ISSN 2250-3153

A Novel and Efficient KNN using Modified Apriori Algorithm

Ritika Agarwal, Dr. Barjesh Kochar, Deepesh Srivastava

***Abstract*-** In the field of data mining, classification and association set rules are two of very important techniques to find out new patterns. K-nearest neighbor and apriori algorithm are most usable methods of classification and association set rules respectively. However, individually they face few challenges, such as, time utilization and inefficiency for very large databases. The current paper attempts to use both the methods hand in hand. Here, we have modified the apriori algorithm and used it to classify data for K-nearest neighbor. Modified Apriori helps in finding out only a few of the attributes that mainly define the class. These attributes are named as prominent attributes in this paper. This technique helps in improving the efficiency of KNN to a high extent.

***Index Terms*-** Classification, association set rules, K-nearest neighbor, apriori algorithm

I. INTRODUCTION

The techniques of data mining [1][2][3][20], in the field of classification [14] and association set rules [6], i.e., KNN and Apriori algorithm had been developed many years ago. Till today, they are used to extract knowledge and draw patterns from large sets of information.

The Apriori algorithm [5][7] is an unsupervised learning technique that figures out the likelihood of A taking place if B does. It is very useful in prediction regarding supermarkets and businesses.

While KNN[8][9] on the other hand is supervised learning and is used to gather information of the location of the test samples in the entire database. Its applications lie in robotics, pattern matching and graphics.

Both KNN and Apriori are individually efficient but together they are even better.

A. *Association Set Rules*

[2][3][6]It is a technique in data mining that is used to implement the apriori algorithm.

Three measures are used to determine the rules. They are:

1. Support: It describes the dominance of an attribute.
2. Confidence: It describes the relationship between two attributes.
3. Lift: It is the ratio of support between two independent items.

Using these two measures, the rules are formatted. They are:

1. The frequent combination of attributes.
2. The results based on combination.

The process of association set rules is used in the market basket analysis and in businesses.

1) *Apriori Algorithm,*

[4][5][7]It requires numeric values of each tuple to be presented for the formation of relationships.

Minimum support and minimum confidence are found and then support for each attribute is compared with it.

This algorithm is mainly divided into two processes:

1. Scanning of the database to calculate each attribute's support count and then rejecting the attributes with lesser values.
2. Pruning of candidate sets to remove the candidates without frequent subsets.

The method devised by the apriori algorithm faced severe problems of space and time.

B. *Classification*

[1][2][3][14] It is a method to find out equations from the given data that can be used to place and understand the test samples. Here, the previously unknown tuples need to be assigned a class. The already known data is utilized to complete this task. The classifiers are broadly divided into the categories of early and lazy learners.

Early learner is the one that keeps the data ready before the tuples arrive. Such as decision trees, SVM, etc. On the other hand lazy learners prepare their model last minute. Eg. KNN, etc.

1) K-nearest neighbor,

[4][8][9][12]It is a process based on learning by comparison. KNN follows the principles of classification. The closeness of tuples is measured with the test tuple. The closest K tuples are called the test tuples's neighbors. Usually Euclidean distance is used to find out the closeness. Other distance metrics may also be used. They can be Mahalanobis, Chebychev, Correlation and many more. Choice of the distance metric can be an issue. The search of neighbors is started by giving k, a value equal to 1. [15][16][17] It generates the minimum error. Later the value is increased and it may reach infinity for large data sets. KNN has poor speed and accuracy.

II. MODIFIED APRIORI

The previous algorithm had a major problem of multiple scans through the entire data. It required a lot of space and time. [10]

The modification in our paper suggests that we do not scan the whole database to count the support for every attribute. This is possible by keeping the count of minimum support and then comparing it with the support of every attribute. The support of an attribute is counted only till the time it reaches the minimum support value. Beyond that the support for an attribute need not be known.

This provision is possible by using a variable named flag in the algorithm. As soon as flag changes its value, the loop is broken and the value for support is noted.

The pseudo code for the proposed algorithm is as follows:

```
Input :  Database,  D, of transactions;
         Minimum support threshold, min_sup
Output : L, frequent itemsets in D
Method :
    1)  L(1)= find_frequent _1-itemsets(D);
    2)  For each transaction t belongs to D
    3)       count_items= count_items(t);
    4)  For (k=2; L(k-1)!=null; k++)
    5) {
    6)  C(k)= apriori_gen(L(k-1, min_sup);
    7)  flag=1;
    8)  For each transaction t belonging to D
          Where count_items>=k
    9)  {
   10)  If  (flag==1)
   11)   {
   12)      c=subset(C(k),t);
   13)      c.count++;
   14)     if (c.count==min_sup)
   15)        flag=0;
   16)   }
   17)    if (flag==0)
   18)    Exit from loop
   19)  }
   20)  L(k)={c.count=min_sup}
   21) }
   22) return L=U(k) L(k);
```

This new feature added in apriori algorithm helps us to implement it in the KNN algorithm, where the support count for every attribute is found out separately for each class.

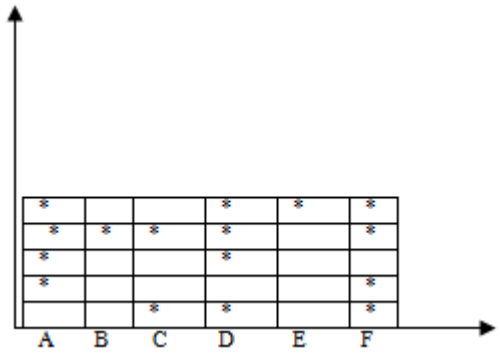
III. IMPROVED KNN

[4][411][13][16]In KNN we tried to find out the nearest neighbor by calculating the distance between the test sample and the training samples. In KNN, the samples with lesser distance are taken as neighbors. But there are possibilities that attributes with large domain and range have predominance over the other attributes and thus this leads to a wrong result, i.e. the test sample may be placed in a wrong class.

Now, we have devised that only a few attributes can be considered to check the distance between the test and trained samples, rather than all of the attributes. But, the attributes selected have to be the most appropriate ones. By appropriate we mean that they have to be sufficient for determining the class of the test sample.

Here we are trying to find out the prominent attributes of every class, using the apriori algorithm. The benefit of these prominent attributes is that only they are required to be checked with the test samples for finding their neighbors because they are the attributes that decide the class of an attribute. As usual, we are using the Euclidean distance metric to find out the distance between the training samples and the test samples.

This makes the KNN easier because now we do not have to check the distance of our test samples with every attribute of the training samples. Also, the problem of the predominance of some of the attributes due to larger values gets eradicated.
Given below is the diagram that shows the idea of the new KNN.



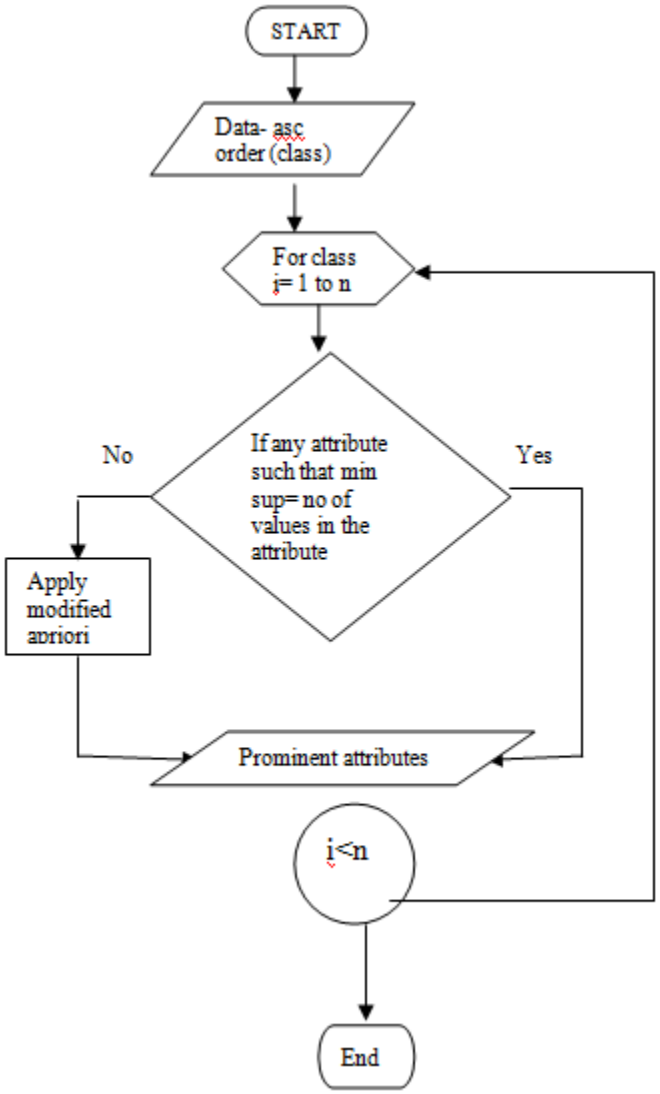
The above diagram represents:

X-axis - attributes
Y=axis - training samples

The rows represent the different tuples. This figure represents data for only a particular class.

Here we can clearly see that the attributes A, D, and F decide the occurrence of a sample in the class. At least two out of these three attributes is always true for every sample. Thus we can say that according to this diagram that we only need to check with these three attributes to get to know the neighbors for our training samples.
Now, to find these prominent attributes, an apriori algorithm is used [11].

The following is the flow chart of the new method:



1. The data is converted to numeric values between 0 to 1 using [18]
   new=(old-a)/b-a
   b=old max value
   a=old min value
   new= the new value
   old=the value that needs to be converted
2. Then it is placed according to the ascending order of the classes.
3. For each class individually,
4. The formulae (for discrete values only)
   Min_sup =no. of tuples is applied

[It can be noted here that values that are not discrete would have to take
   Min_sup = tuples*max value for an attribute in that class]

5. If no attribute clears it, modified apriori is applied.
6. Prominent attributes are found out.
7. KNN is applied on those attributes only.

IV. IMPLEMENTATION

[19]The research has been applied to a dataset. A few values are shown to understand the methodology.

Dataset	Zoo
Number of attributes	18
Number of classes	07
Size	101

The table below shows few of the values taken from the database zoo.

Class 1	aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruit bat, giraffe, girl, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, Oryx, platypus, polecat, pony, porpoise, puma, pussycat, raccoon, reindeer, seal, sea lion, squirrel, vampire, vole, wallaby, wolf
Class 2	chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren
Class 3	pit viper, sea snake, slowworm, tortoise, tuatara
Class 4	bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna
Class 5	frog, frog, newt, toad
Class 6	flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp
Class 7	clam, crab, crayfish, lobster, octopus scorpion, sea wasp, slug, starfish, worm

There are total 18 attributes. The first attribute is the name of the animals. Thus, it is not considered. Here, the values A-* signify the attributes. The attributes are:

A- Hair
B- Features
C- Egg
D- Milk
E- Airborne
F- Aquatic
G- Predator
H- Toothed
I- Backbone
J- Breathes
K- Venomous
L- Fins
M- Legs
N- Tail
O- Domestic
P- Cat size
*- Class

The training samples 1-6 are of the animals:

1- Aardvark
2- Antelope
3- Buffalo
4- Scorpion
5- Worm
6- Starfish

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	*
1	1	0	0	1	0	0	1	1	1	1	0	0	1	0	0	1	1
2	1	0	0	1	0	0	0	1	1	1	0	0	1	0	0	1	1
3	1	0	0	1	0	0	0	1	1	1	0	0	1	0	0	1	1
4	0	0	0	0	0	0	1	0	0	1	1	0	1	1	0	0	7
5	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	7
6	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	7

There are 2 classes.
For class 1:
Min support=3
The support for each attribute is as follows:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
3	0	0	3	0	0	1	3	3	3	0	0	3	0	0	3

Thus for adding new samples into class 1,we have to only check for the attributes A, D, H, I. J, M and P.

For class 7:
Min support=3
The support for each attribute is as follows:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
0	0	2	0	0	1	2	0	0	2	1	0	2	1	0	0

Here, none of the attributes has support equal to the minimum support. Thus, we apply modified apriori.
For that, C1 (list of attributes) and L1 (their respective support value) have to be established first.
Now, new min sup=1(value is changed for applying apriori algorithm)

C1	L1
C	1
F	1
G	1
J	1
K	1
M	1
N	1

Now all those attributes that have support equal to Min support are taken into pairs. They form C2 and L2.'

C2	L2
C,F	1
C,G	1
C,J	1
C,M	1
F,G	1
F,M	1
G,J	1
G,K	1
G,M	1
G,N	1
J,K	1
J,M	1
J,N	1
K,M	1
K,N	1
M,N	1

The pairs with valid support are grouped into three at a time for C3 and corresponding support forms L3.

C3	L3
C,F,G	1
C,F,M	1
C,G,M	1
G,J,K	1
G,J,M	1
G,J,N	1
G,K,M	1
G,K,N	1
G,M,N	1
J,K,M	1
K,M,N	1
F,G,N	1

A group of four is found out from C3, and then C4 is created.

C4	L4
C,F,G,M	1
J,K,M,N	1

Thus only these two prominent groups of variables are required to be checked with the test samples.

V. RESULT AND DISCUSSION

The earliest KNN used the distance metric directly and thus ended up into wrong classes a number of times.
Then KNNBA used association rules to provide weights to the attributes. It increased the efficiency up to 40% but there was still scope for improvement.
Our new KNN takes up only the prominent attributes to check the distance between the training samples and the test samples.

To summarize our KNN, we can generalize it using the following functions:
Here,

X= support of an attribute
Y=no. of tuples
A= attribute

1) For new KNN

$$F[NKNN(x,y)] = \begin{cases} x = y; prominent\ member \\ x < y; modified\ apriori \end{cases}$$

This function explains that it is required to implement the modified apriori algorithm on all those attributes whose values are not true (maximum) in all the tuples.
2) For new Apriori

$$F[ma(x)] = \begin{cases} x = \min support; attribute\ taken \\ x < \min support; attribute\ neglected \end{cases}$$

This function defines the modified apriori algorithm that states that all the attributes with support value equal to minimum support are accepted and all the others are rejected.

3) For new KNN using Modified Apriori

$$F[NKKN[MA(a,x,y)]] = \begin{cases} x = y\ or\ x = \min support; a = prominent\ member \\ else\ not \end{cases}$$

This function explains the novel KNN using modified Apriroi algorithm. It tells us that if an attribute clears the apriori algorithm or if it has all values of an attribute as maximum, it is considered as a prominent attribute

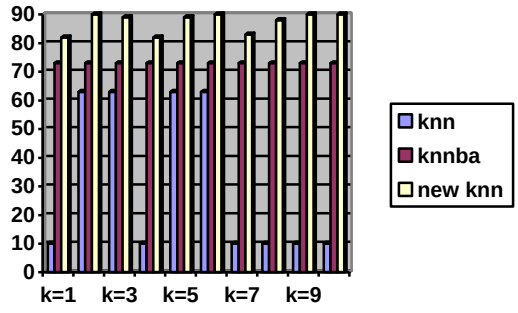
[21]The graph below shows the accuracy gained by the three methods KNN, KNNBA and NKNN when applied on same data.

Accuracy = no of correct placement

Total placements of test samples

K	KNN	KNNBA	NKNN
1	10	73	82
2	63	73	90
3	63	73	89
4	10	73	82
5	63	73	89
6	63	73	90
7	10	73	83
8	10	73	88
9	10	73	90
10	10	73	90

The above table represents the accuracy of KNN, KNN-BA, and novel KNN. It must be noted that different values of minimum support and confidence are required for KNN-BA. Out of them, the values yielding the best solution are taken. [11][13]



Thus, it can be seen that the efficiency of KNN has increased drastically.

VI. CONCLUSION

In this paper we have attempted to give a new perspective to the KNN classifier with the eye of a modified apriori algorithm. This algorithm is better than both of the previous methods, i.e., [8]KNN and [11][13]KNNBA. This method works perfectly for data that has been supervised, i.e., data whose classes are already known. But if the classes are not known already, then we can first take any attributes as prominent attributes and test them for modified apriori. Also, the data taken in this example is discrete and this algorithm works on numeric data.

REFERENCES

[1] Han, Jiawei and Kamber, Micheline, Data Mining Concepts and Techniques. Morgan Kaufman Publishers. San Fransisco 2000.

[2] Han, David, et al. Principles of Data Mining: MIT press. Cambridge, 2001.

[3] G.K. Gupta,Introduction to data mining with case studies:Prentics Hall of India, New Delhi, 2006

[4] Top 10 algorithms in data mining, Xindong Wu, Springer-2007

[5] Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th VLDB conference, pp 487–499

[6] Mining Association Rules between Sets of Items in Large Databases: Rakesh Agrawal ,Tomasz Imielinski,Arun SwamiACM SIGMOD ConferenceWashington DC, USA, May 1993

[7] Fast Algorithms for Mining Association Rules: Rakesh Agrawal Ramakrishnan Srikant VLDB Conference Santiago, Chile, 1994

[8] High Performance Data Mining Using the Nearest Neighbor Join Christian Böhm Florian Krebs

[9] A Review of various k-Nearest Neighbor Query Processing Techniques : International Journal of Computer Applications (0975 – 8887) Volume 31– No.7, October 2011

[10] Mining of Meteorological Data Using Modified Apriori Algorithm, European Journal of Scientific Research ISSN 1450-216X Vol.47 No.2 (2010), pp.295-308EuroJournals Publishing, Inc. 2010 http://www.eurojournals.com/ejsr.htm

[11] Lailil Muflikhah1, Classifying Categorical Data Using Modified K-Nearest Neighbor Weighted by Association Rules, 2011 International Conference on Future Information Technology IPCSIT vol.13 (2011) © (2011) IACSIT Press, Singapore

[12] Bailey, T., Jain, A. A note on distance-weighted k-nearest neighbor rules. IEEE Trans. Systems, Man, Cybernetics, Vol. 8, pp. 311-313,1978.

[13] KNNBA: K-Nearest-Neighbor-Based-Association Algorithm. 1Mehdi Moradian, 2Ahmad Baraani, Journal of Theoretical and Applied Information Technology,2009

[14] Thair Nu Phyu Survey of Classification Techniques in Data Mining, Survey of Classification Techniques in Data Mining, Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009 Vol I

[15] http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[16] Kozak K, M. Kozak, K. Stapor ,Weighted k-Nearest-Neighbor Techniques for High Throughput Screening Data ,International Journal of Biological and Life Sciences 1:3 2005

[17] http://www.mathworks.in/help/toolbox/stats/bsehyju-1.html

[18] http://www.d.umn.edu/~deoka001/WKNN.html

[19] http://archive.ics.uci.edu/ml/machine-learning-databases

[20] Ulmer, David. Mining an Online Auctions Data Warehouse, 2002. (http://csis.pace.edu/csis/msplas/p8.pdf)

[21] Wettschereck D, Aha D,Mohri T (1997) A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artif Intell Rev 11:273–314

AUTHORS

First Author: Ritika Agarwal Research Scholar, Email id - ritika.agarwal49@yahoo.com

Second Author: Dr. Barjesh Kochar, Dean, Guru Nanak Institute of Management, Email id - bkkochar@gmail.com

Third Author: Assistant Professor, Amity University, Email id - deepeshid@gmail.com