CHAPTER 4

NAD - A DISCRETIZATION PROCESS

4.1 PROBLEM STATEMENT

Some of the mining/learning algorithms do not perform well for large amount of data. Those algorithms need continuous features to be discretized for reducing the size of the datasets. A new algorithm NAD (*Normalized And Discretization method*) is proposed for improving the interdependency between continuous and class features. C4.5 and Naïve Bayes classifiers are used for classification process. Empirical comparison of existing algorithms with the proposed algorithm shows that the proposed algorithm gives better result in terms of Computational time and accuracy.

4.2 INTRODUCTION

While extracting useful and hidden knowledge from large dataset, Data Mining algorithm often have to handle both continuous and nominal features. Some classification algorithms in Data Mining such as ID3 (Quinlan 1986) and CN2 (Clark and Niblett 1989) can handle only nominal features, while others such as C4.5 (Quinlan 1994) and CART (Breiman et al. 1984) can handle continuous features also. To handle continuous features, discretization methods are used. Discretization is the process of grouping values of continuous features into finite range of intervals and a unique value is assigned to each range of values (Agre and Peev 2002).

Discretization

Discretization as given by Liu (1997), is a technique to group the values of continuous attributes into a finite set of intervals and to assign each interval with a distinct values. Intervals are considered as single discrete value. Discretization can make mining easier and helps to improve the interdependencies between the continuous attributes and the class attribute.

Definition

Assuming that a dataset consists of *N* instances and *S* target classes, a Discretization algorithm would discretize the continuous attribute *F* in the dataset into *n* discrete intervals $\{[d_0,d_1],[d_1,d_2],....(d_{n-1},d_n]\}$, where d_0 is the minimal value and d_n is the maximal value of continuous attribute *F*. Such a discrete result $\{[d_0,d_1],[d_1,d_2],....(d_{n-1},d_n]\}$ is called a Discretization scheme *D* on attribute *A* (Dougherty et. al. 1995).

Discretization methods are classified according to some criteria. Prieditis and Russell (Dougherty et al. 1995) suggested criteria like global (The continuous feature values are discretized before applying mining process) versus local (discretization is performed within the mining process), supervised (uses both continuous and class features) versus unsupervised (class features is not taken) and static (discretization applied to each features independently) versus dynamic (discretization applied to all features simultaneously). Other criteria (Pal et al. 2006) are direct (discretize the features within the determined k value) versus incremental (criteria applied to stop discretization), top-down (splitting from the initial interval) versus bottom-up (merging the intervals from lowest level) and two classes versus multi-class.

In recent years, researchers have developed discretization algorithms to improve the interdependency between continuous and class features. Many researchers have taken midpoint of each continuous-class feature value as initial boundary point. For each boundary point, interdependency value is calculated and the maximum value is taken as the boundary point. This process is repeated for *n-1* times, where *n* is the number of values in the dataset. This will increase the time taken to discretize the feature as the number of initial boundary points obtained is high if the midpoints are taken in between each continuous feature value.

In section 4.3, the proposed algorithm (NAD) for improving the interdependency between class feature and continuous features is given. In section 4.4, the experimental comparison of existing discretization algorithm with the proposed algorithm is done by using dataset taken from UCI ML repository. Finally, conclusion and future work is given in section 4.5.

4.3 PROPOSED CONCEPT

CADD, CAIM and the new method CACC use class-attribute interdependencies. These algorithms differ slightly in their framework. These algorithms find the midpoints between the two consecutive pair of values and initialize them as initial boundary points. This is illustrated by using a sample set of data *D*=((0,Y), (4,Y), (12,Y), (16,N), (16,N), (18,Y), (24,N), (26,N), (28,N)) where *D* denotes the 9 (feature value/class) pairs.

Let *B* be the boundary points. Then the initial boundary points obtained using CAIM (Kurgan and Cios 2004) algorithm are *B* = (2, 8, 14, 16,17, 21, 25, 27) which is repeated for *n-1* times if *n* is the number of instances.

In the proposed algorithm instead of choosing the midpoint between the two consecutive pair of values as boundary points, the midpoints where the two consecutive values have different class value can be taken as boundary points. Then the initial boundary points obtained using NAD algorithm are *B* = {14, 17, 21} which is repeated for less number of times when compared with CAIM algorithm. The following Table 4.3.1 shows the boundary points and the number of iterative looping for the above sample dataset.

Table 4.3.1 : Boundary Points and number of intervals

Algorithm	**Initial Boundary Points**	**Iterative looping**	**Final Boundary Points**	**# of intervals**
CAIM	2,8,14,16,17,21,25,27	8	14,27	3
CACC	2,8,14,16,17,21,25,27	8	14,27	3
NAD	14,17,21	3	14,21	3

Normalization

Removing redundant data from the dataset is the process of Normalization, by splitting existing set into multiple ones in order to enhance the storage efficiency, data integrity and scalability. Key concepts in normalization are functional dependency and transitive dependency.

- Functional dependency *FD* : X → Y means that if the attribute Y is dependent on attribute X, then Y is functionally dependent on X.
- If the attribute B is dependent on attribute A and attribute C is dependent on attribute B, then the process is called as Transitive dependency.

Many normalization algorithms have been proposed. Normalization must be applied, before discretizing each continuous feature. The dataset which contains only the continuous features can be used for discretization, instead of storing the entire dataset. This will reduce the time of accessing the features and also the storage capacity. As an example, in the *Lung cancer* dataset taken, transitive dependency occurs. Third normal form is used to remove the transitive dependency. The solution is that any transitive dependencies that occur must be moved into a smaller table. The features of the dataset are (class, structure, regularity, cavitations, area, scar, shape, sharpness, smoothness, lobularity, angularity, convergence, vascular shadows, pleura thickness, size, character, attenuation). After normalization, discretization can be applied on the table which contains continuous features only. The dataset containing size and attenuation features is memory-resident which will reduce the storage space.

Algorithm 4.1 : NAD algorithm

Let *D* be the entire dataset. Let the number of instances be *N* described by all features *F*, and *S* classes.

```
Step 1
        $D_1$ : Normalize(D,F).
        Let the dataset in each continuous features $F_{i\ be} D_1$.
        In every $F_i$ do:
Step 2
        2.1 find highest ($d_n$) and lowest ($d_o$) values
        2.2 sort all distinct values of $F_i$ in ascending order and initialize all possible interval boundaries, B, with the minimum, maximum and the midpoints where the continuous features have different classes in the set
        2.3 set the initial discretization scheme to $D_1$:[$d_o$,$d_n$], set variable GlobalCAIM=0
Step 3
        3.1 Let k=1
        3.2 for each interval boundary in B, calculate the CAIM value.
        3.3 Select the boundary having highest value of CAIM
        3.4 if (CAIM > GlobalCAIM or k < S) then update D1 with the accepted, in step 3.3, boundary and set the GlobalCAIM=CAIM, otherwise terminate
        3.5 set k=k+1 and go to 3.2
Result: Discretization scheme $D_1$
```

4.4 EXPERIMENTAL COMPARISON

This section deals with the empirical analysis of the following discretization methods using C4.5 and Naïve Bayes classifiers.

1. CAIM algorithm.
2. NAD algorithm.

The implementation of these algorithms is done using MATLAB 7.0.

The Dataset

The datasets were selected from the UCI ML repository (Hong and Yang 1992). Table 4.4.1 gives the details of the datasets with the number of continuous features, the number of instances and the number of class features.

Table 4.4.1 : Description of dataset

Datasets	Instances	Attributes
Diabetes	468	9
Iris	150	5
Breast Cancer	699	10
Lung Cancer	32	57

Empirical analysis

For breast cancer dataset, Figure 4.4.1(a) shows the initial boundary points obtained by implementing CAIM method and Figure 4.4.1(b) shows the initial boundary points obtained by implementing the proposed NAD method.

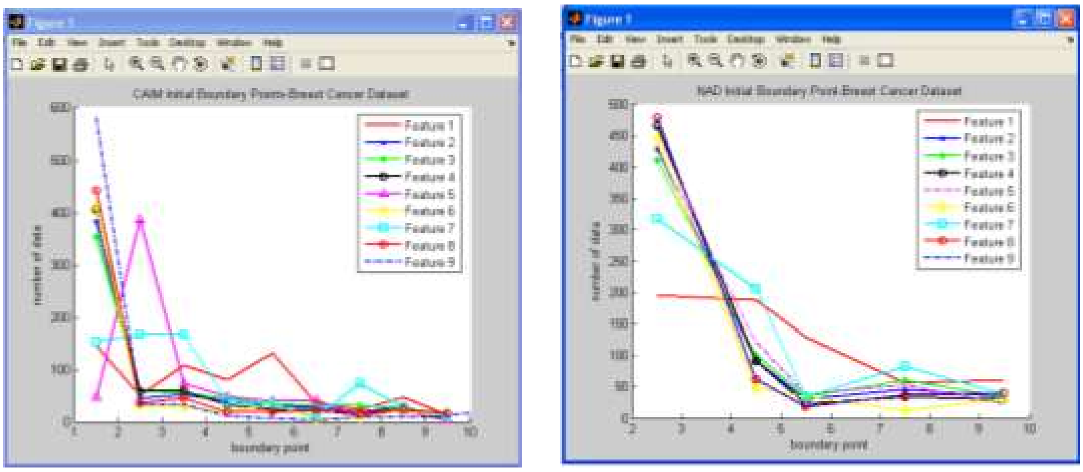

Figure 4.4.1(a): Initial boundary points – CAIM   Figure 4.4.1(b):Initial boundary points– NAD

Figure 4.4.1 (a-b) Initial Boundary points for Breast Cancer dataset

From the Figures 4.4.1(a) and 4.4.1(b), it is clear that the plotting of initial boundary points using NAD is close to each other for each attributes. This shows the improved interdependency between the attributes than the existing methods CAIM. Figure 4.4.1 (b) shows less number of cutting points (intervals generated) which reduces the iterative looping time.

For lung cancer dataset, both the CAIM and NAD yields the same number of initial boundary points as shown in the same Figure 4.4.2(a)-4.4.2(i).

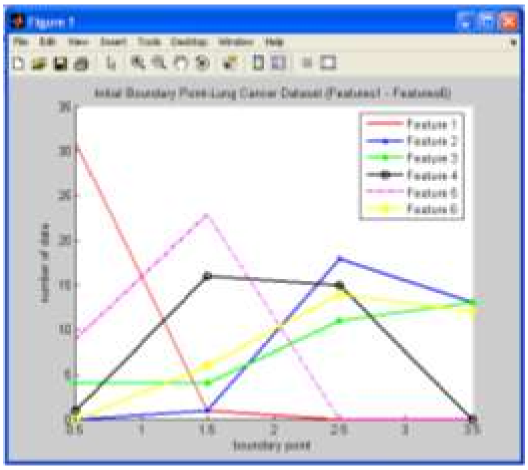

Figure 4.4.2(a) : Initial boundary points for features 1-6

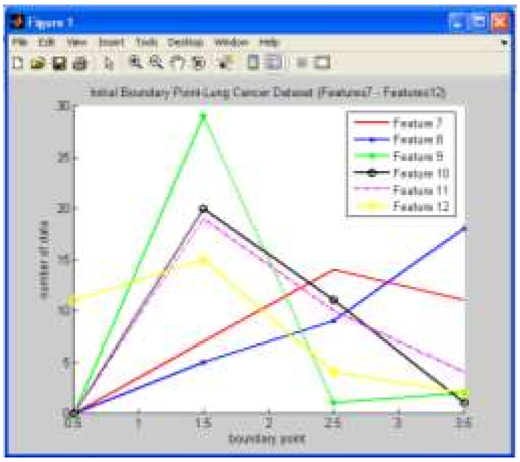

Figure 4.4.2(b) : Initial boundary points for features 7-12

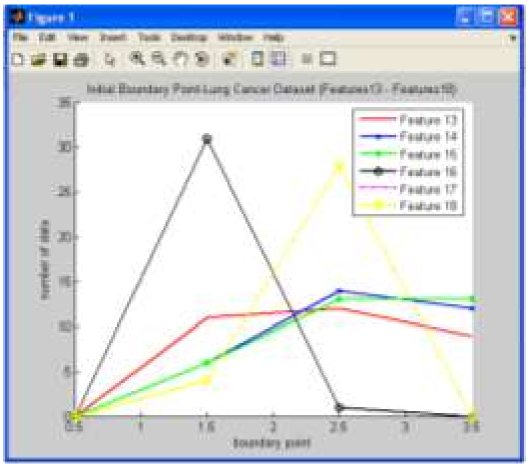

Figure 4.4.2(c) : Initial boundary points for features 13-18

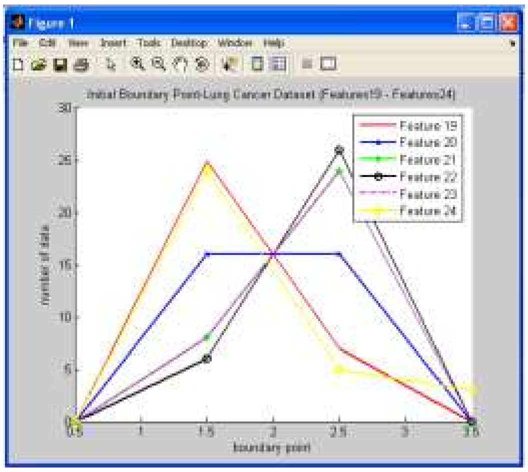

Figure 4.4.2(d) :Initial boundary points for features 19-24

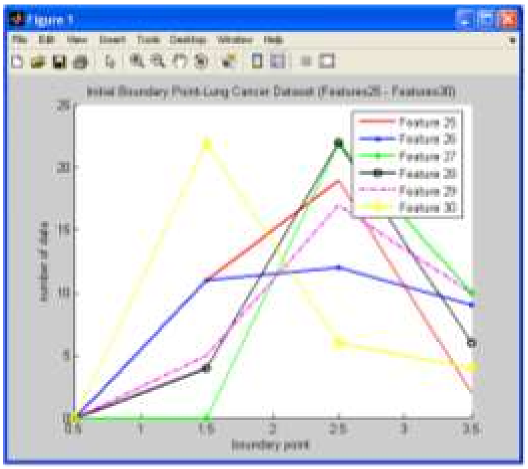

Figure 4.4.2(e) : Initial boundary points for features 25-30

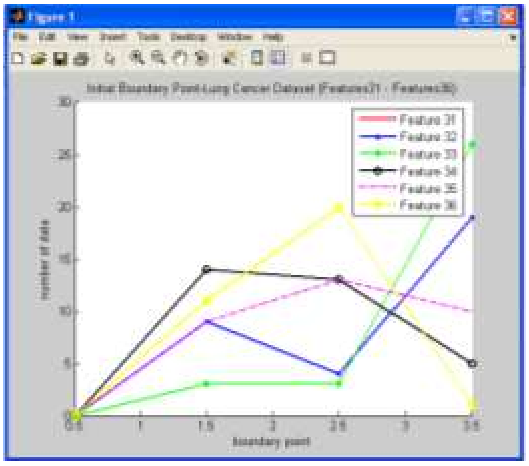

Figure 4.4.2(f) : Initial boundary points for features 31-36

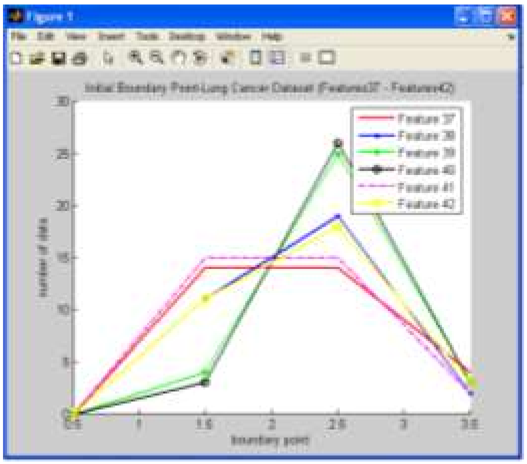

Figure 4.4.2(g) :Initial boundary points for features 37-42

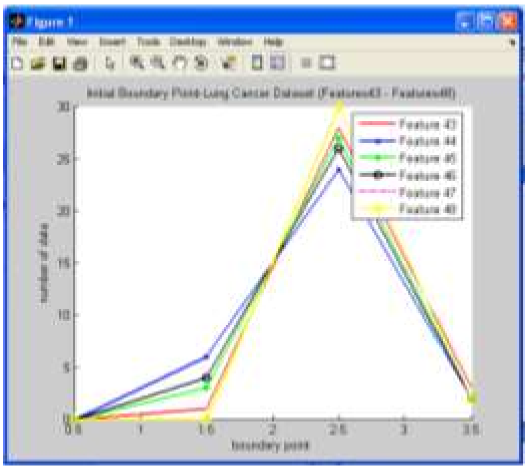

Figure 4.4.2(h):Initial boundary points for features 43-48

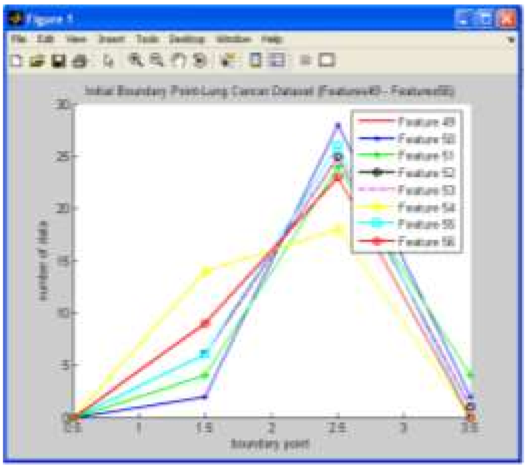

Figure 4.4.2(i) : Initial boundary points for features 49-56

Figure 4.4.2 (a-i) : Initial boundary points for Lung Cancer dataset

For Diabetes dataset, Figure 4.4.3(a) shows the initial boundary points obtained by implementing CAIM method and Figure 4.4.3(b) shows the initial boundary points obtained by implementing the proposed NAD method.

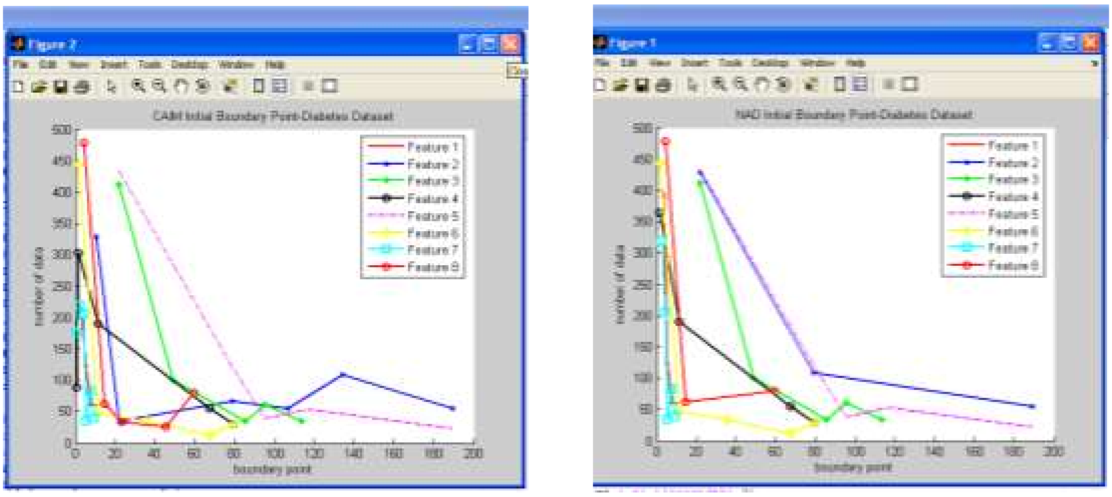

Figure 4.4.3(a): Initial boundary points – CAIM Figure 4.4.3(b):Initial boundary points– NAD

Figure 4.4.3 (a-b) : Initial boundary points for Diabetes dataset

From the result shown above, Figure 4.4.3 (b) shows that the NAD algorithm generates less number of cutting points (intervals generated) than CAIM algorithm shown in figure 4.4.3(a).

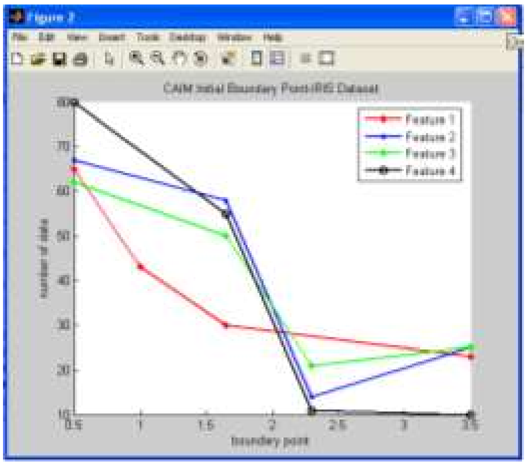

Figure 4.4.4(a): Initial boundary points – CAIM

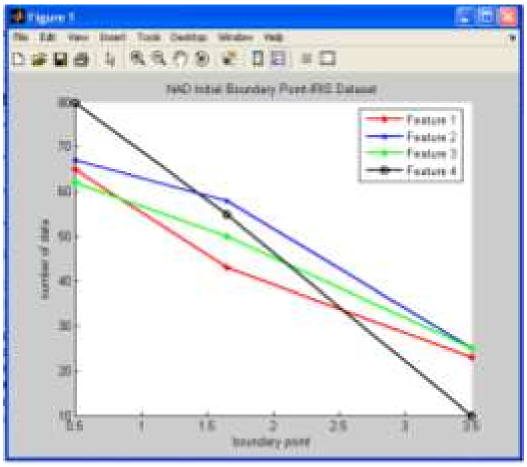

Figure 4.4.4(b):Initial boundary points– NAD

Figure 4.4.4 (a-b) : Initial boundary points for Iris dataset

From the result shown above, Figure 4.4.4 (b) depicts that the NAD algorithm produces less number of cutting points (intervals generated) than CAIM algorithm shown in figure 4.4.4(a).

If the CAIM value is high, then the interdependency between the class feature and the discrete intervals will be high. This is shown in the table 4.4.2 that NAD maximizes the CAIM value resulting in the higher interdependency between the class attribute and the continuous attribute for both the datasets.

Table 4.4.2 : CAIM criterion value

Methods	CAIM value-Breast Cancer dataset	CAIM value-Lung Cancer dataset	CAIM value-Diabetes dataset	CAIM value-Iris dataset
NAD	133.70	172.02	65.06	10.24
CAIM	123.14	171.99	64.75	11.607

Computational time

Let ‘*n*’ be the set of instances and ‘*p’* be the probability that a point be a boundary point and let ‘*s*’ be the set of classes. Then the time of the proposed algorithm is as follows:

Time taken to find the highest and lowest values is O(*n*). Sorting need O(log(*n*)). Time taken to find the initial boundary cut point is O(*p**n), but CAIM (Class Attribute Interdependency Maximization) takes *n-1* times which is O(*n*). The total computational time of the proposed algorithm (excluding the single unit of time for some statements) is O(*n*). The existing algorithm (CAIM) takes O(*n*logn*). This shows that the running time of the proposed method is faster than the CAIM method.

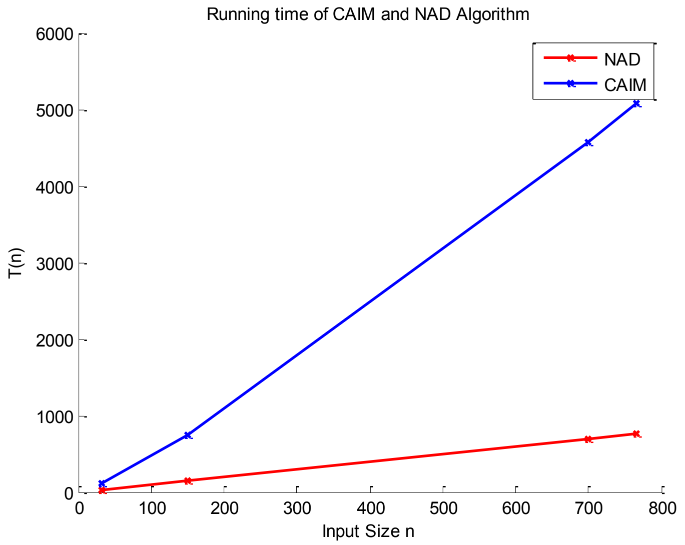

Figure 4.4.5 : Running time (in ms) of CAIM and NAD method for four dataset

Accuracy

Table 4.4.3 : Percentage of accuracy value in (%) using discretization methods

Dataset/ Methods	EWD		EFD		Entropy		CAIM		Proposed (NAD)	
	NB	C4.5	NB	C4.5	NB	C4.5	NB	C4.5	NB	C4.5
Breast Cancer	95.98	94.42	95.28	94.85	96.99	94.99	96.34	95.02	97.45	95.21
Lung Cancer	83.31	77.78	81.25	78.13	82.22	78.41	84	78.81	81.23	78.79
Diabetes	75.3	72.01	75	73.44	77.86	76.04	75	76.99	75.12	77.2
Iris	95.33	96	96	96	94	94	95.43	95.65	96.22	96.46

In the table 4.4.3, Naïve Bayes and C4.5 classifiers are compared on the data, discretized by the 5 discretization algorithms based on accuracy. From the above table, it is clear that the results for Naïve Bayes and C4.5 classifiers show that the accuracy is significantly improved for breast cancer dataset which has large number of instances. The proposed method works well for the multiclass dataset using both the classifiers. Also, the result shows that the supervised discretization methods are superior to the unsupervised discretization methods as these methods take class feature into account.

4.5. SUMMARY

Discretization plays an important role in classifying the large dataset which has both continuous and nominal features. Empirical comparison on four datasets using the classifiers gives better improvement in accuracy. The proposed algorithm yields less number of intervals than the existing algorithm. The proposed algorithm aims in reducing the computational time, for maximizing the interdependency between the continuous and class features and in reducing the storage capacity.