Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 10, October 2019, pp. 2618-2627
DOI: 10.23940/ijpe.19.10.p7.26182627

Multi-Classification Method for Determining Coastal Water Quality based on SVM with Grid Search and KNN

Guoqiang Xie, Yi Zhao, Shiyi Xie*, Miaofen Huang, and Ying Zhang

School of Mathematics and Computer, Guangdong Ocean University, Zhanjiang, 524088, China

Abstract

To address the problem of multi-classification of coastal water quality, this work envisioned the establishment of a multi-classification model of coastal water quality that uses an improved support vector machine. Inorganic nitrogen, active phosphate, chemical oxygen demand, pH, and dissolved oxygen were the input parameters of the model. The parameters of the support vector machine (SVM) model were optimized by cross-validation and the grid search optimization method, and the optimal parameters of the classifier were obtained. Subsequently, the KNN method was combined, and the optimized model was used to classify the water quality. The optimal parameters for the classifier were finally obtained. The experimental results showed that compared with SVM before optimization, the accuracy of the optimized model was improved by up to 10%, and the sample size was less.

Keywords: SVM; grid search; KNN; coastal water quality

(Submitted on September 19, 2019; Revised on September 30, 2019; Accepted on October 12, 2019)

1. Introduction

The content of marine water quality assessment is classified and assessed on the basis of the main components and content of the seawater body. Due to the high cost of collecting ocean samples and the large amount of substances in the water, it is very important to select a suitable classification algorithm to obtain the optimal results for the main substance component analysis. The water quality system is composed of multi-dimensional factors and is a complex system. According to the National Standard for Seawater Quality, there are as many as 35 related indicators affecting the classification of seawater quality, including the following indicators: N, P, dissolved oxygen (DO), chemical oxygen demand (COD), pH, oil, Hg, Cd, Pb, Cr, As, Cu, Zn, and so on. N, P, DO, COD, and pH are the main components, and they have more than 90% influence on water quality classification. There is no universally accepted classification method to determine the categories of water quality. The water quality evaluation methods that currently exist in literature mainly include the single factor index method, the fuzzy theory, and the grey theory. With the development of machine learning and deep learning theory, more suitable machine learning classification methods can be selected to classify water quality. The support vector machine (SVM) model is a convex quadratic programming problem with linear constraints based on the principle of structural risk minimization. The SVM algorithm obtains the best generalization ability based on the limited sample information, and it has been widely used in various domains [1]. Important factors affecting the performance of the SVM algorithm are related to penalty factor C and gamma. The KNN algorithm determines the category of the sample to be classified based on the category of K samples closest to the sample to be classified [2]. Both algorithms are widely used. The SVM and KNN algorithms are often combined for transportation [3], medicine [4], and food [5] purposes.

In this paper, we established a multi-classification support vector machine model for offshore water quality. The C and gamma parameters in the model were optimized by the grid search method. The cross-validation method was used to verify the generalization ability of the model. Subsequently, KNN was combined with SVM to classify offshore water quality. According to the sample of different data volume, the model with the highest accuracy was selected.

* Corresponding author.
E-mail address: xgq_for_stu@163.com

2. Original Algorithm

2.1. Support Vector Machine

Support vector machine (SVM) is a learning method based on the statistical learning theory. Through learning algorithms, support vectors can better distinguish classifications. The purpose is to maximize the distance between the classes [6]. To address the problem of nonlinear separability, it is transformed into linear separability in a high-dimensional space, and then a most suitable hyperplane is found. The SVM algorithm can find the best compromise between the complexity of the model and the learning ability according to the given limited sample information, and it can improve the generalization ability. There are many unique advantages when solving the practical problems of small samples, nonlinearities, and high dimensions. At the same time, it is an effective way to solve the problem of under-learning and learning. The key technology of SVM is choosing the kernel function. The form and parameters of the kernel function are generally different in different problem areas.

The mechanism of SVM is to find an optimal classification hyperplane that meets the classification requirements such that the classification interval is maximized. The classification interval is the distance between the two hyperplanes that are closest to the classification hyperplane and parallel to the classification hyperplane. These samples are the support vectors. The classification line equation is $w \times x + b = 0$. After normalization, a linearly separable sample set is made as (x_i, y_i), $i = 1, \cdots, n,\ x \in R^d,\ y \in \{+1, -1\}$.

Which satisfies

$$y_i(w \times x_i + b) - 1 \geq 0,\ i = 1, \cdots, n \tag{1}$$

Where $2/\|w\|$ is the classification interval, and $\|w\|^2$ is the smallest value that makes the interval the largest. The classification surface that satisfies Equation (1) and minimizes $\frac{1}{2}\|w\|^2$ is called the optimal classification surface. To solve this constrained optimization problem, the Lagrange function is introduced.

$$L(w, b, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i [y_i(w \times x_i + b) - 1] \tag{2}$$

Where $\alpha_i \geq 0$ is the Lagrange multiplier corresponding to each sample. The optimal classification function is

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i (x \times x_i) + b \right) \tag{3}$$

In the case of linear inseparability, a relaxation item $\xi_i \geq 0$ can be introduced, which becomes

$$y_i(w \times x_i + b) \geq 1 - \xi_i,\ i = 1, \cdots, n \tag{4}$$

On changing the target to the minimum,

$$\psi(w, \xi) = \|w\|^2 / 2 + C \sum_{i=1}^{n} \xi_i \tag{5}$$

Consider the maximum classification interval and the least misclassified sample, where $C > 0$ is a constant, indicating the degree of penalty for the wrong sample. In a linear indivisible practical problem, it is necessary to map the nonlinear transformation of the classification sample to a high-dimensional feature space, so that the target is linearly separable in the high-dimensional space. The kernel function maps the training samples in the high-dimensional feature spaces and maintains a linear separability. The following three kernel functions are commonly used:

- Polynomial kernel function:

$$K(x, x_i) = \{\gamma^* (x \times x_i) + coef\}^d \tag{6}$$

Where d is the order of the polynomial and $coef$ is the offset coefficient.

- RBF kernel function:

$$K(x, x_i) = \exp(-\gamma^* \|x - x_i\|^2) \tag{7}$$

Where γ is the width of the kernel function.

- Sigmoid kernel function:

$$K(x, x_i) = \tanh(\gamma(x \times x_i) + coef) \tag{8}$$

2.2. Multi-Class Classification Problem

The basic support vector machine can only solve the two classification problems. To solve the multi-classification problem, two methods are mainly used. One is to convert the multi-class classification problem into two classification problems. Common methods, like the 1-v-r method, 1-v-1 method, and DDAG method, are used [6]. The other is to extend the basic two types of classified SVMs into multi-class classification SVMs. The common methods used by SVM for multi-classification are as follows:

- The 1-v-r method constructs an SVM for each class. The training sample set is composed of the following: samples belonging to the class are positive samples, all other samples are negative samples, and each SVM is a two-class problem. Assuming the sample has k classes, k classifiers are needed. The decision values of each classifier are calculated separately, and the corresponding category with the largest function value is the category to which the data belongs.
- The 1-v-1 method was proposed by KNERR. It constructs a classifier for each of the two classes in the k class, constructing a total of $k(k-1)/2$ sub-classifiers. When classifying unknown samples, each sub-classifier makes a discriminant, and the result is to vote for the corresponding category. The class with the highest number of votes is the category to which the sample belongs.
- Plantt et al. proposed the DDAG SVMs method, which is a decision-oriented acyclic graph method. It constructs $k(k-1)/2$ classifiers to form a directed acyclic graph. Starting from the root node and according to the classification result, the next level of classifier is entered, and this is continued until the final category of leaf node.

2.3. KNN

The nearest neighbor (NN) method is one of the most important methods in the non-parametric method of pattern recognition. The original nearest neighbor method was proposed by Cover and Hart [2] in 1967. K nearest neighbor implies the examination of k samples, which are most similar to the samples to be classified. According to the category of k samples, the category attributes of the samples to be classified are determined.

KNN algorithm: Samples x_i and x_j obtain the most common f values for training k samples nearest to *f*. According to Euclidean distance x_q, it can be described as

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{m} \left(x_i^l - x_j^l\right)^2} \tag{9}$$

$$f(x_q) \longleftarrow \arg_{v \in V} \max \sum_{i=1}^{k} w_i \delta\left(v, f\left(x_i\right)\right) \tag{10}$$

Where w_i is the weighted distance, given as

$$w_i = \frac{1}{d(x_q, x_i)^2} \tag{11}$$

The classification discriminant function is

$$\begin{cases} \delta(a,b) = 1, \text{ if } (a = b) \\ \delta(a,b) = 0, \text{ else} \end{cases} \tag{12}$$

$x = (x^1, x^2, \cdots, x^m)$ represents a point corresponding to the *n*-dimensional space R^n, x_i represents the value of the i^{th} feature in the sample x, and $f(x_i)$ represents the category of x_i.

2.4. Grid Search

The grid search algorithm (GSA) is a basic parameter optimization method. Its principle is to optimize the parameters in steps within a certain spatial range and find the parameters that make the model performance optimal. The grid search method is simple, convenient, easy to understand, and can search multiple parameter values at the same time. When the sample size is small, the speed of the run is fast [7]. Specifically, in the SVM classification, the RBF kernel function is used, and the penalty factor *C* and the kernel function parameter gamma are the two parameters to be determined. Based on the grid method, the range of parameter *C* is from C_1 to C_2, and the step size is *Cs*, while the gamma ranges from g_1 to g_2, and the step size is g_s. The model is trained for each set of parameters (C_x, g_x), and the ability to promote the model is tested. Finally, the optimal set of parameters is selected as the model parameters.

2.5. Cross-Validation

Cross-validation is a way to reduce the bias caused by the random selection of samples and to verify the stability and generalization ability of the model [8]. Commonly used cross-validation methods include the *K*-fold cross-validation, the leave-one method, and the repeated random sampling method. Considering the sample size and computational efficiency, we used the *K*-fold cross-validation method. Specifically, the training set was divided into *K* equal subsets, each of which took *K*-1 data as the training data, and the remaining one was used as the test data for *K* times to ensure that each test set was different.

3. Improved Algorithm

3.1. Basic Idea

The SVM classification effect mainly depends on the kernel function, the parameter gamma in the kernel function, and the selection of the penalty coefficient *C* for the wrong subsample. Due to the sample size, an easy-to-use grid search method suitable for small samples was selected for the optimization of parameters. The use of the optimized SVM model could not achieve the required accuracy. Due to the characteristics of the SVM algorithm, the classification effect of the sample farther from the classification surface was better, and the misclassified sample was most likely concentrated near the optimal classification surface. Therefore, for this part of the data, an appropriate algorithm was needed for optimal classification. The KNN algorithm implements the classification step by selecting the K nearest neighbors closest to the sample, and counting the most plural of these nearest neighbors is the final category of the sample to be tested. At the same time, in order to test the stability and scalability of the model, the K-fold cross-validation method was selected to verify the model. K-fold cross-validation can make full use of all the data in the training set; even when the data is small, it can achieve better results.

3.2. Algorithm Main Flow

Step 1 The parameters were optimized by the grid optimization method. The grid parameters were optimized for the kernel function, the penalty factor, and the kernel function parameter gamma. The range of kernel function parameters was {Polynomial, RBF, Sigmoid}. Considering the running time of the model, the parameter range of the penalty factor C was initially set from (0, 100), the step size was 1, the parameter range of gamma was (0, 50), and the step size was 1. Using the cross-validation method, according to the results of the different parameter combinations on the training set, a set of parameters that made the model effect optimal was selected.

Step 2 According to the result of the previous step, the range of the parameter to the range near the optimal parameter was reduced, and the step size was set to 0.01. If there were multiple sets of parameters that made the model effect optimal, the choice of *C* was the smallest. *C* represents the penalty coefficient for the wrong sample; the larger the *C*, the worse the promotion ability of the model. Hence, a smaller value of *C* was selected.

Step 3 Using the optimized SVM classifier, the value of the optimal classification plane for each sample distance was obtained, and the model was classified using the optimized SVM classifier for samples outside the distance from the optimal classification surface. Samples smaller than a certain value were classified using KNN. For the selection of the distance value, when training the sample, the SVM algorithm optimized by the grid search was used to obtain the distance of each sample from the optimal classification plane, and the i^{th} value in the ranked list was selected according to the arrangement from largest to smallest. In this case, '*i*' is the value of the product of the accuracy of the SVM on the training set and the length of the training set.

Step 4 According to the distance obtained in the previous step, the optimized SVM was initially used on the test set to determine the distance from the hyperplane for each sample on the test set. If the distance obtained from the previous step was greater than this distance, the SVM algorithm was used to obtain the algorithm. The results were superimposed on the training set and trained using the KNN algorithm. The remaining samples of the test set were trained using the trained KNN algorithm. The results of the SVM and KNN algorithms were compared with the results of the test set to obtain an accuracy rate.

Step 5 A different number of training sets were selected, the above steps were repeated, and the optimized model was selected to classify the water quality.

The specific steps for searching the SVM parameters *C* and gamma using the grid are shown in Algorithm 1.

3.3. Algorithm Pseudo Code Representation

Algorithm 1: Uses the grid to search for SVM parameters C and gamma

```
Input: C range cArray, gamma range gArray
Output: optimal parameters cBest, gBest,resultSVM, and distance
1:function GridSearchSVM(cArray,gArray)
2:   param_grid={"gamma":gArray,"C":cArray}
3:   grid_search=GridSearchCV(SVC(),
              param_grid,cv=kflod)     #Grid search
4:   grid_search.fit(x_train,y_train)
                                    #Training set training
5:   cBest=grid_search.best_params_('C')
                                          #Optimal C value
6:   gBest=grid_search.best_params_('gamma')
                                #Optimal gamma value
7:   resultSVM=grid_search.best_score
                                     #Training set result
8:   clf=SVC(kernel='rbf',gamma=gBest,C=cBest)
10:  decision_train=clf.decision_ function(x_train)
#Save the value of the optimized SVM's decision into the list
12:  for max_i in range(len(decision_train))do
13:      max _decision_train. append (max
                          (decision_train(max_i)))
14:  end for
              #Save the decision maximum to the list
15:  max_decision_train.sort()
                           #Sort from big to small
16:  distance=max_decision_train(int(
     (1-(result_SVM))*len(x_train)+0.5))
  #Take values from a sorted list
17:  returncBest,gBest,resultSVM,distance
18: end function
```

Algorithm 2: Optimized SVM combined with KNN

```
input: cBest,gBest,resultSVM
ouput: result_SVM_KNN
1:function SVM_KNN(cBest,gBest,resultSVM )
2:  SVM_GridSearch=SVC(kernel=
                                                'rbf',gamma=gBest,C=cBest)
                                    #Optimized SVM model
3:  decision_function=clf. decision_ function(x_test)
```

```
    #Save the value of the optimized SVM to the list
4:  for num in range(len( decision_function))do
5:    if(max(decision_function(num))>distance)then
          SVM_GridSearch(x_train,x_test)
    #Optimized SVM training model using grid search
          max_list.append(num)              #Record samples far from the hyperplane into the table
6:    elsethen
7:        x_train_KNN=x_train+x_test(num)
8:        y_train_KNN=y_train+y_test(num]
9:        x_test_KNN=x_test-x_test(num)
10:       y_test_KNN=y_test-x_test(num]
                          #Samples with a hyperplane exceeding distance are stored in the training set
11:       KNN(x_train_KNN,y_train_KNN)
                                                #Training with KNN
12:   end if
13: end for
14: res_SVM_KNN=res_SVM+res_KNN
                                                #Comprehensive SVM and KNN results
15: return res_SVM_KNN   #Return the final result
16: end function
```

4. Experimental Results and Analysis

4.1. Data Sources

According to the measured data of offshore water quality, 2,000 simulation data samples were used for simulation verification and analysis (the range of analog data ranged from the maximum measured value to the minimum measured value). Because the form is limited, six water quality samples are listed for each type. The data samples are shown in Table 1.

Table 1. Data samples

Type	N	P	COD	pH	DO
1	0.12856	0.004418	1.3854	8.4536	6.107
1	0.15937	0.003897	1.3617	8.3677	6.7242
1	0.13705	0.006952	1.2876	8.1392	6.6137
1	0.17956	0.014818	1.8167	8.3297	6.783
1	0.10735	0.005776	1.4505	8.0919	6.5666
1	0.19927	0.014149	1.8066	8.4803	6.8113
2	0.28344	0.020033	2.3254	7.9255	5.3354
2	0.23582	0.027665	2.6734	7.9975	5.5106
2	0.28059	0.019322	2.6853	8.4832	5.3852
2	0.25389	0.028632	2.6957	7.8255	5.3106
2	0.25901	0.016698	2.7998	8.0284	5.0036
2	0.22311	0.024208	2.6606	8.4811	5.8152
3	0.38981	0.01848	3.6241	8.1964	4.3118
3	0.36218	0.017939	3.7252	8.2675	4.8952
3	0.34146	0.027745	3.8344	8.1011	4.8348
3	0.36476	0.026124	3.0189	7.8325	4.0023
3	0.34893	0.024933	3.2021	7.4528	4.6402
3	0.33224	0.022059	3.3998	7.8315	4.5004
4	0.4274	0.037354	4.9862	8.1698	3.4704
4	0.40301	0.034776	4.349	8.5233	3.8291
4	0.44411	0.037408	4.0537	6.8653	3.2674
4	0.47007	0.03943	4.0472	7.4639	3.1762
4	0.4703	0.041297	4.0559	8.2975	3.4312
4	0.45102	0.040408	4.0715	8.0887	3.4757

4.2. Data Analysis and Calculation

In order to analyze the characteristics of the data, the parameters of the maximum, minimum, mean, variance, and standard deviation of the feature were used to analyze and process the data. Statistical analysis was performed on four types of samples, and the results are shown in Tables 2-5.

Table 2. Data characteristics of water quality type 1

Type = 1	N	P	COD	pH	DO
Average value	0.15277	0.00750	1.47844	8.15035	6.51869
Variance	0.00086	1.82445	0.07938	0.03886	0.07828
Maximum	0.19989	0.01498	1.99820	8.49780	6.99980
Minimum	0.10001	0.00002	1.00150	7.80080	6.00150

Table 3. Data characteristics of water quality type 2

Type = 2	N	P	COD	pH	DO
Average value	0.24937	0.02209	2.49215	8.16146	5.50664
Variance	0.00081	1.89840	0.08280	0.04167	0.08920
Maximum	0.29977	0.02993	2.99840	8.49960	5.99840
Minimum	0.20011	0.01500	2.00130	7.80020	5.00020

Table 4. Data characteristics of water quality type 3

Type = 3	N	P	COD	pH	DO
Average value	0.35309	0.02255	3.50533	7.80742	4.50371
Variance	0.00085	1.85587	0.08147	0.32493	0.08426
Maximum	0.39994	0.02999	3.99860	8.78240	4.98400
Minimum	0.30001	0.015014	3.00480	6.80130	4.00230

Table 5. Data characteristics of water quality type 4

Type = 4	N	P	COD	pH	DO
Average value	0.44900	0.03737	4.49626	7.82325	3.49598
Variance	0.00079	1.81510	0.08519	0.33393	0.08261
Maximum	0.49941	0.04498	4.99810	8.79960	3.99690
Minimum	0.40019	0.03003	4.00040	6.80200	3.00030

From the results above, it can be seen that the distinction between different categories of data was more obvious. The maximum and minimum limits of each type of feature were obvious, the values of variance and standard deviation were relatively small, and the stability of the data was relatively high.

4.3. Comparison Algorithm

- Random forest algorithm

The random forest algorithm is an extended variant of Bagging in integrated learning based on decision tree. The traditional classification model decision tree is often inaccurate, and it is prone to over-fitting problems [9].

- Gaussian naive Bayesian classification algorithm

The Bayesian algorithm infers the result by probability, which is simple, efficient, and characterized by strong interpretability [10]. The naive Bayesian algorithm assumes that conditions are independent of each other. Assuming that the conditions between the attributes are independent, there is often no more or less connectivity between the attributes in the real problem.

- Standard SVM algorithm

The mechanism of SVM is to find an optimal classification hyperplane that meets the classification requirements, so that the classification interval is maximized.

- Standard KNN algorithm

KNN is the most similar sample to be sampled and to be classified, and the category attribute of the sample to be classified is determined according to the category of the sample.

4.4. Comparison and Analysis of Experimental Results

Considering the difficulty in collecting seawater samples, the sample size was relatively small. In order to study how to choose the optimal classification algorithm from different classification algorithms in the case of less samples, this experiment was divided into two situations, and the graph and table were used separately for a better display of the

classification results. In first case, the fixed test set was 60, and the training set used by the training set varied from 1 0 to 90. The reason for choosing a test set of 60 was that too few test sets may cause contingency and inaccuracy of the classification results. In the second case, the fixed training set was 30, and the test set varied from 10 to 90. Graphical methods were employed to observe the changes in each algorithm. The accuracy of each algorithm was specifically compared using a table approach.

When using the KNN algorithm, it is necessary to determine the value of *K*. In this paper, the amount of data was set from 40 to 120 by experiment, and the results of *K* from 1 to 10 were obtained respectively. According to statistics, when *K* = 4, the classification accuracy was higher than other values, so this paper employed the nearest neighbor algorithm when *K* was 4. The statistical method calculated the sum of the classification accuracy of a total of 80 times under different data quantities, and then the most suitable *K* value was found. The results are shown in Table 6.

Table 6. Accuracy of KNN different *K* values

K-value	1	2	3	4	5	6	7	8	9	10
Accuracy	0.9777	0.9777	0.9826	0.9843	0.9840	0.9790	0.9760	0.9755	0.9775	0.9759

The results of Table 6 indicate that when the K value was 4, the classification result was the best, so we selected the K value of 4.

(1) When the test set was fixed at 60, the training set was from 10 to 90, and the accuracy results of each algorithm are shown in Figure 1.

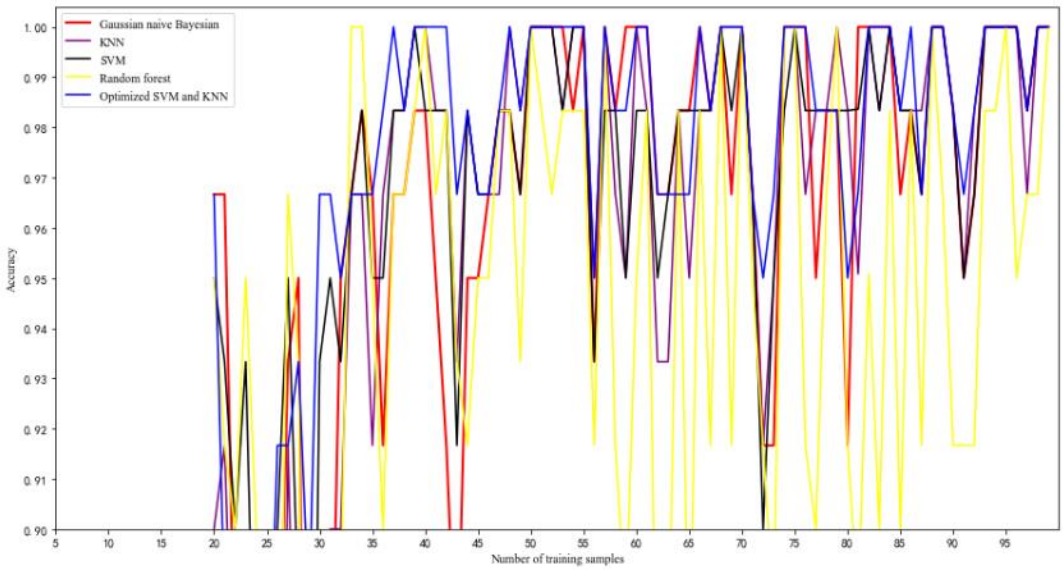

Figure 1. Data classification diagram

From Figure 1, we can conclude that the optimized SVM algorithm exhibited a higher accuracy than the random forest algorithm, Bayesian algorithm, SVM, and KNN when the training set samples were from 10 to 90. In addition to the training sample size of 35, the accuracy of the random forest algorithm was slightly higher by 0.02.

The results can bring a guiding significance to practical applications. When the sample size is small, the optimized SVM algorithm can be selected, and a high accuracy can be obtained.

The results of the optimized algorithm classification are shown in Table 7, where: Group 1 data: training set 30, test set 60; Group 2 data: training set 25, test set 60; Group 3 data: training set 20, test set 60; Group 4 data: training set 25, test set 30.

Table 7. Results of optimized algorithm classification

Experimental group	Gaussian Bayes	Random forest	Traditional SVM accuracy	Traditional KNN accuracy	Optimized SVM-KNN accuracy
Group 1	0.85	0.90	0.9333	0.8667	0.9667
Group 2	0.75	0.7167	0.8333	0.6667	0.85
Group 3	0.9667	0.90	0.95	0.90	0.9667
Group 4	0.5667	0.6667	0.6667	0.6667	0.7667

(2) For the second case, when the training set was fixed at 30 and the test set was from 10 to 90, the accuracy results of each algorithm are shown in Figure 2.

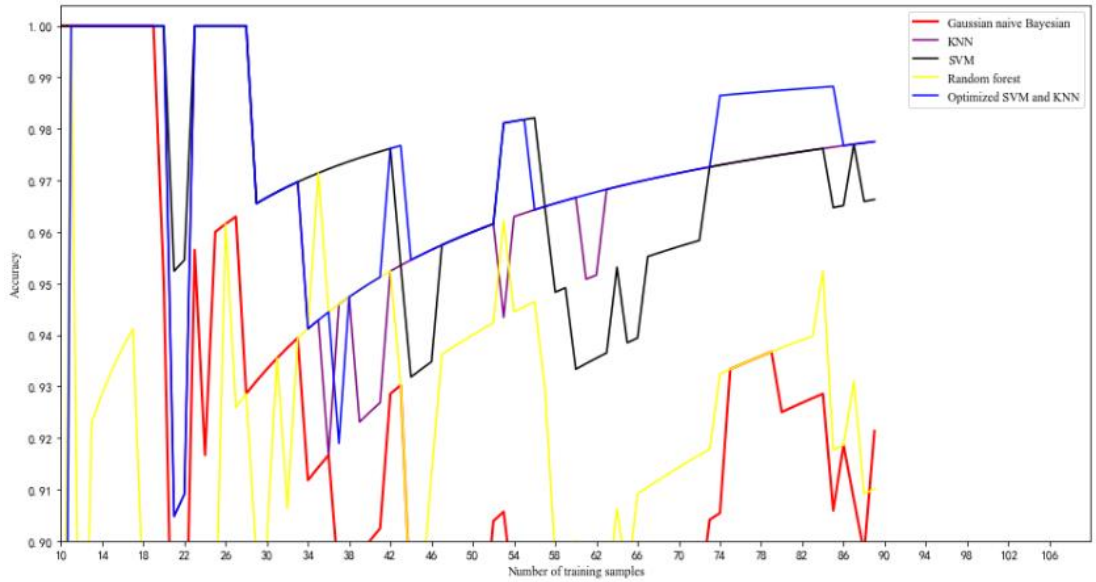

Figure 2. Data classification diagram

Our experimental results show that the accuracy of the optimized SVM-KNN algorithm was much higher than that of the other algorithms. The reason for this result is that when training samples, a grid of search and cross-validation methods were used to obtain a set of parameters C and gamma with better classification accuracy and generalization results, wherein each of the training samples in Figure 1 corresponds to a different set of C and gamma values. From Figure 2, the optimal C value was 0.751, the gamma was 1.451, and the accuracy of the SVM classification was lower after using the grid search, when compared to the SVM with the default parameter $C = 1$ and $gamma = 0.25$. The reason is that the training sample size was small, so that the optimized SVM did not have the characteristics of the overall sample. This problem did not occur when the sample size was large. The trained sample was then trained using the optimized SVM to obtain an accuracy rate on the training set. The product of the accuracy rate on the training set was calculated, and the length of the training sample was *i*. The distance was calculated of each sample of the training sample from the SVM optimal hyperplane from large to small, and the i^{th} value was selected as the distance. For each sample on the test set, the mesh-optimized SVM algorithm was used to classify each sample from the optimal hyperplane. If the distance was greater than the distance obtained in the previous step, the SVM algorithm was used for classification, and the sample and the classification result were added to the training set of the KNN algorithm. If the sample was smaller than the distance, the KNN algorithm was used for training. The final result was a synthesis of KNN and SVM classification results.

5. Research Status at Home and Abroad

In recent years, the domestic models for water quality classification have focused on the gray clustering method, comprehensive index method, fuzzy evaluation method, etc. These methods need to implement the assumption mode or subjectively set some parameters, so that the evaluation results can have a certain subjectivity. The artificial neural network is used to evaluate the water quality and simulates the human thinking mode. The classification does not need to set a certain pattern in advance and only trains according to the characteristics of the object itself. However, due to the over-fitting phenomenon of the neural network itself, the accuracy of the model is low, and the generalization ability of the model is weak. SVM is a statistical-based machine learning method that is superior in solving dimensional disasters and local minimum problems. SVM solves the convex quadratic programming problem with a linear constraint based on the principle of structural risk minimization. The SVM algorithm can obtain the best generalization ability based on limited sample information, and it has been widely used in various fields. Unlike classification, the clustering method can determine the category of the sample accurately if the sample does not have a certain category. For example, LDA is combined with word embedding, using related words obtained based on word embedding to improve the performance of Web service clustering [11]. ST-LDA (short for "Similar Words and TF-IDF Augmented Latent Dirichlet Allocation") approaches these challenges from the perspective of similar word learning and noise word filtering to improve service clustering [12].

6. Conclusions

In this paper, we used grid search to optimize the parameters C and gamma of SVM. We used the SVM training samples after grid optimization to obtain a certain value as the distance A according to the distance of each sample from the optimal hyperplane and the classification accuracy of SVM after parameter optimization. For the samples of the test set, the distance between each test sample and the hyperplane was obtained by using the grid-optimized SVM. When comparing this distance with distance A., it was found that distance A. was larger than distance A using grid optimized SVM classification, and the corresponding samples were stored in the training set of the KNN algorithm. For test samples less than distance A, the KNN algorithm was used to classify them. The results of the two algorithms were synthesized to classify the samples. The experimental results show that the optimized SVM and KNN models are superior to other commonly used algorithms in the classification of offshore water quality, and they also use fewer samples. The optimized model makes full use of the samples. This means that for the problem of high cost of sample collection, the optimized model can be effectively applied to cases involving fewer samples. The next step in our research is to subdivide all data sets and obtain the corresponding optimal model.

References

1. X. D. Wu, V. Kumar, R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, et al., "Top 10 Algorithms in Data Mining," *Knowledge and Information Systems*, Vol. 14, pp. 1-37, December 2007
2. T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, Vol. 13, No. 1, pp. 21-27, January 1967
3. J. L. Xiao, "SVM and KNN Ensemble Learning for Traffic Incident Detection," *Physica A*: *Statistical Mechanics and Its Applications*, Vol. 517, pp. 29-35, March 2019
4. T. Pereira, J. S. Paiva, C. Correia, and J. M. Cardoso, "An Automatic Method for Arterial Pulse Waveform Recognition using KNN and SVM Classifiers," *Medical and Biological Engineering and Computing*, Vol. 54, No. 7, pp. 1049-1059, July 2016
5. N. Gerhardt, S. Schwolow, S. Rohn, P. R. Perez-Cacho, H. Galan-Soldevilla, L. Arce, et al., "Corrigendum to 'Quality Assessment of Olive Oils based on Temperature-Ramped HS-GC-IMS and Sensory Evaluation: Comparison of Different Processing Approaches by LDA, kNN, and SVM'," *Food Chemistry*, Vol. 286, pp. 307-308, February 2019
6. C. W. Hsu and C. J. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, pp. 415-425, March 2002
7. X. Gao and J. Hou, "An Improved SVM Integrated GS-PCA Fault Diagnosis Approach of Tennessee Eastman Process," *Neurocomputing*, Vol. 174, pp. 906-911, January 2016
8. S. Varma and R. Simon, "Bias in Error Estimation When using Cross-Validation for Model Selection," *BMC Bioinformatics*, Vol. 7, No. 1, pp. 91, February 2006
9. D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, et al., "Random Forests for Classification in Ecology," *Ecology*, Vol. 88, No. 11, pp. 2783-2792, December 2007
10. H. Zhang, "Exploring Conditions for the Optimality of Naïve Bayes," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 19, No. 2, pp. 183-198, March 2008
11. Y. Zhao, C. Wang, J. Wang, and K. He, "Incorporating LDA with Word Embedding for Web Service Clustering," *International Journal of Web Services Research*, Vol. 15, No. 4, pp. 29-44, October 2018
12. Y. Zhao, K. He, and Y. Qiao, "ST-LDA: High Quality Similar Words Augmented LDA for Service Clustering," in *Proceedings of 18th International Conference on Algorithms and Architectures for Parallel Processing* (*ICA3PP*), pp. 46-59, Guangzhou, China, November 2018