



Journal of
**Software
Engineering**

ISSN 1819-4311



Academic
Journals Inc.

www.academicjournals.com

Improvement and Comparison of Weighted k Nearest Neighbors Classifiers for Model Selection

¹Ming Zhao and ²Jingchao Chen

¹School of Aeronautic Transportation, Shanghai University of Engineering Science, Shanghai, 201620, China

²School of Information Science and Technology, Donghua University, Shanghai, 201620, China

Corresponding Author: Ming Zhao, School of Aeronautic Transportation, Shanghai University of Engineering Science, Shanghai, 201620, China

ABSTRACT

In order to choose a better model for pattern recognition and machine learning, four nearest neighbor classification algorithms are discussed under different weighted functions, k value and sample sizes, based on Euclidean distance metric. Three factors mainly affect these classifiers' performance, including the number of nearest neighbors, distance metric and decision rule. The experimental results demonstrated that a better classifier can be easily chosen for a given dataset through comparison the accuracy rate and the cost time through comparing different situations of four classification algorithms. Furthermore, several previous related works about improving classifier's performance have also been summarized which can provide a basic overview for both specialists and non-specialists, as well as how to select an algorithm and parameters as appropriate as possible.

Key words: Pattern recognition, supervised classification, k nearest neighbors, Euclidean distance, weighted function, accuracy, model selection

INTRODUCTION

In recent years, machine learning is increasingly becoming an important field of computer science. Numbers of core algorithms for pattern recognition and data mining are related to machine learning. Three machine learning algorithms widely used, namely, classification, clustering and prediction. This article focus on the classification algorithm.

Classification algorithms have been successfully used in several areas. Different applications have their own related issues. In automatic diagnosis of diseases, for example, classifications algorithms predict patients' health under medical observation but how to handle misjudge. In text classification, classification algorithms mainly categorize text for specific user according to topic and relevance but how to deal with gradually complex data objects (Rogers and Girolami, 2011).

K Nearest Neighbors (KNN) classifier is a classical supervised method in the field of machine learning based on statistical data. Cover and Hart had proved the k nearest neighbors classification deviation approximates to optimal Bayesian classifier (Cover and Hart, 1967). The k nearest neighbors classifier is widely used in such areas as text classification, pattern recognition, image processing and spatio-temporal classification.

The decision rule of the k nearest neighbors algorithm is to find test sample k nearest or most similar training samples in the feature space and then the test sample is assigned to a majority vote of its k nearest neighbors. Essentially, it is still based on statistical method. In the real world, most

of the practical data does not obey the typical model (such as Gaussian mixture, linearly separable, etc). Non parametric algorithms like k nearest neighbors happen to deal with the situation. If domain classes of test sample sets has more cross or overlap, k nearest neighbors is the more appropriate method.

The advantages of k nearest neighbors algorithm are as follows: (1) There is no explicit training phase or it is very minimal. Once the data is loaded into memory, it began to classify. This means the training phase is pretty fast, (2) The data set has been labeled, that is to say the selected neighbors of test sample have been classified correctly and (3) Independence of any data distribution, it only needs to be adjusted or assigned an integer parameter k.

The disadvantages of k nearest neighbors algorithm are as follows: (1) It has a costly testing phase. The cost is in terms of both time and memory. More memory is needed to store all training data (Bi and Bi, 2009; Khuman, 2012) and (2) It is highly dependent on number of samples. Along with the number of training samples increases, the classifier performance will be better and better, (3) Classification accuracy is affected because of the property is equivalent to the weight and (4) It is difficult to select an appropriate k value. Small or large value of k has different characteristics.

In this study, we focus on improving and comparing the accuracy of different k nearest neighbors algorithm based on Euclidean metric. After the introduction, we present the principle of k nearest neighbors algorithm, discuss the different parameters for k nearest neighbors and summarize previous research. Based on Euclidean distance, we experimented simple k nearest neighbors algorithm, attribute-weighted k nearest neighbors algorithm and instance-weighted k nearest neighbors algorithm in different k value, as well as compared performance in different situation. Still, reviews here are sufficiently rich and general so that one can gain useful intuition for other, perhaps more complex and learning problems.

MATERIALS AND METHODS

Basic principle for k nearest neighbors: The k nearest neighbors classifiers test objects based on the closest or most similar training sample in the feature space and the object is classified by a majority vote of its neighbors (Fig. 1). The nearest neighbor is determined by metric function.

Figure 1 presents example of the k nearest neighbors classification. The test sample (dot) should be classified by the k nearest neighbors classifier. If k = 3, it is classified to triangle. If k = 5, it is classified to square.

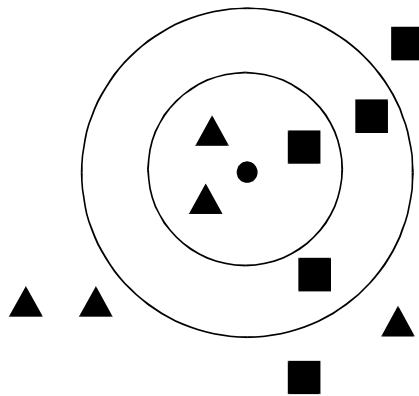


Fig. 1: Example of the k nearest neighbors classification

Table 1: Features of small or large value of k

Small value	Large value
Over-fit	Generalization
Negative effect of noise	Reduce negative effect of noise
Create distinct class boundaries	Create distinct class boundaries

That is, the k nearest neighbors of test sample X are identified as the labeled subset of feature vectors $\{(X_1, Y_1), \dots, (X_k, Y_k)\}$ of the training sample. The k nearest neighbors classifier then assigns X to the class $Y' = \text{maj}(Y_1, \dots, Y_k)$. If no single class appears with greatest frequency, then an auxiliary procedure can be invoked to handle ties.

In its original form, the nearest neighbor classifier is perhaps the simplest pattern classification algorithm, the classifier assigns any input feature vector to the class indicated by the label of the nearest vector in the reference sample. More generally, the k nearest neighbors classifier maps any feature vector to the pattern class that appears most frequently among the k nearest neighbors (Kulkarni *et al.*, 1998). The main algorithm steps can be described as follows:

Algorithm: Simple k nearest neighbors pseudo code

Procedure: Find class labels

Input: k, the number of nearest neighbors; D, the set of test sample; T, the set of training sample

Output: L, the label set of test sample

```

1:  read DataFile (Training Data)
2:  read DataFile (Testing Data)
3:  L = {}
4:  For each d in D and each t in T do
5:    Neighbors (d) = {}
6:    If |Neighbors (d)| < k then
7:      Neighbors (d) = Closest (d, t) ∪ Neighbors (d)
8:    End if
9:    If |Neighbors (d)| = k then
10:   Break
11: L = test Class (Neighbors (d)) ∪ L
12: End for

```

Notes:

Neighbors (d) return the k nearest neighbors of d

Closest (d, t) return the closest elements of t in d

test Class (S) return the class label of S

Generally, the training phase is executed once and the classification phase is executed many times until all test samples are classified.

Different parameters for k nearest neighbors: The k nearest neighbors classifiers performance mainly depends on three factors: k value the number of neighbors, distance metric and decision rule.

Choice of k: Small or large value of k has different effects (Table 1). Heuristic techniques such as cross-validation can help in selecting a good value for k (Khuman, 2012). If k is even it is necessary to define an auxiliary procedure to handle ties.

However, a natural question arising is; if there is a way of increasing the value of k as the sample size grows such that the resulting k_n nearest neighbor rule is consistent. Stone (1977) proved that the k_n nearest neighbor rule is universally consistent. Devroye *et al.* (1994) showed that the k_n nearest neighbor rule is also strongly universally consistent under the same conditions on the sequence $\{k_n\}$. Stone's consistency theorem suggests that the number of neighbors k considered in the decision should grow with the sample size n but not too rapidly. However, this theorem gives little guidance for the choice of k in a practical problem. As it turns out, the appropriate choice of k depends heavily on the actual distribution of (X, Y) .

As Cover and Hart (1967) observed, for some distributions $k = 1$ is optimal for all numbers. If $k = 1$, that is nearest neighbor algorithm, the nearest neighbor label of X is assigned. Indeed, Devroye *et al.* (1996) argued further that, when the expected posteriori probabilities of error is small, there is little advantage in choosing k larger than 3. Generally, the k value is a relatively small integer.

Choice of distance metric: Another parameter to be set by the user of the k nearest neighbors rule is the metric according to which distance are measured. The choice of metric plays an important role in the performance of the nearest neighbor classifier for a given sample size n . Different distance metrics can be used when calculating distance between the object points.

Minkowski distance: It can be universally described as Minkowski metric which is given in Eq. 1:

$$\text{Dist}(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad (1)$$

where, different values of $p \geq 1$ result in different commonly-used metric.

Manhattan distance: When $p = 1$, that is Manhattan distance (Eq. 2). It is the absolute distance total between two points on the standard coordinate system. The correlation of different features is not taken into account here:

$$\text{Dist}(X, Y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

Euclidean distance: When $p = 2$, that is Euclidean distance (Eq. 3). It is the most common metric used in KNN:

$$\text{Dist}(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3)$$

Euclidean distance measures an absolute distance between various points in a multidimensional space. At the same time, Euclidean metric need to ensure dimension indicators in the same level because of distance calculating based on the absolute values of various dimension characteristics. This metric should be used when the different features are not strongly correlated.

Chebyshev distance: When $p \rightarrow \infty$, Eq. 1 is represented as:

$$\text{Dist}(X, Y) = \lim_{p \rightarrow \infty} \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} = \max |x_i - y_i| \quad (4)$$

Mahalanobis distance: Since Euclidean distance cannot ignore differences in metric indicators, the data need to be standardized before using the Euclidean distance. That can derive another distance measure Mahalanobis distance. Let the vectors x and y be two input samples of the same distribution with the covariance matrix P . The Mahalanobis distance between sample x and y is defined as:

$$\text{Dist}(X, Y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)} \quad (5)$$

This metric should be used when the different features are strongly correlated. The covariance matrix Σ represents this correlation (Khuman, 2012).

Global weighted Euclidean metric is derived by Fukunaga and Flick (1984). Snapp and Venkatesh (1998) argue the merits of a weighted metric and show that for a family of smooth problems, the weighted Euclidean metric is optimal over all weighted metrics.

Smoothing the decision rule: One way of smoothing the k nearest neighbors decision rule is to introduce different weights to the different neighbors of the input point X , typically decreasing with the distance from Bailey and Jain (1978) proved that if k is fixed, the asymptotic probability of error of a weighted k nearest neighbor rule is minimal for all distributions if the weights are uniform and in this sense weighting is useless. However, if k and the weight vector are allowed to change with the sample size, weighting may be advantageous for finite sample sizes.

The k nearest neighbors classifies each unlabeled example by the majority label among its k nearest neighbors in training set. Its performance thus depends crucially on the distance metric used to identify nearest neighbors (Shi *et al.*, 2011). However, the principle based on maximum voting easily leads to equal votes of neighbors, that's draw phenomenon. In addition, if the quantity of sample is unbalance, such as one class samples is amount, the others is minority which cause the samples of the amount class are majority in k nearest neighbors when a test sample is coming. In reality, especially in small sample cases such as curse dimensionality and existing outliers, the sample quantity is little which usually leads to a dramatic degradation of classification accuracy. Thus different weighting methods are be used.

The goal of weighted functions is to make the distant neighbors have less effect on the majority vote than the closer neighbors. Most commonly used weighted functions of k nearest neighbors are attribute weighted and instance weighted based on Euclidean metric.

RESULTS AND DISCUSSION

A very large literature has focused on the nearest neighbor classifier dealing with diverse issues ranging from algorithmic innovations, error estimation, imperfect samples, editing experiments and computational concerns.

Characteristics of high-dimension of feature vector will not only reduce the classification efficiency but also affect the accuracy of the classification algorithm. At the same time, the efficiency of k nearest neighbors algorithm is influenced greatly by the distribution of the training set. Some approaches make use of KD-tree (Bentley, 1975), R-trees (Kim *et al.*, 1996), or X-trees

(Guttman, 1984) to find the nearest neighbors of each candidate points. These index structures are extremely efficient in finding the k nearest neighbors of a data point. However, the time required to search through these index structures scales exponentially with the number of dimensions. Consequently, their usefulness is constrained to low-dimensional spaces.

Kim and Park (1986) proposed a fast nearest neighbor finding algorithm based on the ordered lists of the training samples of each projection axis. The ordered partition contains two properties, one is ordering to bound the search region and the other is partitioning to reject the unwanted samples without actual distance computations. Similarly, researchers have proposed partitioning the space into regions (Knorr and Ng, 1999; Ramaswamy *et al.*, 2000) which allows for fast determination of nearest neighbors. Unfortunately, the aforementioned approaches are affected by the curse of dimensionality and do not scale to high-dimensional data (Bay and Schwabacher, 2003).

For high-dimensional dataset, the dataset is pre-processed into bins such that are close to each other in space are likely to be assigned to the same bin. The pre-processing performed facilitates fast convergence to a point's approximate nearest neighbors, only a point's approximate nearest neighbors and not its nearest neighbors (Ghoting *et al.*, 2008).

For imbalanced dataset classification, Wang *et al.* (2012) defined a new weight assignment model which can fully take into account the adverse effects caused by the uneven distribution of training sample between classes and within classes. Firstly, the model uses K-means algorithm based on Genetic Algorithm (GA) to cluster the training sample set. Secondly, the model computes the weight for each training sample in accordance to the clustering results and weight assignment model. Finally, the model uses the improved k nearest neighbors algorithm to classify the test samples which can significantly improve the identification rate of the minority samples and overall classification performance.

For multi-label classification problem, Oliveira *et al.* (2008) proposed a slightly modified version of the standard structure of the Probabilistic Neural Network algorithm (PNN) to deal with multi-label classification problems because the PNN classifier has high computational speed in the training stage. While k nearest neighbors classifier is very sensitive to the distribution of the samples, DBKNN algorithm based on density is an improved k nearest neighbors text algorithm.

In recent years, researchers focus on how to integrate k nearest neighbors algorithm with genetic algorithm, neural networks or fuzzy set theory in order to improvement its efficiency. Zanchettin *et al.* (2012) presented a hybrid KNN-SVM method for cursive character recognition. The main idea was to increase the k nearest neighbors recognition rate, sensible to different classes with similar attributes, using the SVM as a decision classifier, similar to Bellili *et al.* (2001) proposal. The adaptation here is to get the two most frequently classes in the KNN and use the SVM to decide between these two classes. The main disadvantage is the processing time (Bellili *et al.*, 2001). Musdholifah *et al.* (2010) proposed an algorithm that combines KNN density estimation and Kernel density estimation, to cluster the spatio-temporal data. In this approach, the KNN density estimation is extended and combines with Kernel function, where KNN contributes in determining the best local data iteratively for kernel density estimation. The local best is defined as the set of neighbor data that maximizes the kernel function. Silva and Del-Moral-Hernandez (2011) explored the advantages of Self-Organizing Maps (SOM) Artificial Network and KNN, so the KNN performs the classification process and the SOM work as pre-processing to the KNN classifier.

Experiment of different parameter

Selection of distance functions: Though there are several distance measurements in k nearest neighbors algorithm, by using leave one out cross validation of error estimation, for different values of k and for different distance metrics, the Euclidean metric is selected because of the best result.

Different weighted functions: In practice, the percentage of misclassified test samples is taken as an estimate of the error rate. In order for this error estimate to be reliable in predicting future classification performance, not only should the training set and the test set be sufficiently large but the training samples and the test samples must be independent.

The accuracy of k nearest neighbors algorithm is subject to some irrelevant attributes, when the numbers of irrelevant attributes increase, the predict accuracy of classification using Euclidean distance will become inaccurate.

Experiment results: Four k nearest neighbors algorithms have implemented by Siddharth Deokar in C++, that is SimpleKNN, AttributeWKNN, InstanceWKNN and BackwardWKNN. In this paper, we also implemented a modified version of them in VC++ and experiment on the following data sets:

Dataset 1: Consisting of 2 classes, 14 attributes, 224 samples for training and 46 samples for testing

Dataset 2: Consisting of 5 classes, 14 attributes, 103 samples for training and 100 samples for testing

In simpleKNN algorithm, all attributes are treated equally. The sample is multidimensional. x_i represents one attribute of sample x and y_i represents one attribute of sample y. The distance between sample x and sample y is defined as Eq. 6:

$$\text{Dist}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (6)$$

In AttributeWKNN algorithm, every attribute is associated with a weight W and learns weights by gradient decent and cross validation. The distance between sample x and sample y is defined as Eq. 7:

$$\text{Dist}(x, y) = \sqrt{(x_1 - y_1)^2 w_1^2 + (x_2 - y_2)^2 w_2^2 + \dots + (x_n - y_n)^2 w_n^2} \quad (7)$$

We experimented these algorithm on dataset which contains 224 training samples, 46 test samples and 2 classes. Every sample has 14 attributes. Figure 2 shows the classification accuracy of different algorithm in different k value.

We compared and analyzed the results, when k = 5, 7, 12, 13, 15, the simpleKNN algorithm has best classification accuracy 86.97%, when k = 7 k = 5, the AttributeWKNN algorithm has best classification accuracy 91.30%, when k = 6, the BackwardWKNN has best classification accuracy 86.96%. No matter how k value change, the classification accuracy rate of InstanceWKNN algorithm is constant, approximately 58.70%. Then we selected k = 5, 6, 7, 12, 13, 15 and calculate the runtime every algorithm cost (Table 2).

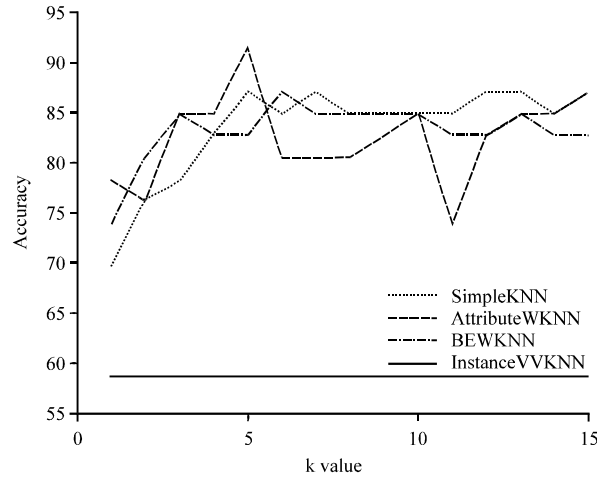


Fig. 2: Accuracy of various k nearest neighbors algorithm with different parameters

Table 2: Runtime (s) of k = 5, 6, 7, 12, 13, 15

Runtime (s)	5	6	7	12	13	15
SimpleKNN	1.359	1.313	1.297	1.328	1.297	1.328
AttributeWKNN	1178.910	1185.880	1179.810	1186.440	1179.630	1181.380
InstanceWKNN	387.687	388.938	387.750	387.641	386.703	386.734
BackwardWKNN	84.078	83.469	84.032	83.141	84.625	84.391

Comparison analysis: For model selection, we analyzed the performance of k nearest neighbors algorithm. In experiment, firstly, we must chose the right parameters for the classifiers. In previous works, Snapp and Venkatesh (1998) have proved that the weighted Euclidean metric is optimal over all weighted metrics and Devroye *et al.* (1996) suggested that the k value is a relatively small integer.

In our experiment, we have selected previously published algorithm by Siddharth Deokar, chosen k = 5, 7, 12, 13, 15 and Euclidean distance metric. Then we propose a novel method for classifier selection which depends on two factors: the accuracy rate and the runtime cost. We calculated the runtime every algorithm cost. We found that AtttributeWKK has better classification performance but it cost too long time from Table 2. Finally, we selected SimpleKNN algorithm and k = 7 for the specific dataset classification. But the runtime cost is a relative parameter. When the hardware configuration is different, the algorithm runtime cost is different. That is to say, for any dataset, every algorithm has its advantage and disadvantage, we should measure carefully to choose better classification scheme. In practice, the choice of a classifier is a difficult problem and it is often based on which classifiers happen to be available, or best known, to user.

CONCLUSION

Classification is the most researched topic of machine learning. In this paper, for selecting an appropriate classifier for real-world problem, the experiment study focus on the performance of classification algorithms based on k nearest neighbors. There are three factors which affect classification performance, that is the number of neighbor, distance measure and decision rule.

k Nearest Neighbor classifier is a supervised learning method based on statistical data. With four classical KNN algorithms, the experiment is implemented on specific dataset. Through

comparing the accuracy rate and calculating the cost time of four classification algorithms in different k value, researchers can choose one classification algorithm in one k value with relatively good performance for the specific dataset. The goal of experiment was to provide a basic overview for both specialists and non-specialists to how to decide a good algorithm for classification.

Furthermore, this study analyzes the general procedure of classification algorithm and summarizes several important issues and recent developments of classification problems. These include building index structure to find nearest neighbor, reducing dimension of high-dimensional dataset and being integrated into SVM or neural network. We do not discuss problems such as feature selection, dimension, class number which are other important and difficult problem of classification in the paper. Although, there are many other research topics that have been investigated in the literature, we believe that this selected study has covered the most important aspects of KNN classification.

REFERENCES

- Bailey, T. and A.K. Jain, 1978. A note on distance-weighted k-nearest neighbor rules. *IEEE Trans. Syst. Man Cybern.*, 8: 311-313.
- Bay, S.D. and M. Schwabacher, 2003. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 24-27, 2003, Washington, DC., USA.*, pp: 29-38.
- Bellili, A., M. Gilloux and P. Gallinari, 2001. An hybrid MLP-SVM handwritten digit recognizer. *Proceedings of the 6th International Conference on Document Analysis and Recognition, September 10-13, 2001, Seattle, WA.*, pp: 28-32.
- Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18: 509-517.
- Bi, X. and R. Bi, 2009. A survey of K nearest neighbor algorithm. *J. Sci. Technol. Innov. Herald*, 14: 31-31.
- Cover, T. and P. Hart, 1967. Nearest neighbor pattern classification. *IEEE Trans. Inform. Theor.*, 13: 21-27.
- Devroye, L., L. Györfi, A. Krzyżak and G. Lugosi, 1994. On the strong universal consistency of nearest neighbor regression function estimates. *Ann. Stat.*, 22: 1371-1385.
- Devroye, L., L. Györfi and G. Lugosi, 1996. *A Probabilistic Theory of Pattern Recognition*. Springer Science and Business Media, New York, USA., ISBN-13: 9780387946184, Pages: 636.
- Fukunaga, K. and T.E. Flick, 1984. An optimal global nearest neighbor metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6: 314-318.
- Ghoting, A., S. Parthasarathy and M.E. Otey, 2008. Fast mining of distance-based outliers in high-dimensional datasets. *Data Mining Knowledge Discovery*, 16: 349-364.
- Guttman, A., 1984. R-Tree: A dynamic index structure for spatial searching. *Proceedings of the 13th ACM SIGMOD International Conference on Management of Data, June 18-21, 1984, Boston, USA.*, pp: 47-57.
- Khuman, M.B., 2012. Classification of remote sensing data using KNN method. *J. Inform. Knowledge Res. Electron. Commun. Eng.*, 2: 817-821.
- Kim, B.S. and S.B. Park, 1986. A fast k nearest neighbor finding algorithm based on the ordered partition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8: 761-766.

- Kim, Y.I., Y.B. Park and J.H. Chun, 1996. A dynamic indexing structure for searching time-series patterns. Proceedings of the 20th International Conference on Computer Software and Applications, August 21-23, 1996, Seoul, pp: 270-275.
- Knorr, E.M. and R.T. Ng, 1999. Finding intensional knowledge of distance-based outliers. Proceedings of the 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK., pp: 211-222.
- Kulkarni, S.R., G. Lugosi and S.S. Venkatesh, 1998. Learning pattern classification-a survey. IEEE Trans. Inform. Thoery, 6: 2178-2206.
- Musdholifah, A., S.Z.B.M. Hashim, I. Wasito, 2010. KNN-Kernel based clustering for spatio-temporal database. Proceedings of the International Conference on Computer and Communication Engineering, May 11-12, 2010, Kuala Lumpur, Malaysia, pp: 1-6.
- Oliveira, E., P.M. Ciarelli, C. Badue and A.F. de Souza, 2008. A comparison between a KNN based approach and a PNN algorithm for a multi-label classification problem. Proceedings of the 8th International Conference on Intelligent Systems Design and Applications, November 26-28, 2008, Kaohsiung, pp: 628-633.
- Ramaswamy, S., R. Rastogi and K. Shim, 2000. Efficient algorithms for mining outliers from large data sets. Proceedings of the International Conference on Management of Data, May 15-18, 2000, Dallas, TX., USA., pp: 427-438.
- Rogers, S. and M. Girolami, 2011. A First Course in Machine Learning. CRC Press, Boca Raton, Florida, ISBN-13: 9781439824146, Pages: 305.
- Shi, K.S., L. Li, H.T. Liu, J. He, N. Zhang and W. Song, 2011. An improved KNN text classification algorithm based on density. Proceedings of the IEEE International Conference on Cloud Computing and Intelligence Systems, September 15-17, 2011, Beijing, China, pp: 113-117.
- Silva, L.A. and E. Del-Moral-Hernandez, 2011. A SOM combined with KNN for classification task. Proceedings of International Joint Conference on Neural Networks, 31 July-5 August, 2011, San Jose, CA., pp: 2368-2373.
- Snapp, R.R. and S.S. Venkatesh, 1998. Asymptotic expansions of the k nearest neighbor risk. Ann. Stat., 26: 850-878.
- Stone, C.J., 1977. Consistent nonparametric regression. Ann. Stat., 5: 595-645.
- Wang, C.X., Z.M. Pan, C.S. Ma, L.L. Dong and T. Zhang, 2012. Classification for imbalanced dataset of improved weighted KNN algorithm. J. Comput. Eng., 38: 160-168.
- Zanchettin, C., B.L.D. Bezerra and W.W. Azevedo, 2012. A KNN-SVM hybrid model for cursive handwriting recognition. Proceedings of the IEEE International Joint Conference on Neural Networks, June 10-15, 2012, Brisbane, Australia, pp: 1-8.