0.3008008008008008** Running Time Total: 182.4s				
kNN	1	0.93	0.63	0.75	41
	2	0.94	0.39	0.56	38
	3	1.00	0.11	0.20	62
	4	0.27	0.80	0.40	55
	5	0.98	0.94	0.96	68
	6	0.61	0.71	0.65	158
	7	0.87	0.74	0.80	27
	8	0.95	0.30	0.46	60
	Avlg/Total	0.77	0.60	0.60	509
	Accuracy: **0.6746746746746747** Running Time Total: 145.4s				

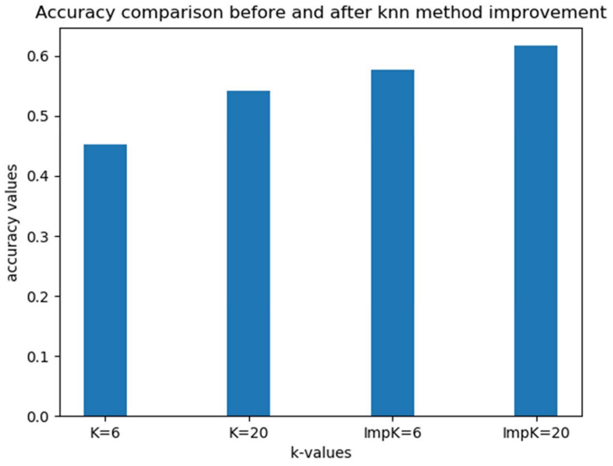

Fig. 1. Comparison of classification accuracy before and after kNN method improvement. *K* represents the value before the improvement, and ImpK represents the value of K of the improved KNN method.

C. Experimental Result and Analysis

The experimental results are shown in Table 4, Table 5, and Fig. 1. The initial data running under the three models and showing the accuracy of each model is shown in Table 4. For each category, the precision, recall, and f1-score values are displayed. The performance of the three models after normalizing the data is shown in Table 5. It is a visual comparison of the accuracy of the different k values before and after the improvement for the kNN model in Fig. 1. From the experimental results, one can draw the following conclusions:

1. When using the initial data, the CART model performed best in the classification prediction, and the SVM model performed the worst. After improvement, the kNN model performed best and the SVM model performed the worst. Among the overall evaluation indicators, the improved kNN model performed best.

2. It can be seen intuitively from Figure 1 that the accuracy of the improved kNN method classification is significantly improved.

Analysis:

1. Before the improvement, the CART model has the best accuracy and precision, but the accuracy of the second category is relatively low, indicating that the entertainment news text is shorter and the sample size is relatively small. The SVM model predicts a value of 0 for some classifications, indicating that the SVM model is not suitable for small sample classification.

2. Compared to the pre-improvement classification, the accuracy of the CART model has not changed, but the average accuracy has decreased. The accuracy of the SVM model has declined. The accuracy and precision of the kNN model have been improved, with an accuracy rate of 11.5% and a precision of 20.3%, which is the best.

3. In terms of classification time, in general, the classification time is shorter than before, especially with the improved kNN method prediction time reduced by 36.5%.

V. REFLEXION AND DISCUSSION

Research on machine learning for sentiment classification has been ongoing. The earliest paper on sentiment analysis was Po Pang's 2002 article on using SVM, ME, and Naive Bayes to calculate sentiment orientation [14]. Turney et al. [15] extended the positive and negative sentiment words using the method of point mutual information. The polar semantic algorithm was used in the analysis of text emotions, with the general corpus data being 74%. The kNN method is one of the simplest methods in the Natural Language Processing (NLP) classification method. It is simple, easy to implement, and easy to understand. It is suitable for classifying rare time and is especially suitable for multi-classification problems, which is better than SVM.

However, there are some problems in the kNN method. On the one hand, when the sample is unbalanced, the problem of low classification accuracy may occur. When the sample size of one class is exceedingly large, while the sample size of other classes is quite small, a large sample of the large-capacity class among the K neighbors of the sample may result when a new sample is input. On the other hand, the kNN method is computationally intensive. For each sample to be classified, it is necessary to calculate the distance from all the known samples and then rank K the nearest neighbors, which will increase the prediction time. At present, the commonly used improvement method is editing the known sample points in advance to remove the samples that have little effect on the classification. This improved algorithm is more suitable for a large class domain, while a smaller domain is prone to misclassification. Shweta et al. proposed an improved method of MFZ-KNN [16], and an improved kNN based on feature weight proposed by Jie Huang et al. [17]. As a future work, a better method to improve the kNN algorithm may be researched, so that the accuracy of the classification will be enhanced again.

VI. CONCLUSIONS

This paper presented an improved sentiment classification method based on kNN, and elaborated on the improved methods, algorithms, and implementation process. Based on the extracted small datasets compared with the CART and SVM sentiment classification methods, the improved kNN method performed well in the experiment. In particular, the accuracy of the improved kNN classification had improved and the running time had been greatly reduced. Experiment results show that the accuracy and precision of the kNN model rose with 11.5% and 20.3% in the sentiment classification. In particular, the improved kNN method prediction time has been reduced by 36.5%. These results prove that the proposed method not only achieved high accuracy but also performed best in the classification effect in these models.

REFERENCES

[1] B. Smith and G. Linden, “wo decades of recommender systems at amazon.com,” *IEEE Internet Computing*, vol. 21, no. 3, pp.12-18, 2017. DOI:10.1109/MIC.2017.72.

[2] S. Halder, Md. Samiullah, A. M. Jehad Sarkar, and Y.-K. Lee, “Movie swarm: Information mining technique for movie recommendation system,” in *Proceeding of 2012 7th International Conference on Electrical and Computer Engineering*, pp. 462-465, 2013. DOI: 10.1109/ICECE.2012.6471587.

[3] P. Chen and X. Fu, “Research on sentiment classification of tests based on SVM,” *Journal of Guangdong University of Technology*, vol. 31, no. 3, pp. 95-101, 2014. DOI:10.3969/j.issn.1007-7162.2014.03.017.

[4] S. Tan, Y. Li, H. Sun, Z. Guan, amd X. Yan, “Interpreting the Public Sentiment Variations on Twitter,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1158-1170, 2014. DOI: 10.1109/TKDE.2013.116.

[5] N. Arunachalam, S. J. Sneka, and G. MadhuMathi, “A Survey on text classification techniques for sentiment polarity detection,” in *Proceeding of 2017 Innovations in Power and Advanced Computing Technologies* (*i-PACT*), pp. 1-5, 2017. DOI: 10.1109/IPACT.2017.8245127.

[6] J. M. Desai and S. R. Andhariya, “Sentiment analysis approach to adapt a shallow parsing based sentiment lexicon,” in *Proceeding of 2015 International Conference on Innovations in Information, Embedded and Communication Systems* (*ICIIECS*), pp. 1-4, 2015. DOI: 10.1109/ICIIECS.2015.7193160.

[7] Q. Li, S. Shah, R. Fang, A. Nourbakhsh, and X. Liu, “Tweet sentiment analysis by incorporating sentiment-specific word embedding and weighted text features,” in *Proceeding of 2016 IEEE/WIC/ACM International Conference on Web Intelligence* (*WI*), pp. 568-571, 2016. DOI: 10.1109/WI.2016.0097.

[8] THUCNews DataSet, [Online] Available: http://thuctc.thunlp.org/.

[9] C. Yu, “Adaptive japanese teaching optimization based on classification and regression tree,” in *Proceeding of 2017 International Conference on Robots & Intelligent System* (*ICRIS*), pp.15-18, 2017. DOI: 10.1109/ICRIS.2017.12.

[10] R. Li, X. Zhao, X. Yu, J. Li, N. Cheng, and J. Zhang, “Incident duration model on urban freeways using three different algorithms of decision tree,” in *Proceeding of 2010 International Conference on Intelligent Computation Technology and Automation*, pp..526-528, 2010. DOI: 10.1109/ICICTA.2010.602.

[11] R. Izmailov, V. Vapnik, and A. Vashist, “Multidimensional splines with infinite number of knots as SVM kernels,” in *Proceeding of the 2013 International Joint Conference on Neural Networks* (*IJCNN*), pp.1-7, 2013. DOI: 10.1109/IJCNN.2013.6706860.

[12] L. Zhou, L. Wang, X. Ge, and Q. Shi, “A clustering-Based KNN improved algorithm CLKNN for text classification,” in *Proceeding of 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics* (*CAR 2010*), pp. 212-215, 2010. DOI: 10.1109/CAR.2010.5456668.

[13] H. Yigit, “A weighting approach for KNN classifier,” in *Proceeding of 2013 International Conference on Electronics, Computer and - Computation* (*ICECCO*), pp. 228-231, 2013. DOI: 10.1109/ICECCO.2013.6718270.

[14] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification using machine learning techniques,” in *Proceeding of the Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Philadelphia, pp. 79-86, 2002,.

[15] P. D. Turney and M. L. Littman, “Measuring praiseand critism inference of semantic orientation from as sociaton,” *ACM Transon Information Systems*, vol. 21, no. 4, pp. 315-346, 2003.

[16] S. Taneja, C. Gupta, S. Aggarwal, and V. Jindal, “MFZ-KNN-A modified fuzzy based K nearest neighbor algorithm,” in *Proceeding of 2015 International Conference on Cognitive Computing and Information Processing* (*CCIP*), pp. 1-5, 2015. DOI: 10.1109/CCIP.

2015.7100689.

[17] J. Huang, Y. Wei, J. Yi, and M. Liu, “An improved kNN based on class contribution and feature weighting,” in *Proceeding of 2018 10th International Conference on Measuring Technology and Mechatronics Automation* (*ICMTMA*), pp. 313-316, 2018. DOI: 10.1109/ICMTMA.2018.00083.

Guangxing Wang

He received his M.S. degree in Computer Application Technology from Huazhong University of Science and Technology, Wuhan, China in 2009. From 2016 to the present, he has been an associate professor in the Information Technology Center of Jiujiang University in China. His research interests include data science, information system, and artificial intelligence.

Seong Yoon Shin

He received his M.S. and Ph.D. degrees from the Dept. of Computer Information Engineering of Kunsan National University, Gunsan, Korea, in 1997 and 2003, respectively. From 2006 to the present, he has been a professor in the same department. His research interests include image processing, computer vision, and virtual reality.