

# Enhanced K-Nearest Neighbor Algorithm

Dalvinder Singh Dhaliwal, Parvinder S. Sandhu, S. N. Panda

**Abstract**—In this paper, an enhancement of the k-Nearest Neighbor (k-NN) algorithm is proposed by incorporating min-max normalization of data as the initial stage before classification via the conventional k-NN algorithm and outlier removal as the final step. Under the proposed method, raw data is first normalized and the outlyingness factor ( $O_i$ ) for each observation computed. A threshold value of 1.0 is used to detect outliers and observations with  $O_i < 1.0$  fed for clustering by the k-NN method. In this study, the training set consisted of biological data derived from the Munich Information Center for Protein Sequences (MIPS) database. The algorithm was implemented in the PHP: hypertext preprocessor (PHP) scripting language. The data used was stored in a database implemented using MySQL using the Windows platform. The user interface for the application was constructed using advanced html using the Notepad text editor and linked to the backend using the PHP language. Using a dataset of 200 observations and a K parameter of 10, the outcomes obtained via the enhanced method were compared to those obtained via the conventional k-NN method. Comparisons of the results were made using the rand index. Results indicate that, compared to the naïve k-NN method, the enhanced method returns significantly improved performance.

**Keywords**—K-Nearest Neighbor Algorithm

## I. INTRODUCTION

THE k-nearest neighbor algorithm (k-NN) is a machine learning algorithm that has found wide usage in pattern recognition and data mining. In this method, the algorithm is fed with a training set and it uses this training set to classify objects. Each of the samples in the training set is tagged with a label. The input objects are classified based on the K parameter meaning that they are assigned to the class that is most pervasive among its closest k neighbours. For instance, if  $k=2$ , then the particular object is assigned to the classes of its closest 2 neighbours. Classification yields clusters of data where the input objects with similar features are grouped together or closely.

Different types of distance metrics are used in the implementation of the k-NN algorithm. Euclidean distance is widely used especially in instances where the dataset is made up of continuous variables. Defined, the Euclidean distance is the L2 distance between points  $p_1 = (x_1, y_1)$  and  $p_2 = (x_2,$

$y_2)$  represented by  $\sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$ . The Manhattan distance can also be used as a distance metric. The Manhattan distance is the L1 distance between points  $p_1=(x_1, y_1)$  and  $p_2=(x_2, y_2)$  represented by  $(x_1-x_2)^1 + (y_1-y_2)^1$ . Other common distance metrics are the Chebyshev, Mahalanobis, Minkowski and Hamming distance metrics.

Selection of the K parameter is an important albeit difficult choice in many instances. Even so, various heuristic methods have been used to optimally select the K parameter. A general rule of thumb in selecting this parameter is that this selection should be guided by the available dataset. Whereas larger K values decrease the impact of noise during classification, they significantly blur the distinction between classes.

*Pseudocode for the k-NN algorithm:*

The k-NN algorithm consists of the following steps. The first step is to identify the K parameter. The K parameter is the number of the closest neighbours in the space of interest. The second step involves the computation of the distance between the query vector and all the objects in the training set. A distance algorithm is used to compute this distance. The next step involves sorting the distances for all the objects in the training set and determining the nearest neighbor based on the k-th minimum distance. Thereafter, all the categories of the training set for the sorted values falling under  $k$  are collected. The final step involves using the majority of the closest neighbours as prediction values.

The main advantage of the k-NN method is that it is highly effective especially where use is made of large datasets. Additionally, the k-NN algorithm is robust even where noisy data is used. Disadvantages of the k-NN algorithm have also been widely documented. One of the main disadvantages of this technique is that the user is required to find out the K parameter. Besides, the k-NN technique is encumbered by a high computation cost which often reduces its speed and requires more powerful computers to use especially where very large datasets are involved. However, use of indexing techniques can help to drastically cut down on this computation cost.

Yet another drawback associated with the k-NN method is that it is a distance-based learning method and determining the distance type to use is often a difficult decision to make. Additionally, the choice of the number and or type of attributes to use is often not a clear one. Finally, the utility of the technique is compromised by noise or redundancy.

Various methods have been proposed to enhance the naïve k-NN algorithm. For example, it has been demonstrated that use of proximity graphs can enhance the utility of the technique. Incorporation of indexing techniques such as the

Dalvinder Singh Dhaliwal is working with Computer Science & Engineering Department, RIMIT Institute of Engineering & Technology, Mandi Gobindgarh (Punjab)- INDIA

Dr. Parvinder S. Sandhu is Professor with Computer Science & Engineering Department, Rayat & Bahra Institute of Engineering & Bio-Technology, Sahauran, Distt. Mohali (Punjab)-140104 INDIA (Phone: +91-98555-32004; Email: parvinder.sandhu@gmail.com).

Dr. S. N. Panda is working as Director & Professor at Regional Institute of Management and Technology, Mandi Gobindgarh (Punjab)- INDIA.

KD-tree or RS-tree is also cited as one of the ways that can be used to enhance the performance of the technique.

This paper proposes to enhance the utility of the naïve k-NN technique in the following ways. First, the paper seeks to ensure that the data fed through the algorithm is standardized so that all noise is eliminated and no redundant or useless features are incorporated into the feature space. This will help to address one of the main drawbacks of the naïve technique with regard to reduced utility due to noise and redundancy. The min-max normalization method is proposed as an effective way of dealing with the noise and redundancy problems.

Secondly, the paper seeks to reduce the high error rate that is typical of the naïve method, especially with reference to infinitely increasing datasets. By calculating the distance of the objects from the nearest neighbours and using a threshold to get rid of outliers, it is hoped that only valid data will be fed into the naïve k-NN algorithm consequently helping to weed out inconsistent data, and restricting the dataset to the minimally valid set. This will help to address the issue of the high error rate as well as improve the accuracy of the outcomes that are generated.

The paper is organized into 5 sections. The literature review section considers aspects of normalization of datasets especially as relates to clustering techniques and also reviews a novel method for the removal of outliers proposed by [13]. The methodology section presents the methods employed in this study. Results obtained using the proposed enhanced method and the naïve k-NN technique are then analyzed, comparisons between the 2 methods made and inferences drawn. Finally, the conclusion section summarizes the main findings of the study and proposes the work that should be carried out in future.

## II. NORMALIZATION OF DATASETS

Effective data mining can only occur if an equally effective technique for normalizing the data is applied. As explained by [30], the main purpose of normalization is to standardize all the features of the dataset into a specified predefined criterion so that redundant or noisy objects can be eliminated and use made of valid and reliable data. This is of utmost importance, not only because of the inconsistencies associated with data retrieved from widely differing sources but also because it has a bearing on the accuracy of the final output. To reemphasise the point, one of the big causes of inaccurate results during clustering is the use of datasets that are inundated with internal inconsistencies. This is more so true in methods such as the naïve k-NN clustering which make extensive use of distance metrics. These metrics are highly susceptible to differences in the scope of the object features.

In normalization, scaling of the data features is carried out so that these features can be restricted to a predefined and small range which usually varies from -1.0 to 1.0 or from 0 to 1.0. Since distance metrics such as the Euclidean distance are highly susceptible to inconsistencies in the size of the features

or scales being employed, normalization is essential as it can help prevent larger features from outweighing the smaller ones. In this regard, it helps to even out the magnitude and inconsistencies of the features [30].

As reported by [31] therefore, the importance of normalization is that it enhances the accuracy of the results that are obtained during clustering [31]. Findings by [30] suggest that significantly better outcomes are generated when pre-processing normalization is carried out than when clustering is done without prior normalization of data.

Several techniques for normalizing raw datasets have previously been described. These include the z-score normalization method, the min-max normalization method and the decimal scaling normalization method. Different methods that are available for normalization include the Min-max normalization, normalization by decimal scaling and the Z-score normalization [30]. Authorities propose the use of either the Min-max or Z score techniques in instances where there is a large dataset and the decimal scaling technique where the dataset is smaller. However, there is no universally defined rule for normalizing datasets and thus the choice of a particular normalization rule is largely left to the discretion of the user [30].

This project utilized the Min-max normalization technique. This technique involves the linear transformation of raw data, where min represents the theoretical minimum value and max the theoretical maximum value. Normalization by min-max followed the method described by [30] and represented by equation 1 below:

$$V' = (v - \min_a) / (\max_a - \min_a) \quad (1)$$

In the equation above,  $\min_a$  is the minimum value of a feature  $a$  while  $\max_a$  is its maximum value. A value  $v$  belonging to  $A-v$  in the range (0, 1) is mapped using the equation above.

## III. OUTLYINGNESS FACTOR (OI) AND THE REMOVAL OF OUTLIERS

Outliers are noisy data which do not correspond to the implicit model that produced the data under observation. As observed by [12], outliers are observations that should be eliminated so as to enhance the accuracy of clustering. The method proposed by [13] was used to modify the naïve k-NN technique. As described by [13], the outlier removal clustering (ORC) technique is an algorithm that is used to identify outliers from the input dataset and to simultaneously classify the objects. The simultaneous elimination of outliers and classification of the data helps to enhance the determination of distances and centroids. The purpose of the ORC algorithm is to create a codebook which is as similar as possible to the central features of the objects that were used to create the original data [13].

In their paper [13], Hautamaki et al explain that the ORC algorithm involves 2 steps. The first step encompasses the classification (clustering) of objects while the subsequent step

involves the iterative removal of vectors which are situated further away from the centroid. In order to determine whether to tag a vector as an outlier or not, a threshold factor is usually used. In the method described by Hautamaki et al [13] a threshold of 1 is used and this implies that all the vectors that have an outlyingness factor exceeding or equal to 1 are deemed to be outliers. On the other hand, vectors with factors less than 1 are not considered as outliers. Conversely, vectors with a threshold less than 1 are admitted as they are not deemed to be outliers. As the threshold is set at  $T < 1$ , vectors that have a higher threshold value are removed. The number of vectors which are removed from the dataset increases significantly as the threshold is reduced and as the number of iterations are increased [13].

The outlyingness factor for each vector in the space is calculated according to equation 2 below:

$$o_i = \frac{\|x_i - c_{p_i}\|}{d_{\max}} \quad (2)$$

In equation 2 above, the outlyingness factor is represented by  $O_i$ . Other terms in the vector include  $X_i$  which stands for each specific vector,  $C_{p_i}$  which is the centroid and  $d_{\max}$  which is the partition centroid. Explained, the outlyingness factor is obtained by subtracting the centroid of each cluster from the position of each input object from the dataset and dividing this difference by the maximum distance of the vector from the centroid or the partition centroid.

*Algorithm 1: Algorithm for the removal of outliers (I, T)*

```

C ← Run K-NN with multiple initial solutions, pick
best C

for j ← 1, . . . , I do
    dmax ← maxi{||xi - cpi||}

    for i ← 1, . . . , N do
        oi = ||xi - cpi|| / dmax

        if oi > T then
            X ← X \ {xi}

        end if

    end for

    (C, P) ← K-NN(X, C)

end for

```

The above algorithm is adapted from [13].

The pseudocode for the enhanced k-NN algorithm would

thus be as described below:

TABLE I  
THE ENHANCED K-NN ALGORITHM

<b>Input:</b>	Homogenous datasets that have d dimensions
<b>Output :</b>	Global partitions that have p datasets
<b>Step 1:</b>	Determine the minimum and maximum values of each attribute from each item in the dataset and set them in a central location
<b>Step 2:</b>	Calculate the global min <sub>a</sub> and max <sub>a</sub> values
<b>Step 3:</b>	Use equation 1 to normalize the items in the dataset
<b>Step 4:</b>	Identify the K parameter
<b>Step 5:</b>	Compute the distance between the query vector and all the objects in the training set.
<b>Step 6:</b>	Sort the distances for all the objects in the training set
<b>Step 7:</b>	Determine the nearest neighbor based on the k <sup>th</sup> minimum distance.
<b>Step 8:</b>	Collect all the categories of the training set for the sorted values under k
<b>Step 9:</b>	Use the majority of the closest neighbors as prediction values
<b>Step10:</b>	Remove vectors which are situated further away from the centroid iteratively

#### IV. METHODOLOGY

##### A. Client side / Server side scripting

The graphic user interface (GUI) was created using advanced html. The GUI was linked to the backend by embedding PHP code into the html script. Scripting was done using the notepad text editor on the Windows platform. Apache version 2.2.16 was used as the server.

##### B. Data collection

This study used biological data, specifically threading scores. The objective was to cluster the protein data so that the functional similarity between the proteins of interest can be quantified and the outcomes for the naïve k-NN and the enhanced k-NN compared thereafter. The threading scores were obtained from various biological databases. The training set was derived from the Munich Information Center for Protein Sequences (MIPS) database.

TABLE II  
BIOLOGICAL DATA TYPES AVAILABLE FOR EVALUATION

Data type
mRNA expression data
Essentiality data
Biological functions data
MIPS functional similarity data
GO functional similarity data
Co-evolution scores
Threading scores
Synthetic lethality
Gene cluster (for operon method)
Rosetta Stone
Interlogs in another organism
Absolute mRNA expression
Marginal essentiality
Absolute protein abundance
Co-regulation data
Phylogenetic profile data
Gene neighborhood data

### C. Database design

A database of the collected data was designed using MySQL. The database had a user and the protein features table.

### D. Normalization, K selection and $O_i$ calculation

Once the dataset was complete, the data was fed into the designed system. The min-max technique was used to normalize the data as explained in equation 1 above. Parameter selection was done using heuristic methods (the KNNXValidation module) and the value of  $k=10$  was chosen. The outlyingness factor for each of the observation was determined using the method of [13] and which is summarized in equation 2 above. A threshold of 1.0 was used to discriminate between the outliers and the observations deemed as more accurate.

## V. EXPERIMENTAL RESULTS AND COMPARISON

The table III shows the details of the dataset. The results obtained using the naïve k-NN and the enhanced k-NN methods respectively were compared using the rand index. The equation for the rand index is shown below:

$$R = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}} \quad (3)$$

TABLE III

Dataset	Number of features	Number of classes	Number of instances
Protein threading scores	10	2	200

DETAILS OF THE DATASET

TABLE IV  
ASSESSMENT OF THE NAÏVE K-NN METHOD AND THE ENHANCED K-NN METHOD USING THE RAND INDEX

Dataset	Naïve k-NN	Enhanced k-NN
Threading scores	0.0441177	0.0548187

From table IV above, it can be seen that the enhanced method yields superior results than the naïve k-NN method as the rand index of the former is higher.

## VI. CONCLUSION

This paper proposes an enhancement of the k-Nearest Neighbor (k-NN) algorithm by incorporating min-max normalization of data as the initial stage before classification via the conventional k-NN algorithm and outlier removal as the final step. Using a dataset of 200 observations and a K parameter of 10, the outcomes obtained via the enhanced method were compared to those obtained via the conventional k-NN method. Comparisons of the results were made using the rand index. Results indicate that, compared to the naïve k-NN method, the enhanced method returns significantly improved performance.

## REFERENCES

- [1] Bahl, P. & Padmanabhan, V. N., (2000). "RADAR: an in-building RF-based user Location and tracking system", IEEE INFOCOM 2000.
- [2] Barceló, F., Evonnou, F., de Nardis, L. & Tomé, P., (2006). "Advances in indoor Location", TOPO-CONF-2006-024.
- [3] Bock, H.H., & Diday, E. (2000): Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data, Series: Studies in Classification, Data Analysis, and Knowledge Organisation, Vol. 15, Springer-Verlag, Berlin.
- [4] Bolliger, P. (2008). "Redpin - adaptive, zero-configuration indoor localization through user collaboration", ACM International Workshop.
- [5] Brunato, M., & Battiti, R.,(2005). "Statistical Learning Theory for Location Fingerprinting in Wireless LANs", Computer Networks.
- [6] Carlotto, A., Parodi, M., Bonamico, C., Lavagetto, F., & Valla, M., (2008), "Proximity classification for mobile devices using wi-fi environment similarity", ACM International Workshop.
- [7] Correa, J., Katz, E., Collins, P., & Griss, M., (2008). "Room-Level Wi-Fi Location Tracking", Carnegie Mellon Silicon Valley, CyLab Mobility Research Center technical report MRC-TR-2008-02.
- [8] Chang, C.C. & Lin, C.L., (2001). LIBSVM : a library for support vector machines. Software Retrieved on 9th October 2010.
- [9] David, M. D., Elnahrawy, E., Martin, R.P, Wen-Hua J., Krishnan, P. & Krishnakumar, A. S. (2005). "Bayesian Indoor Positioning Systems", IEEE Infocom.
- [10] Esposito, F., Malerba, D., & Tamma, V. (2000). Dissimilarity Measures for Symbolic Objects. Chapter 8.3 in H.-H. Bock and E. Diday (Eds.), Analysis of Symbolic Data.
- [11] Gora, G., & Wojna, A.(2002): RIONA: A Classifier Combining Rule Induction and k-NN Method with Automated Selection of Optimal Neighbourhood, Proceedings of the Thirteenth European Conference on

- Machine Learning, ECML 2002, Lecture Notes in Artificial Intelligence, 2430, pp. 111-123, Springer-Verlag.165-185, Springer-Verlag
- [12] Guha, S., Rastogi, R., Shim, K.(1998): CURE an efficient clustering algorithm for large databases. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, Washington (1998) 73–84
- [13] Hautamaki V., Svetlana Cherednichenko, Ismo Kärkkäinen, Tomi Kinnunen, and Pasi Franti. (2005). Improving K-means by Outlier Removal. SCIA 2005, LNCS 3540, pp. 978–987, 2005. Springer-Verlag Berlin Heidelberg 2005
- [14] Hightower, J. & Borriello, G., (2001). "Location systems for ubiquitous computing", IEEE Computer.
- [15] Ho, W., Smailagic, A., Siewiorek, D.P. & Faloutsos, C., (2006). "An adaptive two phase approach to WiFi location sensing", IEEE Int. Conf.
- [16] Joubish, F. (2009). Educational Research Department of Education, Federal Urdu University, Karachi, Pakistan
- [17] Li, B., Salter, J., Dempster, A.G., & Rizos, C., (2006). "Indoor positioning techniques Based on Wireless LAN" IEEE Int. Conf
- [18] Fan, R.E., Chen, P.H. & Lin, C.J. (2005). "Working set selection using the second Order information for training SVM", Journal of Machine Learning Research 6,1889-1918.
- [19] Malerba, D., Esposito, F., & Monopoli, M. (2002). Comparing dissimilarity measures for probabilistic symbolic objects. In A. Zanasi, C. A. Brebbia, N.F.F. Ebecken, P. Melli (Eds.) Data Mining III, Series Management Information Systems, Vol 6, pp. 31-40, WIT Press, Southampton, UK.
- [20] Malerba, D., Esposito, F., Gioviale, V., & Tamma, V. (2001). Comparing Dissimilarity Measures in Symbolic Data Analysis. Pre-Proceedings of EKT-NTTS, vol. 1, pp. 473-481.
- [21] Manning C. D. & Schütze, H., (1999). Foundations of Statistical Natural Language Processing [M]. Cambridge: MIT Press.
- [22] Joachims T., (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features In: Proceedings of the European Conference on Machine Learning.
- [23] Li, B., Chen, Y., & Yu, S., (2002). A Comparative Study on Automatic Categorization Methods for Chinese Search Engine. In: Proceedings of the Eighth Joint International Computer Conference. Hangzhou: Zhejiang University Press, 117-120.
- [24] Li, B., (2003). Studies on Topic Tracking and Detection in Chinese News Stories. Ph.D. Thesis, Department of Computer Science and Technology, Peking University.
- [25] Saharkiz, (2009). K Nearest Neighbor Algorithm Implementation and Overview. An overview and implementation of KNN.
- [26] Sunil, A., David, M. M., Nathan, S. N., Ruth, S., & Angela, Y. W., (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions, Journal of the ACM (JACM), v.45 n.6, p.891-923, [doi>10.1145/293347.293348]
- [27] Wu, J. Q. (2007). Machine Learning and Computer Vision; University of Windsor, ON, Canada.
- [28] Wu, C. L., Fu, L. C., & Lian, F. L. (2004). "WLAN location determination in e-home via support vector classification", IEEE Int. Conf.
- [29] Yang, Y. & Liu X., (1999). A Re-examination of Text Categorization Methods. In: Proceedings of 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 42-49.
- [30] N. Karthikeyani Visalakshi, K. Thangavel (2009): Distributed Data Clustering: A Comparative Analysis. Foundations of Computational Intelligence (6) 2009: 371-397
- [31] Marcilio CP de Souto, Ivan G Costa, Daniel SA de Araujo, Teresa B Ludermir, Alexander Schliep (2008): Clustering cancer gene expression data: a comparative study, BMC Bioinformatics. 2008; 9: 497. doi: 10.1186/1471-2105-9-497.