Dynamic Nearest Neighbours Classifier For Integrated Data Using Object Oriented Concept Generalization

Ajita Satheesh

UIT, RGPV,Bhopal, MP,India
ajitargpv@yahoo.co.in

Ravindra Patel

UIT, RGPV, Bhopal, MP,India
ravindra@rgtu.net

***Abstract* - The k-nearest neighbor (k-NN) algorithm has been a promising classification tool. In spite of its extensive application, k-NN suffers from few inherent problems. A considerable number of the proposed approaches have exhibited quite promising results and has motivated further research on improving the k-NN method. In this paper, we devise a dynamic nearest neighbor classifier for data integrated via Generalization. Here, we make use of OO concept generalization to integrate the data collected from the different service providers, having varying formats, into a single consolidated data unit (training instance). Then, for effectual classification, we devise an enhanced variant of k-NN algorithm that employs: 1) Novel distance metric for similarity computation using Normalization and 2) Majority voting based on varied values of k. The performance of the proposed classifier was evaluated. The experimental results demonstrate the performance efficiency of the proposed k-NN classifier when compared to the traditional k-NN.**

Keywords: Classification, Generalization, k-Nearest Neighbor (k-NN) algorithm, Distance Metric.

I. INTRODUCTION

Classification is a well-known Data Mining task and it has been considered widely in the fields of statistics, pattern recognition, decision theory, machine learning literature, neural networks and more. The organization of data into a set of predefined classes is termed as Classification analysis. Classification approaches generally use a training set where all objects are already linked with known class labels. With the assistance of the training set the classification algorithm creates a model, and then, new objects can be classified based on the model [2].

One of the simplest classification methods used in data mining and machine learning is the *k*-Nearest Neighbor (*k*-NN) [13], [14]. It is the most accepted classification method due to its ease and practical efficiency: it doesn't necessitate fitting a model and it has been proved to have superior performance for classifying several types of data. But, the superior classification performance of *k*-NN is greatly dependent on the metric used for computing pair-wise distances between data points. Practically, to calculate *k* nearest neighbor data points of interest, Euclidean distances are often used as similarity metric. One often needs to find out or select a good distance metric to categorize high-dimensional data in real applications [9, 8].

Even though different measures can be used to calculate the distance between two points, the most desirable distance measure is one for which a smaller distance between two objects implies a greater likelihood of having the same class. One another factor that influences the performance of the classification algorithm is the complexity of the instances involved in classification. Generally, complex objects are structured into class/subclass hierarchy where every object attribute may contain additional complex objects. Majority of the existing works on complex data classification begins by performing a simplification to get data in a proper abstraction level, then classifying those generalized data by classical data classification algorithms [11]. Generalization is an abstraction principle in which classes are structured into a class hierarchy, which means that the properties shared by several classes are abstracted out into a common super-class [5].

Despite the popularity of *k*-NN in classification tasks, the performance and classification accuracy of the *k*-NN algorithm is still far from summit of optimality. In essence, the *k*-NN algorithm has been under the research scanner for years now, so as to improve its performance and classification accuracy. In this paper, we devise a dynamic nearest neighbor classifier with an enhanced similarity function for effectual classification of data. The proposed *k*-NN classifier takes as input, airline datasets collected from different airline service providers. The data thus accumulated will be of varied formats and structures, and hence prior to classification, the data are integrated by means of the OO concept, generalization to form a single consolidated data store. Also, we have effected two enhancements to the traditional *k*-NN classifier namely, an enhanced distance metric based on normalization and Majority voting based on a set of distinct *k* values. The experimental results portray the effectiveness of the proposed *k*-NN classifier when compared to the traditional *k*-NN classifier

The organization of the paper is as follows: Problem framework of the proposed research is presented in

Section 2. The proposed methodology of the dynamic nearest neighbor classifier with data generalization is given in Section 3. The experimental results are illustrated in Section 4 and Section 5 sums up the paper.

II. PROBLEM FRAMEWORK

This section presents a brief description of the traditional *k*-NN classifier and the framework of the classification problem dealt with in this paper.

A. *K-NN Algorithm*

The *k*-nearest neighbor classification was presented by Cover and Hart in [1]. The classification rules of *k*-NN are created by the training samples alone, with no other additional data [3]. In a more complicated approach, *k*-NN classification [6,7] , finds a group of k objects in the training set that are nearest to the test object, and bases the assignment of a label on the predominance of a particular class in this neighborhood. The *k*-NN is a lazy learning technique, and instead of estimating the target function once for the whole instance, they delay processing until classification. This necessitates the test instance to be compared with all the samples in the training set [12]. There are three main elements of this method: a set of labeled objects, e.g., a set of stored records, a distance or similarity metric to compute distance between objects, and the value of *k*, the number of nearest neighbors [4]. The Performance of *k*-NN Classification is depended mostly on the proficient selection of *k*-Nearest Neighbors. All the attributes describing an instance does not have same importance in picking the nearest neighbors. In real world, the influence of different attributes on the classification keeps on varying with time [17].

Issues of the Traditional K-NN Classifier

Besides its popularity in classification tasks, the performance and classification accuracy of the *k*-NN algorithm has not been very satisfactory. Several key issues that affect the performance of *k*-NN are,

- The choice of k. The parameter *k* in *k*-NN algorithm is often chosen based on the experience or knowledge about the classification problem at hand. If *k* is too small, then the result can be sensitive to noise points. On the other hand, if *k* is too large, then the neighborhood may include too many points from other classes [10].
- In *k*-NN, instead of estimating the target function once for the entire instance, it delays processing until a new instance must be classified and it needs to compare a test instance or document with all samples in the training set.
- The performance of *k*-NN algorithm greatly depends on the similarity function [12].
- The class probability estimation is based on a simple voting of the nearest neighbors. The simplest method is to take a majority vote, but this can be a problem if the nearest neighbors vary widely in their distance and the closer neighbors more reliably indicate the class of the object.

B. *Our Contribution*

The proposed research is aimed at enhancing the traditional *k*-NN algorithm such that some of the issues identified in the *k*-NN are resolved. The solutions proposed for the issues considered are,

- Majority voting based on dynamic values of *k*.
- A novel distance metric based on normalization as similarity function. The novel distance metric based on normalization presented in this paper ensures that, individual features with higher order numerical ranges do not disproportionately influence the classification process.
- Training Set Preparation: The proposed research processes on data collected from multiple airline service providers. The data accumulated will be of varying formats and structures making it unsuitable for classification processes. The proposed approach makes use of the OO Concept, Generalization for integrating the data collected from multiple data sources into a single data unit suitable for classification.

III. DYNAMIC NEAREST NEIGHBORS CLASSIFIER FOR INTEGRATED DATA USING GENERALIZATION

The proposed dynamic nearest neighbor classifier based on normalization is detailed in this section. *k*-NN is a renowned method of object classification that classifies objects based on closest training samples in the feature space. *k*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The training input to the *k*-NN will be the set of feature vectors and the corresponding class labels of the training instances. During the classification phase, the *k*-NN actually measures the distance between the training instances and the test instance employing a suitable distance metric. The training instances are then sorted based on the distance measures computed and ‘*k*’ closest samples are selected. Based on the majority voting obtained from ‘*k*’ closest samples, the test instance is classified into the particular class. Hence, it is well understood that the suitability of the training instance greatly influences the accuracy of the classification. The proposed research is composed of two phases namely,

- Data Generalization

- Data Classification via Proposed dynamic nearest neighbor classifier.

The proposed dynamic nearest neighbor classifier is aimed at classifying the airline data instances collected from different airline service providers.

A. Data Integration via Generalization

Generally, different organizations availing identical services will have their individual databases according to their own specific needs. Consider a centralized organization that wants to offer some special services common to all these different service providers. So, the centralized organization necessitates all those set of tables available with the service providers. The data collected from these multiple data sources will be in varying formats. Clearly, it is highly difficult to process on data with distinct formats. So as to process this data, two individual solutions has been processed,

1. Individual tables have to be queried and the results will have to be integrated.
2. The attributes of interest present in the individual tables can be generalized to form a single table suitable for querying.

Of the two solutions offered, the first solution is a computationally expensive task, and so in the proposed approach, we generalize the multiple tables accumulated from the different service providers. Data Generalization can be seen an effective means of combining data residing in different sources and providing users with a unified view of these data. The process plays a significant role in a variety of situations both commercial (when two similar companies need to merge their databases) and scientific (combining research results from different bioinformatics repositories, for example). In our case, we consider the airline datasets collected from different airline service providers. In the proposed approach, the attributes $\{a_1,\ldots,a_n\}$ of interest for the end user are extracted from the different tables $\{T_1,\ldots,T_m\}$ to create a generalized table G. Generally, a database D can be termed as a collection of tables T, with each table related to others by means of some defined relationship R. The basic structure of a relational database holds for every organization, yet the individual attributes and the relations' existing between them differs among organizations. With generalization, we first extract all those significant attributes of interest common in all tables. These set of common attributes form the attribute set of the generalized table G. Subsequently, the corresponding attribute values in the individual tables are fetched to attain the generalized table G. For instance, consider airway datasets collected from different airline service providers (Air India, SpiceJet, Jet Airways, Paramount and more) each with distinct set of attributes. For generalization, the common and decisive attributes in the airline datasets i.e. Flight name, Flight id, Source, Destination, Arrival time, Departure time, Fare and more. Other than these attributes, the individual airline datasets are likely to have some specific attributes. The generalized table G will have all the aforesaid attributes as its attribute set with their corresponding values extracted from the different tables.

Object Oriented Concept - Generalization

Generalization can be defined an abstraction principle based on which classes are ordered into a class hierarchy in such a way that properties shared by several classes are abstracted out into a common super class [5]. A sub-class and a super-class are connected to each other by means of Generalization. With generalization hierarchies, the common set of properties and operations among super-class and subclasses are defined only once in the super-class, and the objects of the corresponding subclasses inherit those properties and operations from the super class. A relationship between a more general thing and a more specific thing is termed as generalization:

- The more specific thing is unswerving in every way with the more general thing.
- The more specific thing can be substituted where ever the more general thing is expected. This is called as substitutability principle.
- Generalization applies to all classifiers and some other modeling elements.

The generalized table G serves as the training instance for the proposed k-NN classifier. Each tuple in the generalized table is then assigned a class label that describes category that the instance belongs to. The generalized table G with respective class labels is represented by G'.

B. Data Classification via Dynamic Nearest Neighbor Classifier

In a general scenario, the classification rules of the traditional *k*-NN are produced by the training samples themselves without any additional data. *k*-NN classification algorithm determines the category of the test sample according to the largest category probability of the *k* training samples which are the nearest neighbors to the test sample. In the traditional *k*-NN algorithm, all the samples with class labels are used for training, that is, to classify new test instance, it is essential to calculate similarities between it and all samples in the training sets, and then choose *k* nearest neighbor samples which have larger similarities [15, 16]. The steps involved in the proposed dynamic nearest neighbor classifier with an enhanced distance metric are presented below.

Algorithm

Input: A set of training instances G', x_o the query point.

Output: Class label of x_o .

P → Number of Features in the training set.

Training set G' , a set of training points $\{x_i, y_i\}_1^N$ with $x_i \in R^p$ and $y_i \in \{1,...,m\}$. Each point in G', x_i denotes the i^{th} feature vector, x_{ij} its j^{th}

N →Total number of instances in Training set.

1. Similarity computation between the training instance x_i and the query point x_o using an enhanced distance metric based on normalization.

Enhanced Distance Metric Based on Normalization

The traditional *k*-NN algorithm works well, with every feature in the training instance equally distributed and evenly influencing the classification of the test instance. Whereas, the condition becomes complex when the features used in *k*-NN vary widely in their corresponding range of values, for instance, two features say A and B, where B may range in values from 1000 to 5000 while another feature A ranges in values from 0 to 1. In the above scenario, the *k*-NN algorithm with defined distance metric say straight Euclidean distance, will classify the test instances greatly in favour of the higher valued feature, say B. In order to improve on this, the Euclidean distance formula is adjusted by using the normalization. The design of the enhanced distance metric based on normalization,

i) The individual values of the training feature vector x_{ij} and the query point x_o are normalized by using the equation 1.

$$x' = \left(\frac{x - x_p^l}{x_p^h - x_p^l} \right) * (U_b - L_b) \qquad (1)$$

Where x_p^l and x_p^h denotes the lowest and highest values corresponding to the attributes in Training set G'. U_b and L_b indicates the upper and the lower bound of the normalized values. On normalization, the sample values are generally scaled within their boundary values.

ii) The Euclidean distance metric is employed to measure the similarity between x_{ij} and x_o . The distance function $d(x, y)$, where x, y are scenarios composed of N features, such that $x = \{x_1, \cdots, x_n\}$, $y = \{y_1, \cdots, y_n\}$.

$$d(x, y) = \sum_{i=1}^{n} \sqrt{x_i^2 - y_i^2}$$

Where x_i is the value of i^{th} feature of x data point and y_i is the value of i^{th} feature of y data point.

2. The training samples are sorted in ascending order based on the similarity measures computed by the enhanced distance metric.

$$S_N = sort(x_i, y_i)_1^N$$

3. Choice of k . The classification efficiency of the k-NN classifier is greatly determined by the k-nearest neighbors chosen for classification. Also, the value of k must be neither too high nor too low. In our approach, s number of distinct k values are chosen and for each value of k ,

i) The first k sorted instances in the training set are fetched.

$$S_k = (x_i, y_i)_1^k \;\; ; k < N$$

ii) The intermediate class label of the query point x_o is determined by means of majority voting among the k nearest neighbors chosen.

$$Y_j = \arg\max(y_i)_1^k$$

The process results with s number of class labels, one corresponding to each of the ‘ s ’ k values chosen.

4. The actual class label of the query point x_o is determined by means of majority voting among the s class labels attained in step 3.

$$class(x_o) = \arg\max(Y_j)_1^s \;\; ; j = 1, \cdots, s$$

IV. EXPERIMENTAL RESULTS

In this section, we have presented the experimental results of the proposed dynamic nearest neighbor classifier. The research work is programmed in java. The proposed research work takes as input airline datasets collected from different airline service providers (Air India, SpiceJet, Jet Airways, Paramount and more). All these airline datasets have attributes namely, flight name, flight id, source, destination, arrival time, departure time and fare in common. In addition, each service provider's offer unique services like food, concession and more. For centralized classification, we just need those common attributes and in the proposed research, the OO concept generalization is employed to group those similar attributes into a single generalized table. Then, the individual instances in the generalized table are labeled with appropriate class names (Low, Medium and High). The labeled table serves as the training set for the proposed *k*-NN classifier. Lastly, when a test instance is provided, the proposed *k*-NN classifier based on normalization classifies the test instance to the suitable class. The performance of the traditional *k*-NN algorithm and the proposed dynamic nearest neighbor classifier algorithm in classification are portrayed in the tables 1 & 2 respectively. Each table possesses the classification results of ten different test instances as classified by the traditional and proposed *k*-NN algorithms. Those test instances misclassified by the traditional *k*-NN classifier are highlighted.

Table 1. Performance of the traditional *k*-NN classifier

Id	Name	Departure	Arrival	Airtime	From Place	To Place	Fare	Class
1	**Jet airways**	**19.50**	**20.45**	**0.55**	**Bhopal**	**Indore**	**2980**	**Low**
2	Spice Jet	5.4	8.5	3.1	Bangalore	Kolkatta	12950	High
3	**Jet airways**	**8**	**11**	**3**	**Trivandram**	**Delhi**	**14980**	**Low**
4	**Air India**	**7.45**	**10**	**2.15**	**Mumbai**	**Delhi**	**5095**	**Low**
5	Jet airways	6	6.45	0.45	Bhopal	Indore	3179	Medium
6	Indigo	10.2	11.3	1.1	Chennai	Bangalore	2490	Low
7	Indigo	3.2	3.2	3.1	Trivandram	Kolkatta	12950	High
8	Jet airways	13.5	16	2.1	Chennai	Kolkatta	3929	Low
9	**Paramount**	**17.4**	**18.43**	**1.03**	**Chennai**	**Bangalore**	**3029**	**Low**
10	Spice Jet	9.15	11.55	2.4	Bangalore	Mumbai	4670	Medium

Table 2. Performance of the proposed dynamic nearest neighbor classifier

Id	Name	Departure	Arrival	Airtime	From Place	To Place	Fare	Class
1	**Jet airways**	**19.50**	**20.45**	**0.55**	**Bhopal**	**Indore**	**2980**	**Medium**
2	Spice Jet	5.4	8.5	3.1	Bangalore	Kolkatta	12950	High
3	**Jet airways**	**8**	**11**	**3**	**Trivandram**	**Delhi**	**14980**	**Medium**
4	**Air India**	**7.45**	**10**	**2.15**	**Mumbai**	**Delhi**	**5095**	**Medium**
5	Jet airways	6	6.45	0.45	Bhopal	Indore	3179	Medium
6	Indigo	10.2	11.3	1.1	Chennai	Bangalore	2490	Low
7	Indigo	3.2	3.2	3.1	Trivandram	Kolkatta	12950	High
8	Jet airways	13.5	16	2.1	Chennai	Kolkatta	3929	Low
9	**Paramount**	**17.4**	**18.43**	**1.03**	**Chennai**	**Bangalore**	**3029**	**Medium**
10	Spice Jet	9.15	11.55	2.4	Bangalore	Mumbai	4670	Medium

V. CONCLUSION

In this paper, we have designed a dynamic nearest neighbor classifier for data integrated via Object oriented concept, Generalization. Here, we utilized the OO concept generalization to integrate the data collected from multiple service providers into a single consolidated data unit (training instance). Then, for effectual classification, we have devised an enhanced alternative of traditional *k*-NN that exploits a novel distance metric for similarity computation using Normalization and majority voting based on varied values of *k*. The classification accuracy of the proposed *k*-NN classifier was evaluated with data collected from different Airline service providers. The experimental results demonstrated the classification efficiency of the dynamic nearest neighbor classifier when compared to the traditional *k*-NN.

REFERENCES

[1] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification”, IEEE Transactions on Information Theory, Vol. 13, No. 1, pp. 21-27, 1967.

[2] Osmar R. Zaïane, “Introduction to Data Mining”, Chapter I, CMPUT690 Principles of Knowledge Discovery in Databases, 1999.

[3] Yang Lihua, Dai Qi and Guo Yanjun, “Study on KNN Text Categorization Algorithm”, Control and Automation, No.172, pp.269-270, 2006.

[4] Zalan Bodo and Zsolt Minier, "On Supervised and Semi-Supervised K-Nearest Neighbor Algorithms", Studia Informatica, No.2, 2008.

[5] G. Gottlob, M. Schrefl, B. Rock, “Extending Object-Oriented Systems with Roles”, ACM Transactions on Information Systems, Vol.14, No.3, pp. 268–296, 1996.

[6] Pang-Ning Tan, Michael Steinbach and Vipin Kumar, "Introduction to data mining", Pearson Addison-Wesley, 2006.

[7] Fix E, Hodges JL, “Discriminatory analysis, non-parametric discrimination”, International Statistical Review, Vol. 57, No. 3, pp. 238-247,1989.

[8] Pradeep Kumar, M. Venkateswara Rao, P. Radha Krishna, and Raju S. Bapi, "Using Sub-sequence Information with kNN for Classification of Sequential Data", in proceedings of Second International Conference on Distributed Computing and Internet Technology (ICDCIT), Bhubaneswar, India, December 2005.

[9] Martin Renqiang Min, David A. Stanley, Zineng Yuan, Anthony Bonner, Zhaolei Zhang, "Large-Margin kNN Classification Using a Deep Encoder Network", pp. 13, 2009.

[10] Xindong Wu, Vipin Kumar, Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu ,Philip S. Yu , Zhi-Hua Zhou, Michael Steinbach, David J. Hand and Dan Steinberg, "Top 10 algorithms in data mining", Knowledge and Information Systems, Vol.14 , No.1, pp. 1-37, 2007.

[11] Chidchanok Songsiri, Thanawin Rakthanmanon, Kitsana Waiyamai, "Object Oriented Database Mining: A Novel Approach for Data Classification", in proceedings of National Computer Science and Engineering Conference Data Mining and Data Classification (NCSEC'2002), 2002.

[12] Chuanyao, Yuqin, Zhang, Hu, "A Fast KNN Algorithm Based on Simulated Annealing", in proceedings of the International Conference on Data Mining, pp.46-51, Las Vegas, USA, June 2007.

[13] Yiming Yang and Xin Liu, "A re-examination of text categorization methods", in proceedings of 22nd Annual ACM Conference on Research and Development in Information Retrieval, Berkeley, California, United States, pp.42-49,1999.

[14] Thorsten Joachims, “Text categorization with support vector machines: Learning with many relevant features”, in proceedings of the 10th European Conference on Machine Learning, pp. 137 – 142, Berlin, 1998.

[15] Dasarathy “Nearest-Neighbor Classification Techniques”, IEEE Computer Society Press, Los Alomitos, CA, 1991.

[16] Globig and Wess, “Symbolic Learning and Nearest-Neighbor Classification”, Information Systems and Data Analysis, 1994.

[17] Anupam K.N., Syed M. R., Akram S., "An Enhancement of k-Nearest Neighbor Classification Using Genetic Algorithm", proceedings of the Conference of the Midwest Instruction and Computing Symposium, USA, April 8 - 9, 2005.