2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
Banff Center, Banff, Canada, October 5-8, 2017

A Between-Class Overlapping Coherence-Based Algorithm in KNN Classification

[1]Nian Zhang, [1]Welezane Karimoune, [2]Lara Thompson, and [1]Hongmei Dang
[1]Department of Electrical and Computer Engineering
[2]Department of Mechanical Engineering
University of the District of Columbia
Washington, D.C., USA
{nzhang, k. Welezane-gazere, lara.thompson, hongmei.dang}@udc.edu

***Abstract*—This paper proposes an improved KNN algorithm to overcome the *class overlapping* problem when the class distribution is skewed. Different from the conventional KNN algorithm, it not only finds out the k nearest neighbors of each sample (even the test object itself) in the training dataset, but also the neighbors of the unknown test object. Then the validity value of a data point is computed based on the label of the data and the labels of its k nearest neighbors. A classifier is designed to assign the unknown test object to a class membership based on the proposed validity ratings equations. A numerical analysis provides a detailed example to demonstrate the effectiveness of the algorithm. The improved KNN algorithm is compared with the conventional KNN and the modified KNN algorithm on the real world wine data with the increasing number of k from 1 to 20. The experimental results show that the proposed improved KNN algorithm outperforms the conventional KNN and the modified KNN algorithm. In addition, classification accuracy of KNN algorithm and our algorithm in terms of various combinations of k-value and k-fold cross validation are compared. It is found that while the classification accuracy of the conventional KNN algorithm has changed drastically, our algorithm remains constantly high over the k values. Additionally, the conventional KNN algorithm has shown a declining trend when k is large, i.e. k = 20, while our algorithm remains stable.**

***Keywords*—k-nearest neighbors (KNN) classification; class overlapping; modified k-nearest neighbors (MKNN); supervised classification; k-fold cross validation**

I. INTRODUCTION

In recent years, the use of improvised explosive devices (IEDs) throughout the world has been the primary weapon of terrorist groups. Chemical signatures are not constant and depend heavily on environmental conditions and duration of emplacement. Insurgent countermeasure innovations in response to defeat technologies have demonstrated that it is important to develop detection schemes that are device independent [1]. However the ability to detect small amounts of analytes across large relevant substrates is complicated by the optical and thermal interactions between analyte and substrate [2]. The primary challenge is to distinguish explosives from the substrates, such as glass or clothing. While glass or clothing is chemically distinct from explosives, they nonetheless have overlapping infrared absorption/emission features with explosives. Further complications are the facts that polymeric materials tend to absorb and emit throughout the IR. Therefore, the development of analytical tools that can identify explosive remains is of tremendous importance in the forensic field for crime-scene reconstruction [3].

The k-nearest neighbor's algorithm (KNN), known as a non-parametric method used for classification and regression, was first proposed by Fix and Hodges in 1951 [4]. Whenever the data distributions are either unidentified or unreachable in many physical perspective applications, the non-parametric approaches are required. The k closest training data are within the inputs, where the output is designated to be an affiliated class. To improve the accuracy rate of KNN in the process, several modified KNN have been anticipated and enhanced. Hou and Gao presented a new weighted average combination method by integrating multiple kernel learning (MKL) into the kNN framework [5]. An extended k-nearest neighbor's symbolized as ENN was formulated by Tang and He [6]. Parvin et al roughly created type of resemblances between the training data, and gain supplementary knowledge on the weight of each neighbor [7]. Taneja et al suggested a constructed k-nearest neighbor's algorithm named fuzzy logic. Through this method, fuzzy clusters are gained prior to a processing phase, and then a calculation of the affiliated class with respect to the clusters centroid is evaluated [8]. Zhang et al proposed an enhanced k-nearest neighbor classification method based on maximal coherence and validity ratings [9].

Although these algorithms demonstrated that the classification accuracy performs better than the conventional KNN, their performance is unknown when the sample sizes are small or the data are highly overlapped. Therefore, it's important to develop an upgraded the KNN algorithm to further improve the accuracy and compare with the previous variations. We propose an improved KNN variation algorithm by exploring the performance on the same datasets (i.e. wine dataset) used by other researchers for the purpose of comparison, which has small sample size versus much larger feature size. Unlike the conventional KNN algorithm where only the nearest neighbors are used to determine the class of

978-1-5386-1645-1/17/$31.00 ©2017 IEEE

the test object, we also include the test object to the training dataset to maximize the intra-class coherence and then make a classification decision.

The remaining of the paper is organized in the following organization. In Section 2, the class overlapping problem for a two-class classification problem is formulated. In addition, the improved k-nearest neighbor algorithm is described. In Section 3, we provide a numerical example on the two-class classification problem and derive the output according to our algorithm. In Section 4, experimental results are presented. In Section 5, the conclusions are provided.

II. The Enhanced K-Nearest Neighbor Algorithm

A. Class Overlapping Problem for Two-Class Classification Problem

In the conventional KNN classification algorithm, an object is assigned to the majority class among its k nearest neighbors (k is a small positive integer). If the object has only one neighbor, then it is simply assigned to the class of that single nearest neighbor. A drawback of such "majority voting" classification has occurred when the class distribution is skewed. That is, instances of most frequently occurring class tend to dominate the assignment of the new test object, because they tend to be prevalent among the k nearest neighbors due to their large number. In order to overcome this problem, we proposed an improved KNN algorithm will not only find out the k nearest neighbors (including the unknown test object) of each sample in the training dataset, but also the unknown test object. Then we use a concept of validity rating to quantify the degree to which a pre-determined group of samples resemble their k nearest neighbors. Finally, a classifier will assign the unknown test object to a class membership based on the validity ratings.

In some classification applications such as remote explosive detection, fraud detection, and network intrusion detection, a highly overlapped dataset, it often occurs that samples from different classes have very similar signature or fingerprint as they reside in overlapping regions in the feature space. The problem is known as the *class overlapping* problem. It has become one of the toughest problems in data mining and business intelligence communities. It is major factor of bad classification performance. Fig. 1 illustrates a typical case of class overlapping, which would pose several great challenges for classification. In this figure, X1 is overlapped with Y2, and X3 is overlapped with Y3. In addition, an unknown test object, P to be classified is closest to X1 (Class 1), but it is closer to Y1 and Y2 (Class 2) on the top left than X2 and X3 (Class 1) on the bottom right. Thus it is hard to determine its correct classification based on the traditional KNN algorithm.

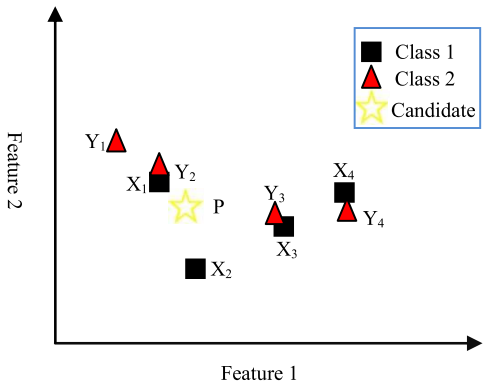

Fig. 1. A two-class classification example is illustrated in this figure. X1 is overlapped with Y2, and X3 is overlapped with Y3. In addition, an unknown test object, P to be classified is closest to X1 (Class 1), but it is closer to Y1 and Y2 (Class 2) on the top left than X2 and X3 (Class 1) on the bottom right.

Let T = {X_1, X_2, …, X_m, Y_1, Y_2, …, Y_n} be the training data, where {X_1, X_2, …, X_m} has a given class label, C_1, and {Y_1, Y_2, …, Y_n} has a given class label, C_2. P is an unknown test candidate data that we want to classify.

B. Concept of Validity

The value of validity indicates to which degree a data point looks like its *kth* nearest neighbors. The validity of a data point is computed based on the label of the data and the labels of its k nearest neighbors, as defined in (1). The validity varies within the range between 0 and 1. The larger the value, the more it tends to resemble its neighbors.

$$V(x) = \frac{1}{k}\sum_{i=1}^{k} S(label(x), label(\mathrm{T}_i)) \qquad (1)$$

Where k is a pre-defined number of nearest neighbors. *label(x)* is the class membership of data x. $label(\mathrm{T}_i)$ is the class membership of the ith nearest neighbor of x. T_i stands for the ith nearest neighbor of x inside T. S is a function representing the similarity between x and its ith nearest neighbor. The function S is defined in (2).

$$S(i,j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \qquad (2)$$

C. Validity Rating

In this way, each training sample will attain a validity attribute based on the similarity of the labels of the test object and its surroundings. The unknown test object will be tentatively assigned to a class membership based on some defined criteria, and form a group.

Specifically, we will first determine the k nearest neighbors (including the unknown test object itself) of each sample in the training set as well as the unknown test object. For the test object, P, k_1 number of its nearest neighbors belongs to class 1, while k_2 number of its nearest neighbors comes from class 2. We first predict the test object, P as Class 1, then we count the increasing number of Class 1 data among

the k nearest neighbors of the samples (denoted as Δn_1^1). We also count the decreasing number of Class 2 data among the k nearest neighbors of the samples (denoted as Δn_2^1). On the other hand, we predict *P* belongs to Class 2, and then we count the number of decreasing number of Class 1 data among the k nearest neighbors of the samples (denoted as Δn_1^2). We then count the increasing number of Class 2 data among the k nearest neighbors of the samples (denoted as Δn_2^2).

Then we calculate the validity ratings to quantify the degree to which the aforementioned group of samples resembles their k nearest neighbors, as shown in (3).

$$M_i(x) = \frac{1}{N+k}\sum_{x\in C}\sum_{i=1}^{k} S(label(x), label(\mathrm{T}_i)) \quad (3)$$

i = 1, 2

Where *N* is the size of samples determined by criteria, *C*. *k* is a pre-defined number of nearest neighbors. The criteria, *C* are defined in (4):

$$C = \{\mathrm{Samples in\ Class}_i\},\ i = 1, 2 \quad (4)$$

P is the unknown test sample. *Label(x)* is the class membership of data *x*. $label(\mathrm{T}_i)$ is the class membership of the *i*th nearest neighbor of x. T_i stands for the *i*th nearest neighbor of x inside T. *S* is a function representing the similarity between *x* and its *i*th nearest neighbor. The function *S* is defined in (2).

D. Design of Classifier

A classifier will take into account the coherence and validity ratings, and assign the class label associated to the maximum coherence to the unknown test sample, as defined in (5) [6].

$$Classifier = \arg\max_{j\in 1,2,N}\{\frac{\Delta n_i^j + k_i - kM_i}{(n_i+1)k} - \frac{\Delta n_i^j}{n_i k}\} \quad (5)$$

III. NUMERICAL ANALYSIS

In order to demonstrate the effectiveness of the algorithm described in Section II, we provide a numerical analysis corresponding to the enhanced KNN algorithm.

If we assume there are 3 nearest neighbors (excluding the unknown test object, *P*) of each sample in the training dataset, as well as the unknown test object, as shown in (6). In the training dataset *T*, {X_1, X_2, …, X_m} has a given class label, C_1, and {Y_1, Y_2, …, Y_n} has a given class label, C_2. *P* is an unknown test object that we want to classify.

$$\begin{array}{ll} NN_{1,2,3}(X_1)=[Y_2,Y_1,X_2] & NN_{1,2,3}(Y_1)=[Y_2,X_1,X_2] \\ NN_{1,2,3}(X_2)=[X_3,Y_3,X_1] & NN_{1,2,3}(Y_2)=[X_1,Y_1,X_2] \\ NN_{1,2,3}(X_3)=[Y_3,X_4,Y_4] & NN_{1,2,3}(Y_3)=[X_3,X_4,Y_4] \\ NN_{1,2,3}(X_4)=[Y_4,X_3,Y_3] & NN_{1,2,3}(Y_4)=[X_4,X_3,Y_3] \end{array} \quad (6)$$

Then the unknown test object will be tentatively assigned to a class membership based on criteria *C*, and then get involved in the intra-class correlation computation.

First we assume $\mathrm{P} \in \mathrm{C}_1$, and then solve for M_1 and M_2, respectively, as defined in (3).

Solve for M_1 : Given $P \in C_1$, i=1, $\mathrm{T}_i = C_1 \cup C_2$, $C = C_1 = \{X_1, X_2, X_3, X_4\}$, $N = 4$, k = 3

According to (3), $M_1(x) = \frac{1}{N+k}\sum_{x\in C}\sum_{i=1}^{k} S(label(x), label(\mathrm{T}_i))$. The validity ratings can be calculated using (1), (2), and (6). Thus,

$$M_1 = \frac{1}{4+3}\sum_{x\in(X_1,X_2,X_3,X_4)}\sum_{i=1}^{3} S(label(x), label(\mathrm{T}_i))$$

$$= \frac{1}{8}\times\begin{bmatrix} S_1(X_1,Y_2)+S_2(X_2,Y_1)+S_3(X_3,X_2)+ \\ S_1(X_2,X_3)+S_2(X_2,Y_3)+S_3(X_3,X_1)+ \\ S_1(X_3,Y_3)+S_2(X_2,X_4)+S_3(X_3,Y_4)+ \\ S_1(X_4,Y_4)+S_2(X_2,X_3)+S_3(X_4,Y_3) \end{bmatrix} = 0.42$$

Solve for M_2 : Given $P \in C_2$, i=2, $NN_i(x, \mathrm{T}) = C_1 \cup C_2$, $C = C_2 = \{Y_1, Y_2, Y_3, Y_4\}$, $N = 4$, k = 3

According to (3), $M_2(x) = \frac{1}{N+k}\sum_{x\in C}\sum_{i=1}^{k} S(label(x), label(\mathrm{T}_i))$. The validity ratings can be calculated using (1), (2), and (6). Thus,

$$M_2 = \frac{1}{4+3}\sum_{x\in(Y_1,Y_2,Y_3,Y_4)}\sum_{i=1}^{3} S(label(x), label(\mathrm{T}_i))$$

$$= \frac{1}{7}\begin{bmatrix} S_1(Y_1,Y_2)+S_2(Y_1,X_1)+S_3(Y_1,X_2)+ \\ S_1(Y_2,X_1)+S_2(Y_2,Y_1)+S_3(Y_2,X_2)+ \\ S_1(Y_3,X_3)+S_2(Y_3,X_4)+S_3(Y_3,Y_4)+ \\ S_1(Y_4,X_4)+S_2(Y_4,X_3)+S_3(Y_4,Y_3) \end{bmatrix} = 0.33$$

Next, we re-evaluate 3 (assume k = 3) nearest neighbors of each sample in the training dataset including the unknown test object, *P*, as shown in (7).

$$\begin{array}{ll} NN_{1,2,3}(X_1)=[Y_2,P,Y_1] & NN_{1,2,3}(Y_1)=[Y_2,X_1,P] \\ NN_{1,2,3}(X_2)=[P,X_3,Y_3] & NN_{1,2,3}(Y_2)=[X_1,P,Y_1] \\ NN_{1,2,3}(X_3)=[Y_3,X_4,Y_4] & NN_{1,2,3}(Y_3)=[X_3,X_4,Y_4] \\ NN_{1,2,3}(X_4)=[Y_4,X_3,Y_3] & NN_{1,2,3}(Y_4)=[X_4,X_3,Y_3] \\ NN_{1,2,3}(P)=[X_1,Y_2,X_2] & \end{array} \quad (7)$$

Then we determine Δn_1^1, Δn_2^1, Δn_1^2, and Δn_2^2, which indicate the change of the number of the nearest neighbors due to the class assignment of *P*.

Solve for Δn_1^1 : Assume $\mathrm{P} \in \mathrm{C}_1$. Then determine the class membership of the 3 nearest neighbors for each sample in $C_1 = \{X_1, X_2, X_3, X_4\}$. For example, for sample, according to (6), X_1 has 3 nearest neighbors, including Y_2, Y_1, X_2. The class membership of Y_2 is C2 (i.e. Class 2), while the class membership of Y_1 and X_2 is C2 and C1, respectively. Thus

we denote the class membership of these three nearest neighbors of X_1 as: *X_1: [C2 C2 C1]*. Therefore, the number of C1 in the nearest neighbors of X_1 is 1, as shown in the label to the right most. In this manner, we count the number of C1 among the nearest neighbors of the remaining samples. The results are shown in the left column. The labels in the circles represent the number of C1.

After that, we consider the 3 nearest neighbors of each sample including the test object, *P*, as shown in (7). For example, for sample, according to (6), X_1 has 3 nearest neighbors, including Y_2, P, Y_1. The class membership of Y_2 is C2 (i.e. Class 2), while the class membership of P and Y_1 is C1 and C2, respectively. Thus we denote the class membership of these three nearest neighbors of X_1 as: *[C2 C1 C2]*. Therefore, the number of C1 in the nearest neighbors of X_1 is 1, as shown in the label to the right most. In this manner, we count the number of C1 among the nearest neighbors of the remaining samples. The results are shown in the left column. The labels in the circles represent the number of C1.

X_1: [C2 C2 C1] 1 ⇒ [C2 C1 C2] 1
X_2: [C1 C2 C1] 2 ⇒ [C1 C1 C2] 2
X_3: [C2 C1 C2] 1 ⇒ [C2 C1 C2] 1
X_4: [C2 C1 C2] 1 ⇒ [C2 C1 C2] 1
# of Class 1

Therefore, Δn_1^1=0. It indicates there is no increase of number of Class 1 among the nearest neighbors by assigning the test object, *P* to Class 1, i.e. C1.

Solve for Δn_2^1: Assume $P \in C_1$. Then count the number of Class 2 among the nearest neighbors of each sample. The labels represent the number of Class 2.

Y_1: [C2 C1 C1] 1 ⇒ [C2 C1 C1] 1
Y_2: [C1 C2 C1] 1 ⇒ [C1 C1 C2] 1
Y_3: [C1 C1 C2] 1 ⇒ [C1 C1 C2] 1
Y_4: [C1 C1 C2] 1 ⇒ [C1 C1 C2] 1
# of Class 2

Therefore, Δn_2^1=0. It indicates there is no decrease of number of Class 2 among the nearest neighbors by assigning the test object, *P* to Class 1, i.e. C1.

Solve for Δn_1^2: Assume $P \in C_2$. Then count the number of Class 1 among the nearest neighbors of each sample. The labels represent the number of Class 1.

X_1: [C2 C2 C1] 1 ⇒ [C2 C2 C2] 0
X_2: [C1 C2 C1] 2 ⇒ [C2 C1 C2] 1
X_3: [C2 C1 C2] 1 ⇒ [C2 C1 C2] 1
X_4: [C2 C1 C2] 1 ⇒ [C2 C1 C2] 1
# of Class 1

Therefore, Δn_1^2=2. It indicates there is 2 decrease of number of Class 1 among the nearest neighbors by assigning the test object, *P* to Class 2, i.e. C2.

Solve for Δn_2^2: Assume $P \in C_2$. Then count Class 2. The label represents the number of Class 2.

Y_1: [C2 C1 C1] 1 ⇒ [C2 C1 C2] 2
Y_2: [C1 C2 C1] 1 ⇒ [C1 C2 C2] 2
Y_3: [C1 C1 C2] 1 ⇒ [C1 C1 C2] 1
Y_4: [C1 C1 C2] 1 ⇒ [C1 C1 C2] 1
# of Class 2

Therefore, Δn_2^2=2. It indicates there is 2 increase of number of Class 2 among the nearest neighbors by assigning the test object, *P* to Class 2, i.e. C2.

As a summary, Δn_1^1=**0** Δn_2^1=**0** Δn_1^2=**2** Δn_2^2=**2,** $k_1 = 2$, $k_2 = 1, n_1 = 4, n_2 = 4$, $M_1 = 0.42$, $M_2 = 0.33$

The classifier will take into account the coherence and validity ratings, and assign the class label associated to the maximum coherence to the unknown test sample.

According to Eq. 5,

$$Classifier = \arg\max_{j \in 1,2} \begin{cases} \dfrac{\Delta n_1^1 + k_1 - kM_1}{(n_1+1)k} - \dfrac{\Delta n_2^1}{n_2 k} \\ \dfrac{\Delta n_2^2 + k_2 - kM_2}{(n_2+1)k} - \dfrac{\Delta n_1^2}{n_1 k} \end{cases}$$

$$= \begin{cases} \dfrac{0+2-3\times 0.42}{(4+1)\times 3} - \dfrac{0}{4\times 3} \\ \dfrac{2+1-3\times 0.33}{(4+1)\times 3} - \dfrac{2}{4\times 3} \end{cases}$$

$$= \begin{cases} 1.916 \\ -0.036 \end{cases}$$

Therefore, we assign *P* to Class 1.

IV. Experimental Results

The wine data consists of three types of wines grown in the same area in Italy but derived from three different cultivars (https://archive.ics.uci.edu/ml/datasets/wine). The analysis provides the quantities of 13 active constituents found in each of the three types of wines.

First we compare our algorithm to the conventional KNN and the modified KNN [7] on the wine data. We increase the *k* value from 3 to 7, and observe the classification accuracy of the three algorithms. The result is shown in Fig. 2. The left blue column represents the conventional KNN, the middle green column represents the MKNN, and the right yellow column represents our algorithm. The result shows that our algorithm outperforms both of the conventional KNN and the modified KNN.

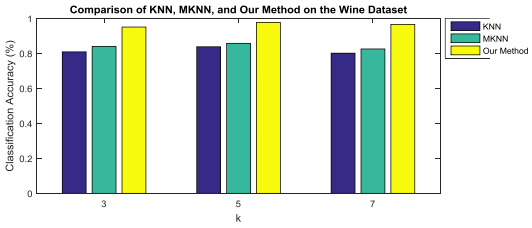

Fig. 2. Comparison of the conventional KNN method, modified KNN method, and our algorithm on the wine data in terms of different *k* values.

We then compare the classification accuracy of our algorithm to the ENN algorithm with the increasing number of k. The k value increases from 1 to 20. The result is shown in Fig. 3. The red curve represents the conventional KNN, and the blue curve represents our algorithm. It shows that our algorithm has higher classification accuracy than the conventional KNN over the k values. In addition, our algorithm keeps relatively high classification accuracy between k = 6 and k = 16, with the highest value occurred at k=11. At k = 11, the classification accuracy of our algorithm is 98.33%. Moreover, while the classification accuracy of the conventional KNN algorithm has changed drastically, our algorithm remains relatively constant over the k values. Furthermore, the conventional KNN algorithm has shown a declining trend when k = 20, while our algorithm remains stable.

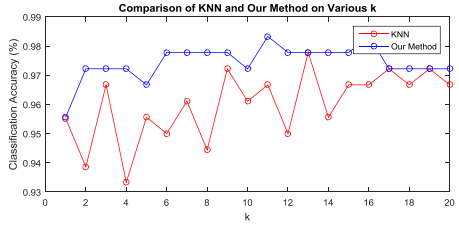

Fig. 3. Comparison of KNN method and our algorithm on various k values from 1 to 20.

We further study the classification performance of the proposed algorithm on the wine data in terms of different combination of k values and k-fold values. We use k-fold cross-validation algorithm instead of using the conventional validation algorithm (i.e. dividing the dataset into three sets of 70% for training, 15% for validation, and 15% for testing) because the dataset has small sample size which results in insufficient data to be partitioned into separate training, validation, and testing sets without losing significant modeling competence. We also compare the result with the conventional KNN algorithm. The classification accuracy of KNN and our algorithm are shown in Fig. 4 and Fig. 5, respectively. X-axis represents the number of folds, y-axis represents the number of k value, and the z-axis represents the classification accuracy. Each ribbon corresponds to the classification accuracy at a specific fold value. There are totally 19 folds (i.e. 2nd – 20th fold), so there are 19 ribbons. In addition, on each ribbon, we can observe the classification accuracy on various k values (i.e. k = 1 to 20). From Fig. 4, we can observe that the KNN algorithm has the highest classification accuracy (97.78%) when k =13 and fold = 10 for all folds. From Fig. 5, we find that when k = 14 and fold = 5, our algorithm reaches the peak of classification accuracy (98.89%) and remain at peak values when k increases.

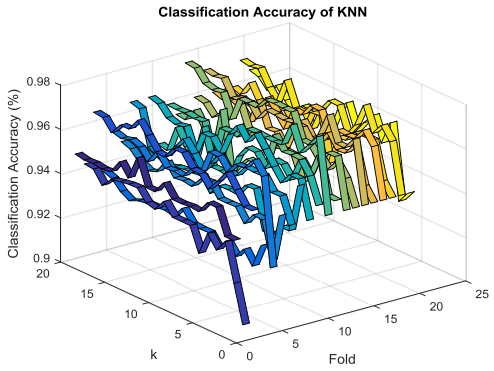

Fig. 4. Classification accuracy of KNN method in terms of various combination of k values and fold values.

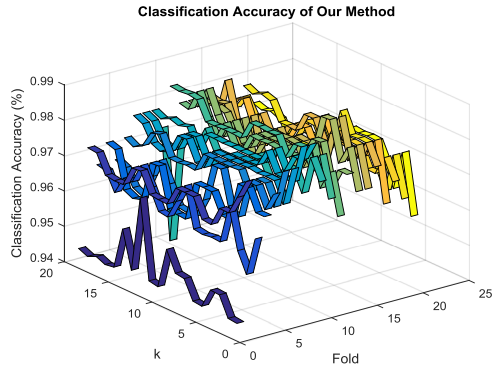

Fig. 5 Classification accuracy of our method in terms of various combination of k values and fold values.

V. Conclusions

This paper proposes an improved KNN algorithm to overcome the *class overlapping* problem when the class distribution is skewed. Different from the conventional KNN algorithm, it not only finds out the k nearest neighbors of each sample (even the test object itself) in the training dataset, but also the neighbors of the unknown test object. Then the validity value of a data point is computed based on the label of the data and the labels of its k nearest neighbors. A classifier is designed to assign the unknown test object to a class membership based on the proposed validity ratings equations. We conducted three sets of experiments to compare our algorithm to the conventional KNN and the modified KNN algorithm on different combination of *k* values and *fold* values. The experimental results demonstrate that our algorithm has significantly higher classification accuracy than the conventional KNN and the modified KNN algorithm on the wine dataset.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) HRD #1505509, HRD #1533479, and DUE #1654474.

REFERENCES

[1] A. Lapointe, "Investigations of novel sensor technology for explosive specific detection," http://www.dtic.mil/dtic/tr/fulltext/u2/a539685.pdf 2009.

[2] C. Kendziora, A. Furstenberg, R. Papantonakis, V. Nguyen, J. Borchert, J. Byers, and R. A. McGill, "Infrared photothermal imaging of trace explosives on relevant substrates," Proceedings of SPIE, vol. 8709, 2013.

[3] N. Zhang and L. A. Thompson, "An Intelligent clustering algorithm for high dimensional and highly overlapped photo-thermal infrared imaging data," Fall 2016 ASEE Mid-Atlantic Regional Conference, Hempstead, New York, October 21-22 ,2016.

[4] E. Fix and J. L. Hodges, "Discriminatory analysis, nonparametric discrimination, consistency properties," Jr. International Statistical Review. vol. 57, no. 3, pp. 238-247, 1989.

[5] J. Hou, H. Gao, Q. Xia, N. Qi, "Feature combination and the KNN framework in object classification," IEEE Transactions on Neural Networks and Learning Systems. 2016, vol. 27, no. 6, pp. 1368-1378, 2016.

[6] B. Tang and H. He, "ENN: Extended nearest neighbor algorithm for pattern recognition," IEEE Computational Intelligence Magazine. vol.10, no.3, pp.52-60, 2015.

[7] H. Parvin, H. Alizadeh, and B. Minaei-Bidgoli, "MKNN: Modified k-nearest neighbor," Proceedings of the World Congress on Engineering and Computer Science (WCECS), San Francisco, USA, 2008.

[8] S. Taneja, C. Gupta, S. Aggarwal, V. Jindal, "MFZ-KNN – A Modified fuzzy based k nearest neighbor algorithm," 2015 International Conference on Cognitive Computing and Information Processing (CCIP), Noida, pp. 1-5, 2015.

[9] N. Zhang, J. Xiong, J. Zhong, L. Thompson, and H. Ying, "An enhanced k-nearest neighbor classification method based on maximal coherence and validity ratings," The 14th International Symposium on Neural Networks (ISNN), Sapporo, Hokkaido, Japan, June 21-26, 2017.

[10] H. Xiong, M. Li, T. Jiang, and S. Zhao, "Classification algorithm based on NB for class overlapping problem," Appl. Math. Inf. Sci. 7, no. 2L, 409-415, 2013.

[11] J. F. Ramirez Rochac, N. Zhang, and P. Behera, "Design of adaptive feature extraction algorithm based on fuzzy classifier in hyperspectral imagery classification for big data analysis," The 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 2016.

[12] N. Zhang, "Cost-Sensitive spectral clustering for photo-thermal infrared imaging data," Sixth International Conference on Information Science and Technology (ICIST), Dalian, China, 2016.

[13] J. F. Ramirez Rochac and N. Zhang, "Reference clusters based feature extraction approach for mixed spectral signatures with dimensionality disparity," The 10th Annual IEEE International Systems Conference (IEEE SysCon), Orlando, Florida, 2016.

[14] J. F. Ramirez Rochac and N. Zhang, "Feature extraction in hyperspectral imaging using adaptive feature selection approach," The Eighth International Conference on Advanced Computational Intelligence (ICACI2016), Chiang Mai, Thailand, pp. 36-40, 2016.