# An Improved k-NN Classification with Dynamic k

Xiao-Feng Zhong[1], Shi-Ze Guo[2], Liang Gao[2], Hong Shan[1], Jing-Hua Zheng[1]
[1] Electronic Engineering Institute, Hefei, 230031, China
[2] Institute of North Electronic Equipment, Beijing 100015, China
eeijunre@126.com, gaolleasy@163.com

## ABSTRACT

In the k-NN algorithm, k is the only parameter and often set to a fixed value empirically. However, it is very difficult to choose an appropriate k in practice, and if the choice is not appropriate, the performance of k-NN will be affected greatly. In order to solve this problem, the paper proposes an improved k-NN algorithm, which is denoted as Dk-NN, by using dynamic k in replace of fixed k value. Firstly, a preprocessed step is designed and added to the traditional k-NN algorithm for determining the dynamic k interval. Then, each class's percentage of test sample is calculated within the dynamic k interval. Finally, three criterions are given to determine the class of the test sample according to the variation tendency of the percentage curves. Experimental results on real-world dataset demonstrate that the proposed algorithm is more effective than the k-NN algorithm with fixed k value.

## CCS Concepts

• **Theory of computation → Theory and algorithms for application domains → Machine learning theory → Models of learning.**

## Keywords

k-Nearest neighbors; dynamic K; classification algorithm.

## 1. INTRODUCTION

The k-Nearest Neighbors (k-NN) algorithm, which is the most common classification algorithm with the features of simple structure, high accuracy, no sensitive to outliers and no data input hypothesis, is widely used in pattern recognition, text classification, and even radicalization detection [1]-[4].

In the k-NN algorithm, the k value is the only parameter and is often set to a fixed value. But it does not give the method how to determine the k value in practice. On the one side, the k value is wished to be large in order to minimize the probability of a non-Bayes decision. On the other side, it is also wished to be small in order to give an accurate estimate of the posterior probabilities of the true class [5]. So the selection of k value would have a great influence on the performance of k-NN, and how to find an appropriate k value would be a big challenge.

To address this problem, an improved k-NN algorithm, named as Dk-NN, is proposed in this paper. In this algorithm, k is designed as dynamic value in replace of fixed one and the classification result is determined based on the variation tendency of their k-nearest Neighbors, which are calculated in the dynamic value. Furthermore, three criteria are given as the basis for the identification of the test samples. The experimental results on real datasets show that the proposed Dk-NN algorithm, which fully considers the influence of k value, can get more accurate result than the traditional k-NN algorithm does.

*ICMLC 2017*, February 24-26, 2017, Singapore, Singapore
© 2017 ACM. ISBN 978-1-4503-4817-1/17/02…$15.00
DOI: http://dx.doi.org/10.1145/3055635.3056604

## 2. Related Works

As a type of instance-based learning or lazy learning, k-NN algorithm is firstly proposed by T. M. Cover and P. E. Hart for the classification problem [4]. For its high performance in real applications and the nonparametric setting [6], the k-NN was regarded as one of top 10 algorithms in data mining.

However, the k-NN algorithm has its inherent disadvantages [7], such as sheer complexity of time and space with too large training sample set or too many feature items, depending too much on the k value selection.

To overcome these disadvantages, numerous investigations have been done to improve the k-NN algorithm. All the existing solutions can be usually classified in one of the following categories:
(1) Managing feature space or sample datasets;

(2) Improving parameter k selection method.

Feature selection by removing irrelevant or redundant features, is ended to data dimension reduction and increment of classification accuracy [8]. In [8] and [9], a comprehensive review of feature selection methods has been done. These feature selection methods include forward and backward sequential selection [10], artificial bee colony (ABC) [11], particle swarm optimization (PSO) [12], biogeography based optimization (BBO) [13], non-negative matrix factorization (NMF) [14] and self-organizing feature maps (SOM) [15].

Some researchers are endeavoring to reduce the dimension of dataset for improving of the accuracy of the k-NN classification. These improvements include dataset shifting [16], dimension reduction [17]-[18] and subset selection [19]. Some other researchers focus on determining the weighted samples and features [20]-[22]. [23]-[25] give the answer on how to deal with unbalance data classification.

The fixed k value ignores the influence of the category and the document number of training text. Reference [7] analyses the influence of selecting different k value, and presents a dynamic k classifier with similarity calculation for improving classification accuracy. An approximate method can be found in [26]. Reference [27] reconstructs test samples with training samples to obtain the optimal k value for each test sample, and proposes a non-parametric test point specific k estimation strategy. In [28], an efficient method for selecting k value based on cross-validation strategies is demonstrated. All these works only consider the dynamic characteristics of parameter k, but set it as a fixed value.

In this paper, we take a full consideration of the dynamic characteristics of parameter k, and set k value within an interval. Furthermore, we present an algorithm how to classify samples with this interval. Reference [29] discussed this interval, but only chose a random k value within this interval as a parameter of k-NN iteration algorithm.

## 3. THE IMPROVED K-NN CLASSIFICATION ALGORITHM

Given a training dataset $D_{train}$ :

$$D_{train} = \{(x,l): x \in R^M, l \in L\} \qquad (1)$$

where $x$ is a M-dimensional feature vector, $l$ is the objects' label and $L$ is the set of all labels.

For an unlabeled sample $y$, the traditional *k*-NN algorithm firstly calculates the distances of each pair $(x_i, y)$, and finds its *k*-nearest neighbors $\{x_1, x_2, \cdots, x_k\}$ , the class labels of which are $\{l_1, l_2, \cdots, l_k\}$ respectively. And then, the algorithm uses the implest majority vote approach to determine the class label of sample $y$ and the classification formulation can be denoted as follows [30]:

$$l_{k-NN}(y) = \arg\underset{l\in L}{Max}\sum_{i=1}^{k}\delta(l,l_i) \qquad (2)$$

We can see that there is only one parameter k in formula 2 and this parameter is often selected empirically. Generally, the value of *k* is fixed to 1, 3, 5, 7 [31] or $\lceil\sqrt{\text{N}}\rceil$ [32]-[33], where N denotes the cardinality of the training dataset. However, the selection of the fixed k value can cause some errors and Figure 1 gives an example for these problems. In Figure 1, the size of training dataset is set to 600 and the test sample is selected as a negative one. Obviously, when $k = \lceil\sqrt{600}\rceil$ , formula 2 give the right label of the test sample, but when $k$=5, the result is wrong because there are four outliers around the test sample. That is to say, the fixed value of k with experience is impracticable, and the k value will directly influence the precision and accuracy of the classification, even lead to wrong classification result.

In order to solve this problem, the paper proposed a novel dynamic k method. The proposed method includes an interval of parameter k, which denoted as $[k_{\min}, k_{\max}]$ , and a matrix $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \cdots, \mathbf{T}_{len(L)}]'$ , where $\mathbf{T}_j$ is a row to store the percentage of the sample that belong to the $j$ class when $k \in [k_{\min}, k_{\max}]$. Make $t_{jk} \in \mathbf{T}_j$ , and $t_{jk}$ can be expressed as follows:

$$t_{jk} = \frac{1}{k}\sum_{i=1}^{k}\delta(l_j, l_i) \quad k \in \mathbf{Z} \text{ and } k \in [k_{\min}, k_{\max}] \qquad (3)$$

For every test example, matrix $\mathbf{T}$ gives the variation tendency curves of each class within interval $[k_{\min}, k_{\max}]$. There are three different situations in the real case. Figure1 shows the first one, and Figure 2 gives the variation tendency curves of the test sample which is shown in figure1.

In Figure 2, the percentages of negative label are lower than those of positive label when $k \in [1,8]$. The reason of this phenomenon is that there are outliers very close to the test sample. But when k>8, the percentage of the two labels are reversed and with the increase of k value, the percentage of negative label tend to be close to 1, while the percentage of positive label tend to be close to 0. Consequently, analyzing the variation tendency curves with matrix **T** can avoid the influence of outliers and give the right answer. Criterion 1 gives the judge basis in this case.

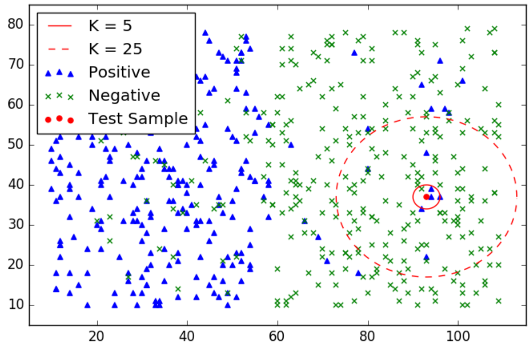


**Figure 1. An illustration on a binary classification for the selection of fixed k value with k = 5 and k =25.**

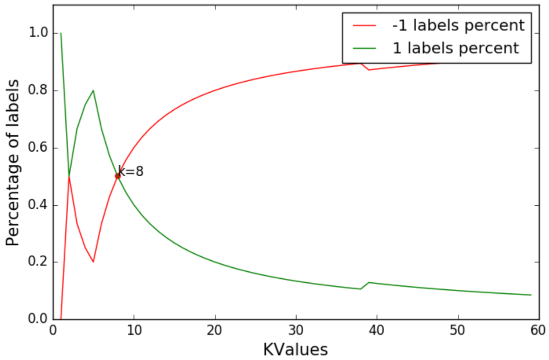


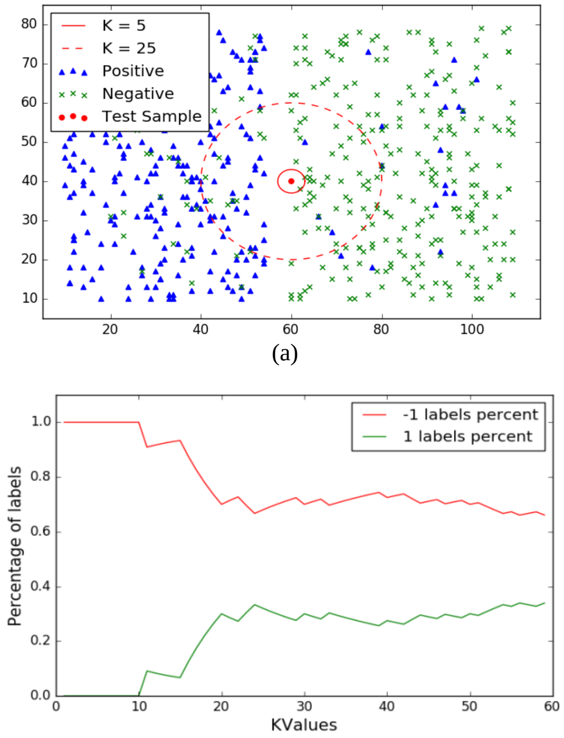
**Figure 2. Variation tendency curves of test sample in Figure 1.**

Criterion 1: If there are a few intersections in the end of the interval $[k_{\min}, k_{\max}]$ , but in the other parts, there are no intersections and the percentage difference is obvious, the label of the sample should be set to the class which has the highest percentage in the intermediate part of the interval $[k_{\min}, k_{\max}]$.

In addition, there are two kinds of datasets and curves in practice, which are shown in Figure 3 and Figure 4 respectively. Figure 3(a) shows that there are few outliers around the test sample and the corresponding tendency curves are given in Figure 3 (b). We can see that the two curves have no intersection. In this situation, the test sample's class can be determined by criterion 2.

Criterion 2: If there is no intersections within $[k_{\min}, k_{\max}]$. The label of the test sample can be set as the class which has highest percentage in the interval $[k_{\min}, k_{\max}]$.

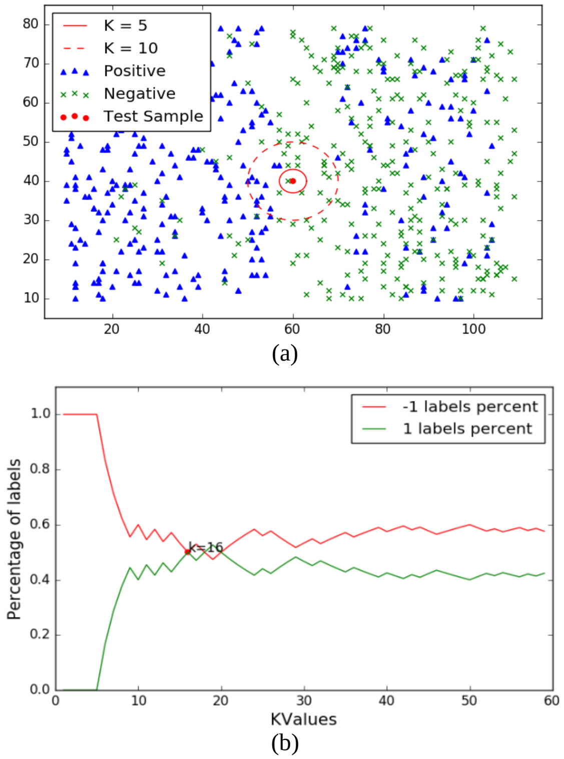
Figure 4(a) shows another situation, in which there are a few outliers around the test sample. Figure 4(b) gives the corresponding tendency curves. At the beginning, two curves have a great distance. But with the increase of k value, the outliers are included in, so the distance between two curves decrease gradually. When k=16, the two curves intersect and then change gently. In this situation, the outliers don't have a large influence on classification results and the test sample can be classified as follows:



(a)

(b)

**Figure 3. The second real situation. (a) The second real situation with 600 samples and one test sample. (b) The second real situation with variation tendency curves.**



(a)

(b)

**Figure 4. The third real situation. (a) The third real situation with 600 samples and one test sample. (b) The third real situation with variation tendency curves.**

Criterion 3: If the percentage difference in the left-inclusive interval $[k_{\min}, k_{\max}]$ is obvious, but in other parts, this difference is very small, the label of the test sample should be set to the class which has highest percentage within the left-inclusive interval $[k_{\min}, k_{\max}]$.

It can be seen from the above, the interval $[k_{\min}, k_{\max}]$ should be figured out before these criterions are carried out. Since k-NN is a typical example of lazy learning and simply stores training instances at training time and delays its learning process until classification time [30], a preprocessed step should be added to the traditional k-NN algorithm at first.

For the purpose of setting the interval $[k_{\min}, k_{\max}]$, an iterative algorithm is given in Algorithm 1. In each iterative, the training dataset and the testing dataset are selected randomly. And then, the intersections of each testing sample are calculated. After the iteration, the frequency of the intersections are counted up and the interval $[k_{\min}, k_{\max}]$ is fixed according to the frequency of the intersections with a frequency threshold.

**Algorithm 1: interval selection**

```
Input sample dataset: $D = \{(x,l) : x \in R^M, l \in L\}$, Iterative times $N$, threshold $h$, percentage of random selection $p$, kList=NULL, fList=NULL
Output: interval: $[k_{\min}, k_{\max}]$
For each iterative do
    Select $p$ of samples randomly as training dataset $D_{train}$, testing dataset $D_{test}$ contains The remaining samples.
    Calculate matrix $\mathbf{T}$ of each sample in $D_{test}$ with samples in $D_{train}$.
    for each $\mathbf{T}$ do
        Calculate intersections, get k value list $k_{x_i}$ and frequency list $f_{x_i}$ of these intersection.
        Combine $k_{x_i}$ to kList and $f_{x_i}$ to fList.
According to $h$, get list $\{k_i, k_{i+1}, \cdots, k_j\}$ by removing the k value whose frequency is beyond $h$.
set $k_{\min} = \min\{k_i, k_2, \cdots, k_j\}$
    $k_{\max} = \min\{k_i, k_2, \cdots, k_j\}$
Output $[k_{\min}, k_{\max}]$
```

# 4. EXPERIMENTS

## 4.1 Dataset Description

The dataset used in the experiments are the Facebook dataset from the myPersonality project [34]. The myPersonality is a popular Facebook application in which users take a standard BIG5 Factor Model psychometric questionnaire and give consent to record their responses and Facebook profiles. For a given Facebook user's profile, the experiments use Dk-NN to predict the user's BIG5 label.

The original personality trait score is in the range of 1 to 5. In order to get classify label, this interval can be transformed into the following classes: low, medium and high [35]. Considering the distribution of personality score of all users, for each personality trait, the low class is assigned to those scores that are below the 33th percentile, the medium class is assigned to those scores that are between the 33th and 66th percentiles, and the high class is assigned to those scores that are exceed 66th.

In this paper, we mainly focus on the prediction of the high and the low class users. As a result, 10,718 high score samples and 9,422 low score samples are selected from the Facebook dataset. Table I shows the description about these samples. Two thirds of them are chosen as the training samples randomly, and the left is the testing samples.

**Table 1. Experiment Facebook datasets.**

| Dataset | Label | Number of Profiles | Number of Samples |
|---|---|---|---|
| High Score Dataset | +1 | 64 | 10,718 |
| Low Score Dataset | -1 | 64 | 9,422 |

## 4.2 Process of Experiments

First of all, the interval $[k_{\min}, k_{\max}]$ is fixed according to Algorithm 1. Each test sample's matrix **T** is calculated using Formula 3 and the frequency of k value is showed in figure 5. According to the frequency threshold, the k value, whose frequency is below 10, is removed from the k value list. Thus, the interval $[k_{\min}, k_{\max}]$ is fixed as [2, 232].

Secondly, based on the matrix of each test sample is calculated. And then, the class of the test sample is determined by analyzing the variation tendency of the percentage curves according to three criterions. As a result, 5921 testing samples are labeled correctly by Dk-NN algorithm. Among these right labeled samples, about 51 percent samples are labeled by criterion 2. This indicates that nearly half part of Facebook users have obvious difference with the samples which have different label. About 27 percent and 22 percent samples are classified by Criterion 1 and 3 respectively. The result shows that the Facebook testing dataset contains outliers.

Finally, the classification accuracy of the Dk-NN algorithm with dynamic k is compared with that of the traditional k-NN algorithm with k=1, 3, 5, 7 and $\lceil \sqrt{\mathrm{N}} \rceil$. Table 2 gives the comparison results.

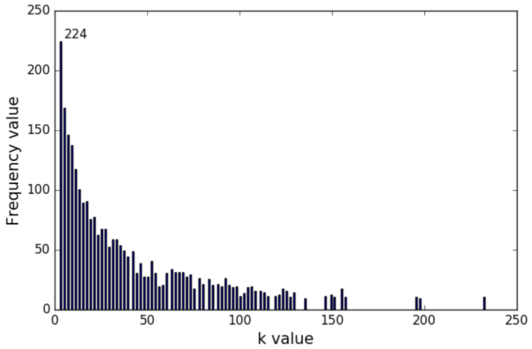


**Figure 5. The frequency of k value list.**

It can be see that the Dk-NN algorithm has outperformed the traditional k-NN algorithm. The classification precision of the Dk-NN is equivalent to the k-NN with fixed k=130, but it is higher than the other k-NN algorithm. In the recall and F value, the proposed algorithm has much improved performance compared to the other k-NN algorithms.

**Table 2. Experiments result of Dk-NN and k-NN.**

| | k value type | k value | precision | recall | F |
|---|---|---|---|---|---|
| k-NN | Fixed | 1 | 0.7467 | 0.7196 | 0.7329 |
| k-NN | Fixed | 3 | 0.8328 | 0.7601 | 0.7948 |
| k-NN | Fixed | 5 | 0.8560 | 0.7670 | 0.8091 |
| k-NN | Fixed | 130 | **0.9081** | 0.7046 | 0.7935 |
| Dk-NN | dynamic | [2,232] | 0.8911 | **0.7800** | **0.8319** |

# 5. CONCLUSIONS

This paper proposes an improved k-NN algorithm, which is called as Dk-NN. In this algorithm, the parameter k, which is fixed in the traditional k-NN, is designed as a dynamic value. Furthermore, an algorithm is presented to determine the interval of the dynamic k value. Experiments on real dataset demonstrate that our proposed algorithm has better classification accuracy in comparison with the traditional k-NN algorithm.

In the future, how to cope with imbalance samples and multi-label classification problems will be studied.

# 6. ACKNOWLEDGMENT

Our thanks to JIAN-SHAN SUN for generously providing Facebook dataset for the present study. Thanks also to mypersonality.org for collecting a wide variety of Facebook users' profiles and their personality.

# 7. REFERENCES

[1] Taneja, S., Gupta, C., Aggarwal, S., and Jindal, V. 2015. MFZ-KNN-A modified fuzzy based K nearest neighbor algorithm. *International Conference on Cognitive Computing and Information Processing*. (April, 2015), 1-5. DOI= http://doi.acm.org/10.1109/CCIP.2015.7100689

[2] Dhurandhar, A., and Dobra, A. 2013. Probabilistic characterization of nearest neighbor classifier. *Int. J. Mach. Learn. & Cyber.* 4, 4 (August, 2013), 259-272. DOI= http://doi.acm.org/10.1007/s13042-012-0091-y

[3] Khadim, D., Fleur, M., and Gayo, D. 2016. Large scale biomedical texts classification: a knn and an esa-based approaches. *J Biomed Semant. 7*, 1 (December, 2016), 1-12. DOI= http://doi.acm.org/10.1186/s13326-016-0073-1

[4] Agarwal, S., and Sureka, A. 2015. Using KNN and SVM Based One-Class Classifier for Detecting Online Radicalization on Twitter. *11th International Conference, ICDCIT.* (February, 2015), 431-442. DOI= http://doi.acm.org/10.1007/978-3-319-14977-6_47

[5] Cover, T. M., and Hart, P. E. 1967. Nearest neighbor pattern classification. *IEEE Trans Inf Theory. 13*, 1, (1967), 21-27. DOI= http://doi.acm.org/10.1109/TIT.1967.1053964

[6] Mary-Huard, T., and Robin, S. 2009. Tailored aggregation for classification. *IEEE Trans Pattern Anal Mach Intell. 31*, 11 (March, 2009), 2098-105. DOI= http://doi.acm.org/10.1109/TPAMI.2009.55

[7] Gong, A., and Liu, Y. 2011. Improved KNN Classification Algorithm by Dynamic Obtaining K. *International*

*Conference, ECWAC.* (April, 2011), 320-324. DOI= http://doi.acm.org/10.1007/978-3-642-20367-1_51

[8] Lin, S. W., and Chen, S. C. 2011. Parameter tuning, feature selection and weight assignment of features for case-based reasoning by artificial immune system. *Applied Soft Computing. 11,* 8 (December, 2011), 5042-5052. DOI= http://doi.acm.org/10.1016/j.asoc.2011.05.054

[9] Guyon, I., and Elisseeff, Andr&#. 2002. An introduction to variable and feature selection. *J Mach Learn Res. 3,* 6 (March, 2002), 1157-1182.

[10] Saeys, Y., Inza, I., and Larrañaga, P. 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics. 23*, 19 (August, 2007), 2507-17. DOI= http://doi.acm.org/10.1093/bioinformatics/btm344

[11] Chuang, L. Y., Chang, H. W., Tu, C. J., and Yang, C. H. 2008. Improved binary pso for feature selection using gene expression data. *Comput Biol Chem. 32*, 1 (February, 2008), 29-38. DOI= http://doi.acm.org/10.1016/j.compbiolchem.2007.09.005

[12] Sahu, B., and Mishra, D. 2012. A novel feature selection algorithm using particle swarm optimization for cancer microarray data. *Procedia Engineering. 38*, 5 (December, 2012), 27-31. DOI= http://doi.acm.org/10.1016/j.proeng.2012.06.005

[13] Kardan, A. A., Kavian, A., and Esmaeili, A. 2013. Simultaneous feature selection and feature weighting with K selection for KNN classification using BBO algorithm. *International Conference, IKT.* 28, 4, (May, 2013), 349-354. DOI= http://doi.acm.org/ 10.1109/IKT.2013.6620092

[14] Kalayeh, M. M., Idrees, H., and Shah, M. 2014. NMF-KNN: Image Annotation Using Weighted Multi-view Non-negative Matrix Factorization. *IEEE Conference on CVPR.* (June, 2014), 184-191. DOI= http://doi.acm.org/10.1109/CVPR.2014.31

[15] Wu, J. L., and Li, I. J. 2010. A SOM-based dimensionality reduction method for KNN classifiers. *International Conference on SSE.* (August, 2010), 173-178. DOI= http://doi.acm.org/10.1109/ICSSE.2010.5551813

[16] Draszawka, K., Szymański, J., and Guerra, F. 2015. Improving css-KNN Classification Performance by Shifts in Training Data. *International Conference on IKC*. (January, 2016), 51-63. DOI=http://doi.acm.org/10.1007/978-3-319-27932-9_5

[17] Omranpour, H., and Ghidary, S. S. 2016. A heuristic supervised euclidean data difference dimension reduction for knn classifier and its application to visual place classification. *Neural Comput & Applic.* (October, 2016), 1867–1881. DOI=http://doi.acm.org/10.1007/s00521-015-1979-8

[18] Jing, Y., Gou, H., and Zhu, Y. 2013. An Improved Density-Based Method for Reducing Training Data in KNN. *International Conference on ICCIS.* (January, 2013), 972-975.

[19] Pawlovsky, A. P., and Nagahashi, M. 2014. A method to select a good setting for the kNN algorithm when using it for breast cancer prognosis. *International Conference on BHI.* (June, 2014), 189-192. DOI=http://doi.acm.org/ 10.1109/BHI.2014.6864336

[20] Li, N., Kong, H., Ma, Y., Gong, G., and Huai, W. 2016. Human performance modeling for manufacturing based on an improved knn algorithm. *Int J Adv Manuf Technol. 84*, 1 (February, 2016), 473-483. DOI=http://doi.acm.org/ 10.1007/s00170-016-8418-6

[21] Wang, B., Liao, Q., and Zhang, C. 2013. Weight Based KNN Recommender System. *International Conference on IHMSC.* 2, (August, 2013), 449-452. DOI=http://doi.acm.org/10.1109/IHMSC.2013.254

[22] Zhang, L., Zhang, C., Xu, Q., and Liu, C. 2015. Weigted-KNN and its application on UCI. *IEEE International Conference on IA.* (August, 2015), 1748-1750. DOI=http://doi.acm.org/10.1109/ICInfA.2015.7279570

[23] Shi, K., Li, L., Liu, H., and He, J. 2011. An improved KNN text classification algorithm based on density. *IEEE Proceedings of IEEE CCIS.* (September, 2011), 113-117. DOI=http://doi.acm.org/10.1109/CCIS.2011.6045043

[24] Liu, X., Ren, F., and Yuan, C. 2010. Use relative weight to improve the kNN for unbalanced text category. *Proceedings of IEEE NLPKE.* (August, 2010), 1-5. DOI=http://doi.acm.org/10.1109/NLPKE.2010.5587799

[25] Haixiang, G., Yijing, L., Yanan, L., Xiao, L., and Jinling, L. 2016. Bpso-adaboost-knn ensemble learning algorithm for multi-class imbalanced data classification. *Eng Appl Artif Intel,* (March, 2016), 176-193. DOI=http://doi.acm.org/10.1016/j.engappai.2015.09.011

[26] Bhattacharya, G., Ghosh, K., and Chowdhury, A. S. 2015. A probabilistic framework for dynamic k estimation in kNN classifiers with certainty factor. *Proceedings of IEEE ICAPR.* (January, 2015), 1-5. DOI=http://doi.acm.org/ 10.1109/ICAPR.2015.7050683

[27] Zhang, S., Zong, M., Sun, K., Liu, Y., and Cheng, D. 2014. Efficient kNN Algorithm Based on Graph Sparse Reconstruction. *International Conference on ADMA*. (December, 2014), 356-369. DOI=http://doi.acm.org/10.1007/978-3-319-14717-8_28

[28] Hamerly, G., and Speegle, G. 2010. Efficient Model Selection for Large-Scale Nearest-Neighbor Data Mining. *International Conference on BNCOD 27*. (July, 2010), 37-54. DOI=http://doi.acm.org/10.1007/978-3-642-25704-9_6

[29] Weng, F., Jiang, Q., Chen, L., Hong, Z., and Jiang, Q. S. 2007. Clustering ensemble based on the fuzzy KNN algorithm. International Conference on ACIS 18. (July, 2007), 1001-1006. DOI=http://doi.acm.org/10.1109/SNPD.2007.504

[30] Jiang, L., Cai, Z., Wang, D., and Zhang, H. 2014. Bayesian citation-knn with distance weighting. *Int. J. Mach. Learn. & Cyber. 5*, 2 (April, 2014), 193-199. DOI=http://doi.acm.org/ 10.1007/s13042-013-0152-x

[31] Kozak, K., Kozak, M., Stapor, K., Kozak, K., Kozak, M., and Stapor, K. 2006. Weighted k-nearest-neighbor techniques for high throughput screening data. *Int. J. of Bio & Med Sci.* (January, 2006), 1-4.

[32] Loftsgaarden, D. O., and Quesenberry, C. P. 1965. A nonparametric estimate of a multivariate density function. *Ann. Math. Statist*. 36, 3 (1965), 1049-1051.

[33] Mitra, P., Murthy, C. A., and Pal, S. K. 2002. Unsupervised feature selection using feature similarity. *IEEE Transactions on PAMI. 24*, 3 (March, 2002), 301-312. DOI=http://doi.acm.org/10.1109/34.990133

[34] Kosinski, M., Matz, S. C., Gosling, S. D., Popov, V., and Stillwell, D. 2015. Facebook as a research tool for the social sciences: opportunities, challenges, ethical considerations, and practical guidelines. *American Psychologist. 70*, 6 (2015), 543-556. DOI=http://doi.acm.org/10.1037/a0039210

[35] Fernández-Tobásd, I., and Cantador, I. 2014. Personality-Aware Collaborative Filtering: An Empirical Study in Multiple Domains with Facebook Data. *International Conference on EC-Web*. (September, 2014), 125-137. DOI=http://doi.acm.org/10.1007/978-3-319-10491-1_13