Improved KNN with Feedback Support

Shubham Mishra
Mumbai University
India

Harshali Patil
Mumbai University
India

ABSTRACT
This paper shows, a new method has been introduce to enhancing the performance of K-Nearest Neighbor is implemented which uses K neighbors for classifying the new data. This new classification method is called Improved K-Nearest Neighbor, IKNN. Inspired the traditional KNN algorithm, the main idea is to provide feedback that is for next iteration it should also consider previous classifications. Other than providing the feedback it also modified its distance calculating formula. In this method a weight vector for class labels vector is initialized. For each iteration this weight vector matrix will play major role for data classification. Experiments show the improvement in the accuracy of the IKNN algorithm.

Keywords
IKNN, KNN Classification, Improved K-Nearest Neighbor, Feedback.

1. INTRODUCTION
In classification the task is to assigning labels to objects which are described by a set of measurements also called as attribute. Current research builds upon foundations laid out in the 1960s and 1970s. Because classification is faced challenges of solving real-life problems, in spite of decades of productive research, graceful modern theories still coexist with independent ideas, intuition and guessing [1] [4].

There are two major types of pattern recognition problems: unsupervised and supervised. In the supervised category which is also called supervised learning or classification, each object in the data set comes with a reassigned class label. The prime task is to train a classifier to do the labeling, sensibly. Mostly the labeling process cannot be implemented in an algorithmic form, so it supply the machine with learning skills and present the class labeled data to it. The Classification knowledge learned by the machine in the mentioned process might be not perfect, but the recognition accuracy of the classifier will be the judge of its adequacy. The new classification systems KNN try to classify data based on the trained datasets [2-5]. There are many more classifications algorithms like SVM, Naïve Bays, Regression algorithms but KNN is the one algorithm which does fast calculation[6] [8-9]. K-Nearest Neighbor (KNN) classification is one of the most effective, fast and simple classification methods. When there is little prior knowledge about the distribution of the data, the KNN method should be the prior choice for data classification. It is a powerful classification system which ignores the problem of probability densities completely [7] [16] [19]. The KNN rule gives class label which having majority label from the K neighbors; this means that, the decision is taken by analyzing the labels on the K-nearest neighbors and taking the class with majority. KNN algorithm Classification was developed from the need to perform discriminate analysis when reliable parametric estimates of Probability densities are unknown or difficult to determine. In 1951, Fix and Hodges introduced a non-parametric method for pattern classification that has since become known the K-nearest neighbor rule [8] [14-16] [21]. Later in 1967, some of the formal properties of the K-nearest neighbor rule have been worked out; for instance it was shown that for K=1 and $n \to \infty$ the KNN classification error is bounded above by twice the error rate [9]. Once such formal properties of KNN classification were established, a long line of investigation ensued including new rejection approaches [10].

2. METHODOLOGY

2.1 Feedback
In the KNN algorithm once it is trained on some data set then it predict output based on this trained data only. It does not take help of tested classified data. In IKNN algorithm while classifying the new classified data is also considered for future classification. As more and more algorithm will be trained more it will become accurate. In traditional KNN algorithm while classification only the trained data sets are considered while in improved KNN with the help of weight vector new classified data are also considered for the classification purpose.

2.2 Python
Python is a widely used high-level programming language. Python is a popular language for the time among computers world. It has been created by Guido van Rossum and the first released was back in 1991. It is more popular due to its ease of syntax and a big and growing community. The Python is now in use in the entire domain whether it would be research or game development. Due to its major impact and efficiency, it is the prime choice for research kind of projects [11] [12]. The language provides good indentation to enable writing correct programs on both a small and large scale.

3. EXPERIMENT
In this KNN algorithm there are major two changes, one the distance calculating formula and introduced feedback in the form of weight vector.

In KNN algorithm at the time of classification the neighbor class with majority wins and new data is assigned label of that class.

```
for i in range(k):
    voteIlabel = labels[sortedDistIndicies[i]]
    classCount[voteIlabel] =
    classCount.get (voteIlabel,0) + 1
sortedClassCount = sorted (classCount.iteritems ( ),
    key=operator.itemgetter(1), reverse=True)
return sortedClassCount[0][0]
```

In above lines of code, for new element all K nearest classes voted. The class with maximum class Count will be the class for new element. But in above algorithm one do not take care

of previous output which reduces the accuracy of the algorithm by ignoring previous outputs.

```
weights=zeros(NoofClass)
for j in range(NoofClass):
    weights[j]=1
```

Feedback on the other hand will help the algorithm to predict best by also considering previous outputs. In modified algorithm weight vector is initialized with elements as the weight of each output class. It has been initialized by 1.

```
for i in range(k):
    voteIlabel = labels[sortedDistIndicies[i]]
    classCount[voteIlabel] = classCount.get(voteIlabel,0) + weights[voteIlabel]
sortedClassCount = sorted( classCount.iteritems(), key=operator.itemgetter(1), reverse=True)
weights[sortedClassCount[0][0]]=weights[sortedClassCount[0][0]]+1
return sortedClassCount[0][0]
```

In above lines of code, the IKNN classification method, it considering all K nearest classes with vote increment by the weight matrix value of that class. After classification at the second last line the weight value of the winning class is incremented. This procedure helps the algorithm to predict new element class by considering classes all last election result.

In figure 1, the new element in red dots will be having three neighbors. According to KNN it should belong to Blue class, but that is not always correct since overall majority is for Black class. In KNN one has to consider K value for selecting total number of neighbors some ease of calculation and hence some important data may be missed out. Due to small K value it generate under fitting problem.

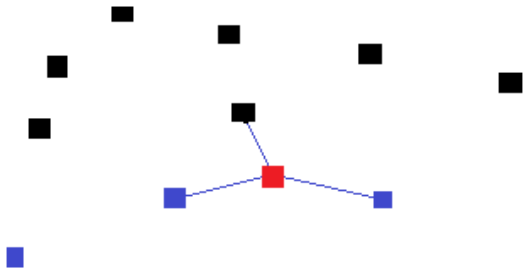

Figure 1. Data Sample

In IKNN the effect of under fitting is reduced by some extend. In IKNN based on weight matrix it will classify the new element.

In KNN algorithm for distance calculation between new element and K nearest neighbor's elements Euclidean distance formula,

$$d(p,q) = d(q,p)$$

$$= \sqrt{(q1-p1)^2 + (q2-p2)^2 + \cdots + (qn-pn)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (qi-pi)^2}$$

Figure 2. Euclidean Distances

But, to making distance more clear the square power is replaced with higher value. By experimenting it has been found that minimum of power 4 is sufficient. By increasing the power value it leads to over fitting which decreases the efficiency. Hence new formula will be,

$$= \sqrt{\sum_{i=1}^{n} (qi-pi)^4}$$

Figure 3. Modified Distances

4. RESULT

Below is the screenshot shows error rate for standard KNN algorithm. It is the last part of the outcome which shows left part as what classifier has predicted and the right part as what was correct output.

```
the classifier came back with: 2, the real answer is: 1
the classifier came back with: 2, the real answer is: 2
the classifier came back with: 1, the real answer is: 1
the classifier came back with: 1, the real answer is: 1
the classifier came back with: 2, the real answer is: 2
the total error rate is: 0.080000
```

Figure 4. KNN Outcome

In above part of the outcome out of 500 tests it has passed 460 correctly. Now for modified KNN algorithm,

```
the classifier came back with: 2, the real answer is: 1
the classifier came back with: 2, the real answer is: 2
the classifier came back with: 1, the real answer is: 1
the classifier came back with: 1, the real answer is: 1
the classifier came back with: 2, the real answer is: 2
the total error rate is: 0.058000
```

Figure 5. IKNN Outcome

In above outcome out of 500 tests it has passed 471 correctly. Hence its efficiency is increased up from 92.00% to 94.20%.

5. CONCLUSION

In this paper an algorithm is modified to improve its efficiency. The concept feedback has been introduced and distance formula has been modified. The IKNN algorithm now makes use of its previous experience while taking decisions. This modified algorithm is implemented in python language and its error rate shown. The distance formula is modified for helping algorithm a clear step for classifying data. IKNN algorithm will become more accurate if it will be trained more. IKNN is an improved algorithm.

6. REFERENCES

[1] L. I. Kuncheva, Combining Pattern Classifiers, Methods and Algorithms, New York: Wiley, 2005.

[2] H. Parvin, H. Alizadeh, B. Minaei-Bidgoli and M. Analoui, "An Scalable Method for Improving the Performance of Classifiers in Multiclass Applications by Pairwise Classifiers and GA", In Proc. Of the Int. Conf.

on Networked Computing and advanced Information Management by IEEE Computer Society, NCM 2008, Korea, Sep.2008, (in press).

[3] H. Parvin, H. Alizadeh, M. Moshki, B. Minaei-Bidgoli and N. Mozayani, "Divide & Conquer Classification and Optimization b Genetic Algorithm", In Proc. of the Int. Conf. on Convergence and hybrid Information Technology by IEEE Computer Society, ICCIT08, Nov. 11-13, 2008, Busan, Korea (in press).

[4] H. Parvin, H. Alizadeh, B. Minaei-Bidgoli and M. Analoui, "CCHR: Combination of Classifiers using Heuristic Retraining", In Proc. of the Int. Conf. on Networked Computing and advanced Information Management by IEEE Computer Society, NCM 2008, Korea, Sep. 2008, (in press).

[5] H. Parvin, H. Alizadeh and B. Minaei-Bidgoli, "A New Approach to Improve the Vote-Based Classifier Selection", In Proc. of the Int. Conf. on Networked Computing and advanced Information Management by IEEE Computer Society, NCM 2008, Korea, Sep. 2008, (in press).

[6] H. Alizadeh, M. Mohammad and B. Minaei-Bidgoli, "Neural Network Ensembles using Clustering Ensemble and Genetic Algorithm", In Proc. of the Int. Conf. on Convergence and hybrid Information Technology by IEEE Computer Society, ICCIT08, Nov. 11-13, 2008, Busan, Korea (in press).

[7] B.V. Daresay, Nearest Neighbor pattern classification techniques for removing probability density problem, Las Alamitos, LA: IEEE Computer Society Press.

[8] Fix, E., Hodges, J.L. Discriminatory analysis, SVM algorithm nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[9] Cover, T.M., Hart, P.E. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory, IT-13(1):21–27, 1967.

[10] Hellman, M.E. The nearest neighbor classification rule with a reject option. IEEE Trans. Syst. Man Cybern., 3:179–185, 1970.

[11] McConnell, Steve (30 November 2009). *Code Complete, p. 100*. ISBN 9780735636972.

[12] Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises".

[13] Bailey, T., Jain, A. A note on distance-weighted k-nearest neighbor rules. IEEE Trans. Systems, Man, Cybernetics, Vol. 8, pp. 311-313, 1978.

[14] Bermejo, S. Cabestany, J. Adaptive soft k-nearest-neighbor classifiers. Pattern Recognition, Vol. 33, pp. 1999-2005, 2000.

[15] Jozwik, A. A learning scheme for a fuzzy k-nn rule. Pattern Recognition Letters, 1:287–289, 1983.

[16] Keller, J.M., Gray, M.R., Givens, J.A. A fuzzy knn neighbor algorithm. IEEE Trans. Syst. Man Cyber., SMC-15(4):580–585.

[17] ITQON, K. Shunichi and I. Satoru, Improving Performance of k-Nearest Neighbor Classifier by Test Features, Springer Transactions of the Institute of Electronics, Information and Communication Engineers 2001.

[18] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification , John Wiley & Sons, 2000.

[19] E. Gose, R. Johnson bough and S. Jots, Pattern Recognition and Image Analysis, Prentice Hall, Inc., Upper Saddle River, NJ 07458, 1996.

[20] Blake, C.L.Merz, C.J. UCI Repository of machine learning databases http://www.ics.uci.edu/~mlearn/ MLRepository.html, (1998).

[21] S. Aeberhard, D. Coomans and O. de Val, "Comparison of Classifiers in High Dimensional Settings", Tech. Rep. no. 92-02, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland.