Journal of Theoretical and Applied Information Technology

www.jatit.org

KNNBA: K-NEAREST-NEIGHBOR-BASED-ASSOCIATION ALGORITHM

[1]MEHDI MORADIAN, [2]AHMAD BARAANI

[1]Department of Computer Engineering, University of Isfahan, Isfahan, Iran-81746

[2]Assis. Prof., [1]Department of Computer Engineering, University of Isfahan, Isfahan, Iran -81746

E-mail: m.moradian61@gmail.com, ahmadb@eng.ui.ac.ir

ABSTRACT

KNN algorithm is one of the best and the most usable classification algorithms which is used largely in different applications. One of the problems of this algorithm is the equal effect of all attributes in calculating the distance between the new record and the available records in training dataset ,how ever ,may be some of these attributes are less important to the classification and some of these attributes are more important. This results in misleading of classification process and decreasing the accuracy of classification algorithm. A major approach to deal with this problem is to weight attributes differently when calculating the distance between two records. In this research we used association rules to weight attributes and suggested new classification algorithm K-Nearest-Neighbor-Based-Association (KNNBA) that improves accuracy of KNN algorithm.

We experimentally tested KNNBA accuracy, using the 15 UCI data sets [1], and compared it with other classification algorithms NB, NN, C4.4, NBTREE, VFI, LWL and IBK. The experimental results show that KNNBA outperforms these seven algorithms.

Keywords: *Attribute Weighting, KNN Algorithm, Association Rules, Classification Algorithms*

1. INTRODUCTION

The KNN algorithm was originally suggested by Cover and Hart [2], nowadays, it is the most usable classification algorithm .it is a lazy algorithm, so it has less usability and is labor intensive when the training dataset is large [3]. This algorithm operation is based on comparing a given new record with training records and finding training records that are similar to it.

Each record with n attributes represents a point in an *n*-dimensional space. Therefore, all of the training records are stored in an *n*-dimensional space. When given a new record, KNN algorithm searches the space for the k training records that are nearest to the new record as the new record neighbors and then predict the class label of new record with use of the class label of these neighbors.

In this algorithm nearest is defined in terms of a distance metric such as Euclidean distance [3]-[4]. Euclidean distance between two records (or two points in *n*-dimensional space) is defined by forrmula1.

$$x_1 = (x_{11}, x_{12}, ..., x_{1n})$$
$$x_2 = (x_{21}, x_{22}, ..., x_{2n})$$
$$dist(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \quad (1)$$

, where x_1 and x_2 are two records with n attributes. Formula1 measures the distance between x_1 and the point x_2, in terms of take the difference between the corresponding values of that attribute in record x_1 and in record x_2.

Now, we should be combining k neighbors to provide a classification decision for the new record, so we need a combination function. Generally, two types of combination functions exist: unweighted voting and weighted voting. In the unweighted voting combination function, the class label which has the majority between neighbors of new record is selected as the class label of the new record without considering the preference of each neighbor. But, in the weighted voting more weight is given to some neighbors, that are so close to the new record or in other words, the ones which are more similar to new record.

Vol6. No1. (pp. 123 - 129)

Thus, more weighted records have more effects on determining the class label. Formula 2 is one way of giving the weight to neighbors [4].

$$weight(neighbor) = \frac{1}{DISTANCE(neighbor, new\,\mathrm{Re}cord)} \quad (2)$$

, $DISTANCE(neighbor, new\,\mathrm{Re}cord)$ is the Euclidean distance between new record and one of its neighbors. Formula2 can calculate vote or weight of all neighbors. With this weighting formula closer neighbors have a larger voice in the classification decision than do more distant neighbors. Now we calculate sum of the weight of neighbors which have a same class label. Class label is selected as the class label of new record that sum of the weight of neighbors which have that class label is greater.

The equal effect of all attributes in figuring out the distance between two records is one of this algorithm's problem ,however, some of attributes have more effect in records classifying and on the other hand, the interference of some irrelevant attributes in the classification process results in misleading of classification process. In order to resolve this problem, one weight is given to each attribute, and the distance between tow records is figured out by Manhattan distance (formula3) instead of Euclidean distance (formula1). The aim of this research is to suggest a new algorithm KNNBA that with giving weight to the attributes, increase the classification accuracy of KNN algorithm. KNNBA suggested a novel approach for weighting attributes with using association rules.

The experimental results show that average classification accuracy of KNNBA in the 15 UCI datasets is improved more than 7% with respect to KNN algorithm. In some datasets this improvement is very more, for example in Hayes dataset this improvement is more than 35%.

The rest of this paper is organized as follows. In section2, we introduce the related work on improving KNN algorithm with weighting attributes. In section3, first we introduce a novel feature weighting scheme to find the most relevant features and then we present the suggested algorithm KNNBA. In section4, Experimental result of using KNNBA in the 15 UCI datasets is discussed and compare accuracy of the KNNBA algorithm with the seven other classification algorithms. In section5, conclusions of the obtained results are described.

2. RELATED WORK

Unlike the most of classification algorithms such as decision tree which use only one subset of attributes for classification, the KNN algorithm uses all the record attributes equally [5]. How ever, all attributes might not have the same role in the classification process. May be some of these attributes are irrelevant to the classification, these irrelevant attributes can lead to distinguish two near records so far from each other and so the classification not to be done correctly. Idiomatically, this problem is called as a curse of dimensionality [5]. In order to resolve this problem should be indicating which attributes are more or less effective for classifying the new record.

To do so, the weight W_i should be identified for each attribute i. How great the attribute weight may be, it affect more in figuring out the distance. Therefore vector $W = (w_1, w_2, ..., w_n)$ is identified for weights of n attributes of records, then the formula3 to figure out the distance between two records x_1 and x_2, is as following [4]-[5]-[6].

$$x_1 = (x_{11}, x_{12}, ..., x_{1n})$$
$$x_2 = (x_{21}, x_{22}, ..., x_{2n}) \quad (3)$$
$$dist(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (w_i * (x_{1i} - x_{2i}))^2}$$

In fact, in this method of figuring out of the distance, not only, the quantity of attributes is considered but also, the quality of attributes is emphasized, so it increases the classification accuracy. Its obvious that how accurate weights may be the classification accuracy increase but, if the weights are not selected accurately, the classification accuracy even decreases than before.

Researchers suggest different method to provide the vector of attributes weight. The most direct method is the cross–validation [4] in which some of records is selected randomly as the training dataset and the weight of all attributes is considered equal to a default value and execute the algorithm KNN, afterwards, change the weights and again execute the algorithm KNN and continue it until get the less error (more accuracy) in classification process or in other words, get the best weights for attributes. Thereafter, in order to get the best possible weight vector for attributes, test these weights in other training datasets in a same manner. Obviously, this method takes too times and is not usable.

The weight adjusted k-nearest–neighbor (WAKNN) algorithm [7] considers giving weight to the attributes in order to increase the classification accuracy. This algorithm proposed for text datasets. In this algorithm discriminating words are given more weight and importance. This algorithm for distinguishing the discriminating words checks the

information relation between that word and class label.

The dynamic k-nearest–neighbor naïve bayes with attribute weighted (DKNAW) algorithm [8] uses the mutual information exist between each attribute and class label, in order to give weight to attributes and tries to improve the classification accuracy. In fact it could be said that this algorithm uses the idea of WAKNN method [7] to classify context in classification of non-context data. After this algorithm selects the k neighbors of new record, gives it to the bayes algorithm as training dataset and the bayes algorithm classify the new record.

The dynamic k-nearest neighbor (DKNN) algorithm [6] uses a method based on chi–squared to give weight to the attributes. The chi–squared distance represents that how affected one particular attribute can be in predicting class label $pr(C \mid x)$ and this is the major idea to finding features relevance weights or in other words weights of attributes.

The improved k nearest neighbors (IKNN) algorithm [9] uses the forward neural network to figure out the attributes weight and of course, in this algorithm, the rate of searching neighbors is increased by clustering records. For each record, the value of attributes as the input is given to the system to figure out the attributes weights and the class label is taken from system as an output. The training process in neural network continues so much until the acceptable accuracy is attained. Afterward, eliminate the input nodes (record attributes) one by one and calculate the error rate. So, the susceptibility to each input attributes like x_i can be attained. Each attribute, which is more affected in distinguish class label, has more susceptibility and finally more weighted.

3. THE KNNBA ALGORITHM

In this part, at first we discussed briefly about the association rules and algorithms witch mined this rules from datasets. Then, we suggest a novel approach for calculating weight of attributes by use of particular kind of association rules to increase the accuracy of k-nearest-neighbor algorithm and suggest the KNNBA algorithm to get more accurate classification of data.

3.1. MINING ASSOCIATION RULES

One way of data analysis is the mining repeatable patterns among the data. In this method, available patterns in dataset are drawn out among dataset as the rules [3]. For example, in database of a shop which its aim is to mine the buying pattern of customers based on their attributes, the following rule as one of the association rules can be mined.

Age = young AND income = high → buy (computer)

This rule represents that if someone is young and has the high income, then he or she buys a computer from the shop. Now the quality of this rule should be considered by two parameters, support and confidence. The support of the following rules represents the percentage of customers who were young and had high income and bought a computer. The confidence of the following rule represents that what percentage of young customers with high income bought computer. Generally, if we have an association rule like $A \rightarrow B$, then its two parameters, support and confidence, is identified based on probability theory in accordance with formula 4.

$$Sup = P(A \cup B)$$
$$Conf = P(A|B) \quad (4)$$

, where $P(A \cup B)$ indicates the probability that a record contains both A and B. $P(A|B)$ is conditional probability, that is indicates the probability that a record containing A also contains B.

3.2. KNNBA ALGORITHM

In this part we use just a particular kind of association rules which in its left side, there is just one item and on the right side of it, there is an item that can predict the class label. Now grouping these association rules after mining them. All the rules witch their left side includes the items related to an attribute, are put in one group, for example, all the rules like $Att_1 = V_i \rightarrow class\_Label = l$ witch is related to the first attribute of record put in one group. Thus the group i includes the rules related to the ith attribute.

Now defining two parameters, group support (G_Sup) and group confidence (G_Conf), for each group. Group support for each group is defined as greatest support of available items in the left side of rules in that group. Group confidence for each group is defined as greatest confidence of available rules in that group.

The major idea in giving weight to the attributes is that, which attributes have the repeatable values (G_Sup) and on the other hand, how affected these repeatable values can be to distinguish the class label (G_Conf). If an attribute having a large number of values then G_Sup of this attribute is

Algorithm: KNNBA (k-nearest-neighbors-based-association).
Input:
D is a set of *d* training records. Each record like R_i has n+1 attributes like (Att_{1i}, Att_{2i},…., Att_{ni} , class label);
T is a new record for prediction of its class label and has n+1 attributes like (Att_1, Att_2… Att_n, ?);
K is the number of neighbors;
G_Sup_Array is an array of group supports of all attributes;
G_Conf_Array is an array of group confidence of all attributes;
MinG_Sup is threshold of group support;
MinG_Conf is threshold of group confidence;
Output:
A class label for new record T;
Method:

```
for j=1 to n
{
        if(G_Sup_Array[j] <= MinG_Sup  OR G_Conf_Array[j] <=MinG_Conf )
                w[J] = 0;
else
        w[J] = (1 / (1 - G_Sup_Array[j]));
}
Min-Max-Normalization-Attributes();
For each (Record Ri in D)
{
        Distance = 0;
        For j=1 to n
                Distance = Distance + (w[j]*(Attj - Attji)^2);
        Distance = sqrt(Distance);
}
Get K records with minimum distance as neighbors of new record;
Return (predicted class label from K neighbors used weighted voting);
```

Fig 1: Pseudo code for KNNBA

small and in this method suppose that this attribute don't effective in classification. On the other hand if an attribute having a small number of values (large G_Sup) but this repeating of values don't effect on determining class label of records or in other words records with same value for this attribute having different class labels and value of this attribute don't effect on class label of records then G_Conf of this attribute is small and in this method suppose that this attribute don't effective in classification so. Therefore attributes with small G_Sup or small G_Conf are irrelevant attributes for classification.
Hence we define two separate thresholds for group support and group confidence in each dataset. Some attributes have weight equal to zero; the group support or group confidence of these attributes are less than threshold of group support or group confidence ($w[i] = 0$); On the other hand, we suppose that between attributes with group confidence and group support greater than defined thresholds, attributes with more repeatable values are more important in classification process. Therefore in this method with the use of formula 5, give weight to these attributes witch is not given zero weight.

$$w[J] = \left(\frac{1}{1 - G\_Sup[j]}\right) \quad (5)$$

, where $w[J]$ is weight of jth attribute and $G\_Sup[j]$ is group support of jth attribute.
Thus, each attribute is given a defined weight, considering the value of group support and group confidence. Then based on the given weight to the

attributes and with the use of formula 6, the distance between two records is figured out.

$$x_1 = (x_{11}, x_{12}, ..., x_{1n})$$
$$x_2 = (x_{21}, x_{22}, ..., x_{2n}) \quad (6)$$
$$dist(x_1, x_2) = \sqrt{\sum_{i=1}^{n} w[i] * (x_{1i} - x_{2i})^2}$$

The attributes with the large ranges of values, have more effect on figuring out the distance between two records in compared with ones with the small ranges of values.

In this algorithm, the values of attributes are normalized by the min-max normalization before figuring out the distance between new record and training records as follows:

$$V' = \frac{V - \min_A}{\max_A - \min_A} (7)$$

Suppose that $\min_A$ and $\max_A$ are the minimum and maximum values of an attribute, *A*. Min-max normalization maps a value, *V*, of *A* to V' in the range [*0, 1*]. By this normalization, the effect of all attributes is put in the span and so, the effect of all attributes is equalized in figuring out the distance between two records.

In this algorithm, also, the method of weighted voting and formula2 are used to figure out the class label of new record through the k neighbor of that record. Figure 1 is the pseudo code of KNNBA algorithm.

4. EXPERIMENTAL RESULTS

In this part, the suggested algorithm in this research KNNBA is compared with some other classification algorithms regarding the classification accuracy. To do so, the suggested algorithm is compared with other seven classification algorithms throughout the 15 datasets of the UCI repository [1] described in table1.

The compared algorithms are as following:

1. The naïve bayes (NB) [10];
2.The multilayer perceptron witch is the version of neural network with back propagation (NN);
3. The J48 [11] or C4.4, witch is the version of C4.5 with laplace correction and without tree pruning;
4. The NBTREE [12] witch implement the decision tree and then, it uses the naïve bayes to classify in the leafs;
5. The voting feature intervals (VFI) [13] witch is doing the classification based on giving weight to the attributes;
6. The locally weighted learning (LWL) [14] witch is a lazy classification algorithm based on giving weight to the new record neighbors;
7. The IBK [15] which is the version of KNN algorithm.

It is to say that all of this seven classification algorithms implemented in WEKA3.4.12 [16].

Table1. Description of the data sets used in the experiments

	Dataset	Size	No. of attributes	No. of classes
1	Balance	625	5	3
2	Breast-cancer	569	32	2
3	Breast-cancer-w	699	11	2
4	Ecoli	336	9	8
5	Glass	214	10	6
6	Haberman	306	4	2
7	Hayes	132	6	3
8	Heart-statlog	270	14	2
9	Labor	57	17	2
10	Parkinson	195	23	2
11	Teaching - Assistant	151	6	3
12	Vehicle	846	19	4
13	Wine	178	14	3
14	Yeast	1484	10	10
15	Zoo	101	18	7

All algorithms are executed in the following conditions:

1) The method ten-fold cross validation is used to execute all algorithms (the default of WEKA).
2) The default amounts of parameters witch is given by WEKA are observed for all of the algorithms. It must be mentioned that the WEKA normalized and discretized the attributes values automatically and in a default way in the most algorithms.
3) The number of neighbors are 10 (k=10) for the algorithms KNNBA, LWL and IBK.
4) The values of parameters, group support threshold and group confidence threshold, in the KNNBA algorithm for 15 datasets are noted in table 2.

In all experiments, the accuracy of each algorithm was based on the percentage of correct classified records. The accuracy of executing the mentioned algorithms on the suggested datasets is noted in the table3.

According to the table 3, the following results can be drawn out:

Table2. The parameters of KNNBA algorithm to perform on 15 databases of UCI

Dataset	group support threshold	group confidence threshold
Balance	0.1	0.5
Breast-cancer	0.006	0.6
Breast-cancer-w	0.2	0.6
Ecoli	0.01	0.49
Glass	0.019	0.3
Haberman	0.1	0.87
Hayes	0.1	0.5
Heart-statlog	0.04	0.5
Labor	0.07	0.5
Parkinson	0.01	0.6
Teaching - Assistant	0.01	0.5
Vehicle	0.02	0.49
Wine	0.01	0.5
Yeast	0.01	0.3
Zoo	0.1	0.5

Table3. Comparison of accuracy of KNNBA with other algorithms on 15 UCI datasets

	Dataset	**NB**	**NN**	**J48**	**NBTREE**	**VFI**	**LWL**	**IBK**	**KNNBA**
1	Balance	90.40	90.72	76.64	76.64	71.52	87.68	90.08	89.19
2	Breast-cancer	92.62	95.78	92.97	92.79	91.56	92.27	97.01	96.25
3	Breast-cancer-w	96.42	95.28	94.56	96.28	95.71	90.56	96.42	96.67
4	Ecoli	85.42	86.01	84.23	82.14	74.40	64.58	86.01	87.88
5	Glass	48.60	67.76	66.82	70.56	54.67	69.62	66.36	68.57
6	Haberman	74.84	72.88	71.90	72.55	59.15	61.76	73.20	73.33
7	Hayes	76.52	71.97	81.06	69.70	62.12	68.18	44.70	79.87
8	Heart-statlog	83.70	78.15	76.67	80.37	80.00	71.85	81.48	81.48
9	Labor	89.47	85.96	73.68	87.72	84.21	85.96	91.23	92.00
10	Parkinson	69.23	91.28	80.51	85.13	73.33	91.28	89.23	92.63
11	Teaching -Assistant	52.98	54.30	59.60	58.28	43.05	62.91	50.33	66.67
12	Vehicle	44.80	81.68	72.46	72.93	53.90	46.69	70.21	71.79
13	Wine	96.63	97.19	93.82	96.63	95.51	89.33	95.51	97.06
14	Yeast	58.80	61.50	75.15	65.37	53.57	40.69	62.61	95.91
15	Zoo	95.05	96.04	92.08	94.06	94.06	86.14	88.12	98.00
	Áverage	77.03	81.77	79.48	80.08	72.45	73.97	78.83	85.82

1) In comparison with the all other algorithms, the classification accuracy average of suggested algorithm KNNBA has a significant improvement. This improvement is between 4% (with NN algorithm) and 13% (with VFI algorithm) in comparison with the other algorithms.

2) KNNBA outperforms the traditional KNN algorithm (IBK) significantly. From our experiments, compared to KNN (IBK), KNNBA wins in 13 data sets, and loses in only 2 data sets with very small difference (less than 1%). The accuracy average of KNNBA with regard to IBK indicates almost 7% improvement. This improvement in some datasets is very more, for example in Hayes dataset this improvement is more than 35%.

3) Compared to seven other classification algorithms, KNNBA wins in 7 data sets, ties in 8 data sets and never loses.

4) In all the datasets, the suggested algorithm has the better results than the VFI algorithm witch is one of the algorithms gives weight to the attributes. Classification accuracy average is improved more than 13%, also in some datasets like Yeast this improvement is more than 42%.

5. CONCLUSION

In this article, we considered about a new classification algorithm KNNBA witch uses the association rules in the KNN algorithm in order to increase the classification accuracy of the KNN algorithm. In the KNNBA algorithm, each of attributes is given weight by mining association rules. In the KNNBA, each attributes that is more weighted, has more effect on figuring out the distance between two records and each attribute is less weighted has less effect on figuring out the distance. The practical test showed that the suggested algorithm has the more accuracy than the algorithms NB, NN, C4.4, NBTREE, VFI, LWL and IBK.

REFRENCES:

[1] A. Asuncion and D.J. Newman: "UCI Machine Learning Repository", Irvine, CA: University of California, School of Information and Computer Science, 2007. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html

[2] Cover and Hart, "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory*, 1967.

[3] J. Han and M. Kamber, "Data Mining Concepts and Techniques", 2nd ed. Amsterdam: Morgan Kaufmann Publishers, 2006.

[4] D. T. LAROSE, "Discovering knowledge in data: an introduction to data mining", New Jersey: John Wiley & Sons, 2005.

[5] Y. Zhan, H. Chen and G. C. Zhang," An Optimization Algorithm Of K-NN Classification", *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics*, Dalian, 13-16 August 2006.

[6] Md. Shamsul Huda, K. Md. R. Alam , K. Mutsuddi, Md. K. S. Rahman and C. M. Rahman," A Dynamic K-Nearest Neighbor Algorithm For Pattern Analysis Problem",*3rd International Conference on Electrical & Computer Engineering*, ICECE 2004, Dhaka, Bangladesh, 28-30 December 2004.

[7] K. Kumar Han, "Text categorization using weight adjusted k-nearest neighbor classification", *Technical report*, Dept. of CS, University of Minnesota, 1999.

[8] L. Jiang, H. Zhang and Z. Cai," Dynamic K-Nearest-Neighbor Naive Bayes with Attribute Weighted", *FSKD 2006*, LNAI 4223, 2006, pp. 365–368.

[9] J. Dongchao, S. Bifeng and H. Fei,"An Improved KNN Algorithm of Intelligent Built-in Test", *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, ICNSC 2008, Hainan, China, 6-8 April2008.

[10] H. George, J. Langley and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers", Proceedings *of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Mateo,1995, pp. 338-345.

[11] R. Quinlan, "C4.5: Programs for Machine Learning", *Morgan Kaufmann Publishers*, San Mateo, CA, 1993.

[12] R. Kohavi, "Scaling up the accuracy of naive-Bayes classifiers: a decision tree hybrid", *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[13] G. Demiroz and A. Guvenir, "Classification by voting feature intervals", *ECML-97*,1997.

[14] E. Frank, M. Hall and B. Pfahringer, "Locally Weighted Naive Bayes", *Working Paper 04/03*, Department of Computer Science, University of Waikato, Atkeson, 2003.

[15] D. Aha and D. Kibler, "Instance-based learning algorithms", *Machine Learning*, vol.6, 1991, pp. 37-66.

[16] Weka---Machine Learning Software in Java, 2008. [Online]. Available: **http://sourceforge.net/projects/weka/**