Journal of Computer Science 5 (7): 519-528, 2009
ISSN 1549-3636

Discretization Numerical Data for Relational Data with One-to-Many Relations

Rayner Alfred
School of Engineering and Information Technology, University Malaysia Sabah,
Locked Bag 2073, 88999, Kota Kinabalu, Sabah, Malaysia

Abstract: Problem statement: Handling numerical data stored in a relational database has been performed differently from handling those numerical data stored in a single table due to the multiple occurrences (one-to-many association) of an individual record in the non-target table and non-determinate relations between tables. Numbers in Multi-Relational Data Mining (MRDM) were often discretized, after considering the schema of the relational database. Study the effects of taking the one-to-many association issue into consideration in the process of discretizing continuous numbers. **Approach:** Different alternatives for dealing with continuous attributes in MRDM were considered in this study, namely equal-width (EWD), Equal-Height (EH), equal-weight (EWG) and Entropy-Based (EB). The discretization procedures considered in this study included algorithms that were not depended on the multi-relational structure of the data and also that are sensitive to this structure. A new method of discretization, called the entropy instance-based (EIB) discretization method was implemented and evaluated with respect to C4.5 on the two well-known multi-relational databases that include the Mutagenesis dataset and the Hepatitis dataset for Discovery Challenge PKDD 2005. **Results:** When the number of bins, b, is big (b = 8), the entropy-instance-based discretization method produced better data summarization results compared to the other discretization methods, in the mutagenesis dataset. In contrast, for the hepatitis dataset, the entropy-instance-based discretization method produced better data summarization results for all values of b, compared to the other discretization methods. In the Hepatitis dataset, all discretization methods produced higher average performance accuracy (%) for partitional clustering technique, compared to the hierarchical technique. **Conclusion:** These results demonstrated that entropy-based discretization can be improved by taking into consideration the multiple-instance problem. It was also found that the partitional clustering technique produced better performance accuracy compared to the one produced by hierarchical clustering technique.

Key words: Discretization, entropy-based, semi-supervised clustering, genetic algorithm, multiple instances

INTRODUCTION

Most multi-relational data mining deals with nominal or symbolic values, often in the context of structural or graph-based mining (e.g., ILP)[3]. Much less attention has been given to the area of discretization of continuous attributes in a relational database, where the issue of one-to-many association between records has to be taken into account. Continuous attributes in multi-relational data mining are seldom used due to the difficulties in handling them particularly when we have a one-to-many association in a relational database.

Handling numerical data stored in a relational database is different from handling those numerical data stored in a single table due to the multiple occurrences of an individual record stored in the non-target table and non-determinate relations between tables.

Firstly, most pre-processing steps, such as the discretization and aggregation operations, that process attributes stored in relational database, need to use the structure (schema) of the relational database and to find out how attributes stored in non-target and target tables are related to each other. One may perform the aggregation operation on the attributes that have numerical multi-set values and then perform the discretization operation on the aggregated value. However, this is not an easy task as the non-target table may have categorical and numerical attributes in the same table.

Next, the task of discretizing continuous attributes is more complex when the occurrences of multiple instances in the non-target table are taken into

consideration, since most traditional discretization methods deal with a single flat table and quite often ignore the one-to-many relationships problem.

And finally, using a class-based discretization method, such as an entropy-based discretization[17], is not a straight-forward task in a relational database as it needs to be done in a single table. Most traditional data mining methods only deal with a single table where all attributes are available in that table and discretize columns that contain aggregated continuous numbers into nominal values. In a relational database, multiple records with non-aggregated numerical attributes are stored in the non-target table, separately from the target table and these records are usually associated with a single individual stored in the target table. As a result, discretizing continuous attributes in non-target table based on the class information requires user to consider the structure of the relational database. Thus, numbers in relational databases are often discretized, after considering the schema of the relational database, in order to reduce the continuous domains to more manageable symbolic domains of low cardinality and the loss of precision is assumed to be acceptable.

Data transformation using Dynamic Aggregation of Relational Attributes (DARA): The DARA algorithm is designed to transform the data representation of a relational database into a vector space model, such that records stored in the non-target table can be summarized to characterize the related records stored in the target table. In a relational database, a single record, R_i, stored in the target table can be associated with other records stored in the non-target table, as shown in Fig. 1. Let R denote a set of m records stored in the target table and let S denote a set of n records (T_1, T_2, T_3, ..., T_n), stored in the non-target table. Let S_i be a subset of S, $S_i \subseteq S$, associated through a foreign key with a single record R_a stored in the target table, where $R_a \in R$. Thus, the association of these records can be described as $R_a \Leftarrow S_i$. In this case, we have a single record stored in the target table that is associated with multiple records stored in the non-target table.

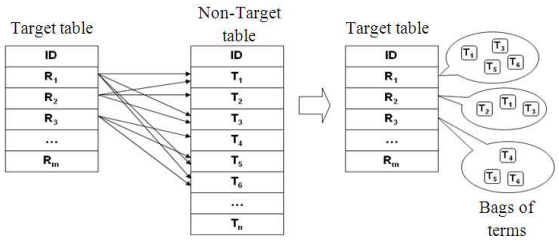

Fig. 1: A one-to-many association between target and non-target relations

The records stored in the non-target table that correspond to a particular record stored in the target table can be represented as vectors of patterns. As a result, based on the vector space model[4], a unique record stored in non-target table can be represented as a vector of patterns. In other words, a particular record stored in the target table that is related to several records stored in the non-target table can be represented as a bag of patterns, i.e., by the patterns it contains and their frequency, regardless of their order. The bag of patterns is defined as follows:

Definition: In a bag of patterns representation, each target record stored in the non-target table is represented by the set of its pattern and the pattern frequencies.

This definition follows the notion of an individual-centered representation defined by Lachiche and Flach[9], where the data is described as a collection of individuals and the induced rules generalize over the individuals, mapping them to a class. For instance, individual-centered domains include classification problems in molecular biology where the individuals are molecules.

In our approach, an individual is represented as a bag of patterns. In the DARA algorithm, these patterns are encoded into binary numbers. The process of encoding these patterns into binary numbers depends on the number of attributes that exist in the non-target table. For example, there are two different cases when encoding patterns for the data stored in the non-target table. In the first case (Case I), a non-target table may have a single attribute. In this case, the DARA algorithm transforms the representation of the data stored in a relational database without constructing any new feature to build the (n×p) TF-IDF weighted frequency matrix[4], as only one attribute exists in the non-target table. In the other case (Case II), a non-target table may have multiple attributes exist in the table. In this case, DARA may construct new features, which results in richer representation of each target record in the non-target table. The method used to encode the patterns derived from these attributes has some influences on the final results of the modeling task[11].

Case 1: A non-target table with a single attribute: Case 1 assumes that there is exactly one attribute describing the contents of the non-target table that is associated with the target table. For instance, in Fig. 2, the Trans attribute is the Primary Key (PK) of the Sales table and the Customer attribute is the Foreign Key (FK) of the table that associates records stored in this non-target table (sales table) with records stored in the target table (consists of individual customer).

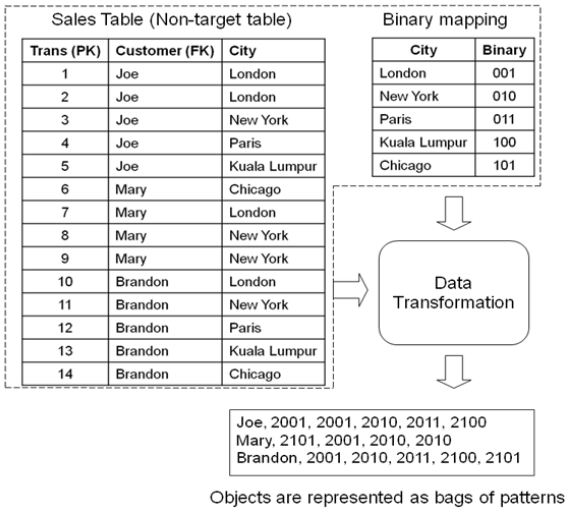

Fig. 2: Case I: A data transformation for data stored in a non-target table with a single attribute

The algorithm computes the cardinality of the attribute domain in the non-target table. Cardinality of an attribute is defined as the number of unique values that the attribute can take. If the data consists of continuous values, the data is discretized and the number of bins taken as the cardinality of the attribute domain. In order to encode the values into binary numbers, the algorithm finds the appropriate number of bits, n, such that it can represent all different values of the attribute's domain, where $2^{n-1} < |\text{Attribute's Domain}| \leq 2^n$.

For example, if the attribute has 5 different values (London, New York, Chicago, Paris, Kuala Lumpur), then we just need 3 ($2^2 < 5 \leq 2^3$) bits to represent each of these values (001, 010, 011, 100, 101), as shown in Fig. 2. A bag of patterns is maintained to keep track of the number of patterns encountered and their frequencies. For each encoded pattern, the counter for the corresponding pattern in the bag is incremented or the pattern is added to the bag of patterns if it is not already in the bag. The resulting bag of patterns, shown in Fig. 2, can be used to describe the characteristics of an individual record. In Fig. 2, the first digit "2" preceded the binary numbers indicates the index of attribute that the binary numbers are belong to. Since there is only one attribute exists in the datasets, all the encoded patterns produced are belong to index attribute "2".

Case 2: A non-target table with multiple attributes: Case 2 assumes that there is more than one attribute that describe the contents of the non-target table associated with the target table. All continuous values of the attributes are discretized and the number of bins is taken as the cardinality of the attribute domain.

Table 1: Number of attributes combined, p and the list of patterns produced

p	Patterns produced
1	$F_{1,a}$, $F_{2,b}$, $F_{3,c}$, $F_{4,d}$, ..., $F_{k-1,b}$, $F_{k,n}$
2	$F_{1,a}F_{2,b}$, $F_{3,c}F_{4,d}$, ..., $F_{k-1,b}F_{k,n}$ (with even number of attributes)
2	$F_{1,a}F_{2,b}$, $F_{3,c}F_{4,d}$, ..., $F_{k,n}$ (with odd number of attributes)
k	$F_{1,a}F_{2,b}F_{3,c}F_{4,d}...F_{k-1,b}F_{k,n}$

After encoding the patterns as binary numbers, the algorithm determines a subset of the attributes to be used to construct a new feature.

Here is an example of a simple algorithm to construct features without using feature scoring to generate the patterns that represent the input for the DARA algorithm. For each record stored in the non-target table, we concatenate p number of columns' values, where p is less than or equal to the total number of attributes. For example, let $F = (F_1, F_2, F_3, ..., F_k)$ denote k field columns or attributes in the non-target table. Let $dom(F_i) = (F_{i,1}, F_{i,2}, F_{i,3}, ..., F_{i,n})$ denote the domain of attribute F_i, with n different values. So, one may have an instance of a record stored in the non-target table with these values ($F_{1,a}$, $F_{2,b}$, $F_{3,c}$, $F_{4,d}$, ..., $F_{k-1,b}$, $F_{k,n}$), where $F_{1,a} \in dom(F_1)$, $F_{2,b} \in dom(F_2)$, $F_{3,c} \in dom(F_3)$, $F_{4,d} \in dom(F_4)$, ..., $F_{k-1,b} \in dom(F_{k-1})$, $F_{k,n} \in dom(F_k)$. Table 1 shows the list of patterns produced with different values of p. It is not natural to have concatenated features like $F_{1,a}F_{2,b}$ but not $F_{1,a}F_{3,c}$, when we have p = 2, since the attributes do not have a natural order. However, a genetic algorithm can be applied to solve this problem[10].

For each record, a bag of patterns is maintained to keep track of the patterns encountered and their frequencies. For each new pattern encoded, if the pattern exists in the bag, the counter for the corresponding pattern is increased, else the pattern is added to the bag and set the counter for this particular pattern to 1. The resulting bag of patterns can be used to describe the characteristics of a record associated with them.

In short, the encoding process described here transforms data stored in the non-target table that has many-to-one relations with the target table, to the representation of data in a vector-space model[4]. With this representation, the data can be conveniently clustered by using the hierarchical or partitioning clustering technique, as a means of summarizing them.

In short, the encoding process described here transforms data stored in the non-target table that has many-to-one relations with the target table, to the representation of data in a vector-space model[4]. With this representation, the data can be conveniently clustered by using the hierarchical or partitioning clustering technique, as a means of summarizing them.

Types of discretization: The motivation for the discretization of continuous features is based on the need to obtain higher accuracy rates in order to handle data with high cardinality attributes using the DARA algorithm, although this operation may affect the speed of any learning procedure that may subsequently use it. There are a few common methods used to discretize continuous attributes that include equal-width, equal-height, equal-weight and entropy-based discretization methods. A new method of discretization, called entropy-instance-based discretization, will also be introduced later. In the DARA algorithm, all attributes with continuous values are discretized before they are transformed into vector space data representation.

Discretization methods can be categorized along 3 axes[6]: (a) Supervised versus unsupervised (b) Global versus local and (c) Static versus dynamic. Supervised methods make use of the class label when partitioning the continuous features. On the other hand, unsupervised discretization methods do not require the class information to discretize continuous attributes. Next, the distinction between global and local methods is based on the stage when the discretization takes place. Global methods discretize features prior to induction. In contrast, local methods discretize features during the induction process.

Given k as the number of intervals or bins, some discretization methods discretize features independently of the other features-this is called static discretization. On the other hand, dynamic discretization methods search for the space of possible k values for all features at the same time and this allows inter-dependencies in feature discretization to be captured. In this study, the global discretization method is used to discretize continuous features. In addition to that, since there is no significant improvement in employing dynamic discretization over static methods[13], we employ the static method when discretizing the continuous features in this studies.

MATERIALS AND METHODS

Unsupervised discretization methods:

Equal-width discretization method: The simplest discretization method is called equal-width interval discretization and this method has often been applied as a means for producing nominal values from continuous ones. This approach divides the range of observed values for a feature into k equal sized bins, where k is a parameter provided by the user. The process involves sorting the observed values of a continuous feature and finding the minimum, V_{min} and maximum, V_{max}, values. The interval (Eq. 1) can be computed by dividing the range of observed values for the variable into k equally sized bins, where k is a parameter supplied by the user:

$$\text{Interval} = \frac{(V_{max} - V_{min})}{k} \quad (1)$$

$$\text{Boundaries} = V_{min} + (i \times \text{interval}) \quad (2)$$

and then the boundaries then can be constructed using Eq. 2 where i = 1,..., k-1. This type of discretization does not depend on the multi-relational structure of the data. However, this method of discretization is sensitive to outliers that may drastically skew the range[6].

Equal-height discretization method: Another simple discretization method, called equal-height interval binning, discretizes data so that each bin will have approximately the same number of samples. This method involves sorting the observed values together with the record ID. If |R| refers to the size of the records and V [|R|] refers to the size of the array that stores the sorted values, then the boundaries can be constructed as:

$$\text{Boundaries} = V\left[\left(\frac{|R|}{k}\right) \times i\right] \quad (3)$$

where, i = 1, ..., k-1. The result is a collection of k bins of roughly equal size. This algorithm is class-blind and does not take into consideration the structure of the database, especially the one-to-many association problem. Since unsupervised methods do not make use of the class information in finding the interval boundaries, the classification information can be lost as a result of placing values that are strongly associated with different classes in the same interval[6].

Equal-weight discretization method: Another unsupervised discretization method called the equal-weight interval binning, which was introduced by Knobbe and Ho[1]. The equal-weight discretization method considers not only the distribution of numeric values present, but also the groups they appear in. This method involves an idea proposed by Van Laer and De Raedt[16]. It is observed that larger groups have a bigger influence on the choice of boundaries because they have more contributing numeric values. In equal-weight interval binning, numeric values are weighted with the inverse of the size of the group they belong to and this weight is defined in Eq. 4:

$$W_t(v) = \frac{1}{|\text{group}_v|} \quad (4)$$

Where:
w_t = The weight function
v = The value being considered
$|group_v|$ = The size of the group that ϖ belongs to

Instead of producing bins of equal size, the boundaries are computed to obtain bins of equal-weight. The algorithm starts by computing the size of each group, then it moves through the sorted arrays of values, keeping a running sum of weights w_t. Whenever w_t reaches a target boundary $(\frac{number-of-groups}{bins})$, the current numeric value is added as one of the boundaries and the process is repeated k-1 times (k is the number of bins).

Supervised discretization methods:

Entropy-based discretization method: One of the supervised discretization methods, introduced by Fayyad and Irani, is called the entropy-based discretization[17]. A lot of significant research in entropy-based discretization has been carried out and an early comparison of entropy-based methods for discretization of continuous features and multi-interval discretization methods can be found in the works conducted by Kohavi and Sahami[13]. Algorithms, such as C4.5, try to find a binary cut for each attribute and use a minimal entropy heuristic for the discretization of continuous attributes. The algorithm uses the class information entropy to select binary boundaries for discretization. In entropy-based discretization, given a set of instances S, a feature A and a partition boundary T, the class information entropy is:

$$E(A,T,S) = \frac{s_1}{s}Ent(S_1) + \frac{s_2}{s}Ent(S_2) \quad (5)$$

where, S_1 and S_2 correspond to the samples in S satisfying the condition A<T and A≥T, respectively. The entropy function Ent for a given set is calculated based on the class distribution of the:

$$Ent(S) = -\sum_{i=1}^{c} p9C_i, S)\log_2(p(C_i, S)) \quad (6)$$

where, $p(C_i, S)$ is the probability of observing the ith class randomly in the subset S. This method can be applied recursively to both partitions induced by T until some stopping condition is achieved, thus creating multiple intervals of feature A. So, for k bins, the class information entropy for multi-interval entropy-based discretization is:

$$I(A,T,S,k) = \frac{\sum_{b=2}^{k}|S_b|.Ent(s_b)}{|s|} \quad (7)$$

The stopping condition proposed by Fayyad and Irani[17] is based on the Minimum Description Length (MDL) principal[2]. The stopping condition prescribes accepting a partition induced by T if and only if the cost of encoding the partition and the classes of the instances in the intervals induced by T is less than the cost of encoding the classes of the instances before the split as shown in Eq. 8:

$$Gain(A,T,S) < \frac{\log_2(N-1)}{N} + \frac{\Delta(A,T,S)}{N} \quad (8)$$

where, N is the number of instances in the set S and:

$$Ent(A,T,S) = Ent(s) - E(A,T,S) \quad (9)$$

$$\Delta(A,T,S) = \log_2(3^c-2) - [cEnt(S) - c_1End(S_1) - c_2End(S_2)] \quad (10)$$

and in Eq. 10, c, c_1 and c_2 are the number of distinct classes present in S, S_1 and S_2 respectively.

Equal-weight discretization method: This study introduces a new method of discretizing continuous attributes that takes into account the one-to-many association between records stored in the target and non-target tables. In this study, the entropy-based multi-interval discretization method introduced by Fayyad and Irani[17] is modified. In the proposed entropy-instance-based discretization method, besides the class information entropy, another measure that uses individual information entropy is added to select multi-interval boundaries for the continuous attributes. Given n individuals taken from the target table, the individual information entropy of a subset S is:

$$IndEnt(S) = -\sum_{i=1}^{1} p(l_i, S)\log_2(p(l_i, S)) \quad (11)$$

where, $p(I_i, S)$ is the probability that a random record associated with individual i from this table is in the subset S. This is due to the fact that in a multi-relational environment in which an entity may have a one-to-many relationship with another entity, an object stored in the target table may have more than one occurrence of its instances stored in the non-target table. For this reason, the total individual information entropy for all partitions is defined as:

$$Ind(A,T,S,k) = \frac{\sum_{b=1}^{k}|S_b|.IndEnt(S_b)}{|S|} \quad (12)$$

In other words, the Entropy-Instance-Based interval binning considers the distribution of numeric values present, the groups they appear in and is also based on all occurrences of each individual record. The individual information entropy (Eq. 12) is added to the existing entropy-based discretization formula in order to get better partitions in a multi-relational setting. For that reason, by minimizing the function IndInf (A, T, S, k) in Eq. 13, which consists of two base functions, I (A, T, S, k) and Ind(A, T, S, k), continuous attributes are discretized based on the class and individual information entropy:

$$\text{IndInf}(A,T,S,k) = \frac{\sum_{b=1}^{k}|S_b|.\text{IndEnt}(S_b)}{|S|} + \frac{\sum_{b=1}^{k}|S_b|.\text{Ent}(S_b)}{|S|} \quad (13)$$

$$\text{IndInf}(A,T,S,k) = \frac{\sum_{b=1}^{k}|S_b|.\text{IndEnt}(S_b) + \sum_{b=1}^{k}|S_b|.\text{Ent}(S_b)}{|S|} \quad (14)$$

Feature construction for data summarization: These experiments are designed to investigate:

- The effects of taking into account one-to-many relationships when discretizing continuous attributes in a multi-relational environment
- Whether the choice of clustering techniques has any impacts on the data summarization results

In this experimental study, the discretization methods, described previously, are implemented in the DARA algorithm[11,12], in conjunction with the C4.5 classifier (J48 in WEKA)[5], as an induction algorithm that is run on the discretized and transformed data by the DARA algorithm. Then, the effectiveness of each discretization method with respect to C4.5[7], is evaluated. Two datasets are chosen from the well-known Mutagenesis dataset[3] and the Hepatitis dataset for Discovery Challenge PKDD 2005[15].

There are four different values used for the number of bins, b = 2, 4, 6, 8, to evaluate different methods of discretization. For each dataset, the data summarization process is performed using both the Hierarchical (H) and Partitional (P) clustering techniques. After summarizing the datasets using the $\text{DARA}_{\text{small}}$ algorithm, the effectiveness of each discretization method with respect to C4.5[7] is evaluated using the 10-fold cross-validation.

RESULTS

Table 2 and 3 provide a detailed overview of the accuracy estimation from 10-fold cross-validation performance of C4.5 for different number of bins, b, tested on Mutagenesis datasets (B1, B2, B3) and Hepatitis datasets (H1, H2, H3), for each method of discretization. In these experiments, all five methods of discretization are evaluated, namely equal-width (EWD), Equal-Height (EH), Equal-Weight (EWG), Entropy-Based (EB) and Entropy-Instance-Based (EIB).

Based on the experimental results, in most cases, the entropy-instance-based discretization method produced better data summarization results that lead to a better performance accuracy for the predictive tool (C4.5), compared to the other discretization methods. There is one exception, in the Mutagenesis dataset B3, where the improvement of the data summarization results produced by entropy-instance-based discretization method is not that obvious.

Table 2 and 3 also show the behaviors of each discretization method with different values of bins, b, performed on the Mutagenesis and Hepatitis datasets. When b is big (b = 8), the entropy-instance-based discretization method produced better data summarization results compared to the other discretization methods, in the mutagenesis dataset.

Table 2: Performance accuracy (%) of 10-fold cross-validation of C4.5 on Mutagenesis dataset with different methods of discretization

		Mutagenesis dataset									
		EWD		EH		EWG		EB		EIB	
Data	b	H	P	H	P	H	P	H	P	H	P
B1	2	82.5	80.0	79.3	78.3	79.3	78.3	80.0	80.0	82.7	79.7
	4	82.5	80.0	74.4	74.2	74.4	74.2	80.0	80.0	82.7	79.7
	6	82.7	80.0	76.4	74.7	74.9	75.0	77.7	77.7	82.7	79.4
	8	80.8	79.6	73.4	74.9	73.9	74.9	72.7	72.7	82.7	79.4
B2	2	80.5	82.3	77.4	76.4	77.4	76.4	75.6	75.6	78.2	76.7
	4	77.5	81.4	74.1	71.7	74.4	71.9	73.4	73.4	82.6	79.7
	6	81.0	79.2	72.7	68.9	73.6	67.5	73.4	73.4	78.5	77.5
	8	73.3	73.9	71.1	72.7	73.6	74.9	73.3	73.3	78.5	77.5
B3	2	81.3	81.1	78.8	78.8	78.2	72.8	82.0	82.0	81.9	81.4
	4	77.5	81.4	81.7	82.7	82.8	83.3	78.9	78.9	79.9	81.4
	6	81.0	79.2	81.3	80.3	81.1	80.8	81.3	81.3	80.6	77.8
	8	73.3	73.9	81.0	80.3	80.8	81.6	80.2	80.2	81.6	80.6

Table 3: Performance accuracy (%) of 10-fold cross-validation of C4.5 on Hepatitis dataset with different methods of discretization

		Hepatitis PKDD 2005 dataset									
		EWD		EH		EWG		EB		EIB	
Data	b	H	P	H	P	H	P	H	P	H	P
B1	2	71.9	72.3	68.3	71.1	70.1	70.9	72.1	74.2	74.1	75.4
	4	71.2	71.5	71.4	71.5	71.4	71.0	70.2	71.8	74.3	75.5
	6	71.6	72.3	69.4	69.4	69.9	69.9	69.2	72.9	74.3	75.5
	8	70.8	69.6	69.4	69.4	69.9	69.9	70.1	71.9	74.3	75.6
B2	2	71.8	73.8	70.7	71.8	70.9	71.8	70.3	73.4	73.6	74.3
	4	72.6	74.5	71.0	72.5	72.0	72.0	69.8	73.4	74.1	75.1
	6	70.6	74.7	68.6	69.2	68.8	69.8	72.6	73.7	74.2	75.5
	8	70.8	71.7	70.0	69.5	70.3	69.8	69.6	72.7	73.7	75.4
B3	2	72.3	74.3	72.2	72.8	71.9	72.8	70.4	73.1	74.4	74.9
	4	73.1	75.6	72.2	72.9	71.8	72.4	70.1	73.7	74.7	75.1
	6	71.5	74.8	69.0	69.3	69.4	69.6	72.3	73.4	74.3	74.9
	8	71.6	70.9	70.4	69.7	70.4	69.9	70.3	71.9	74.2	74.9

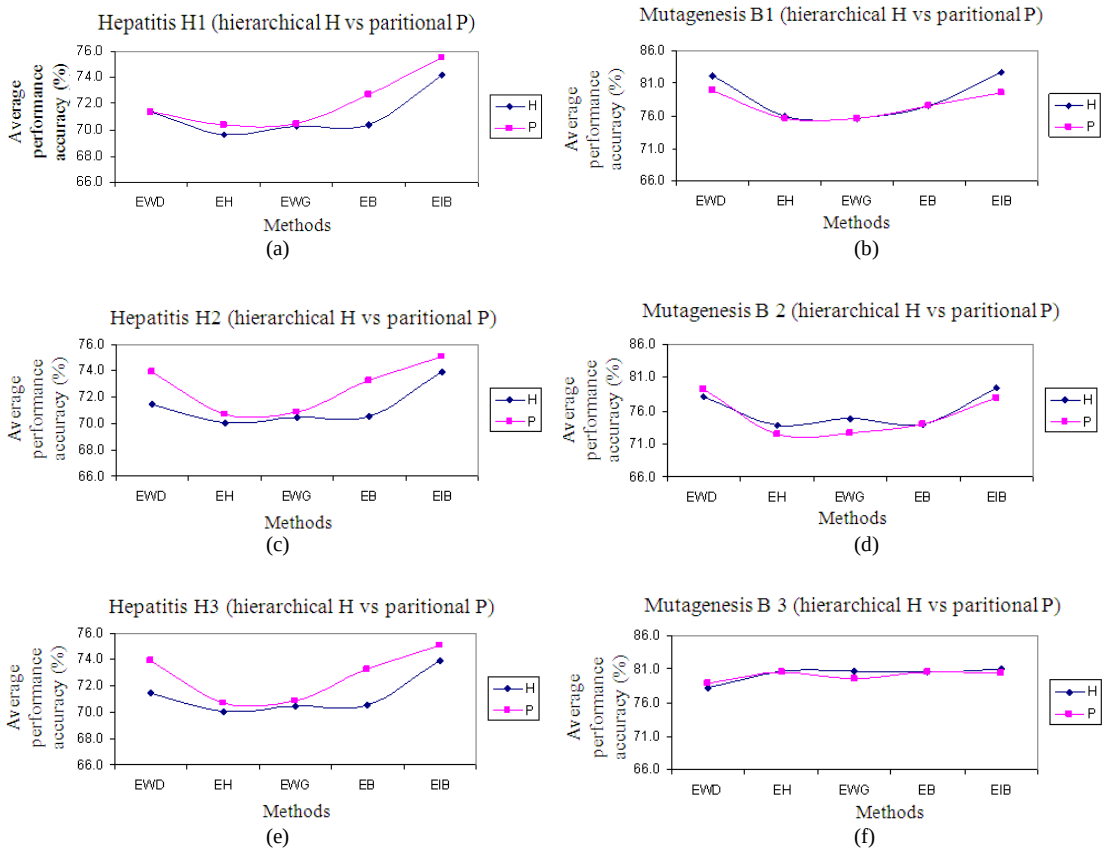

Fig. 3: Comparison of the average performance (%) of 10-fold cross validation of C4.5 on mutagenesis and hepatitis datasets for hierarchical and partitional clustering

For mutagenesis dataset, the optimal value for the number of discretization is relatively low: Between 2 and 4. In contrast, for the hepatitis dataset, the entropy-instance-based discretization method produced better data summarization results for all values of b, compared to the other discretization methods. The optimal value for the number of discretization for the hepatitis dataset is less clear.

Fig. 3 also shows that the Entropy-Instance-Based (EIB) discretization method produced higher average performance accuracy (%), for both hierarchical and partitional clustering techniques, compared to the Entropy-Based (EB), Equal-Height (EH), Equal-Weight (EWG) and finally Equal-Width (EWD) discretization methods, for datasets H1, H2, H3 and B3. However, for datasets B1 and B2, the equal-width discretization method produced comparable results with the one produced by entropy-instance-based discretization method.

The results of paired t-test ($p = 0.05$) to indicate the significant improvement of each discretization method over the other methods for mutagenesis and hepatitis datasets are also collected. Since there are three varieties of datasets for each hepatitis and mutagenesis databases, with four different values for the number of bins (b = 2, 4, 6, 8), there are 24 cases in which each discretization method is evaluated for each database. Table 4 and 5 show the number of cases in which the method of discretization in row indicates significant improvement over the other discretization methods in column. For hepatitis dataset, Table 4 shows that the entropy-instance-based discretization method has higher number of cases in which this discretization method indicates significant improvement over the rest of the discretization methods. For the mutagenesis dataset, Table 4 shows that both the equal-width and entropy-instance-based discretization methods have higher number of cases in which these discretization methods indicate significant improvement over the other discretization methods.

Table 4: The number of cases in which the method of discretization in row indicates significant improvement over the other discretization methods in column for the Hepatitis dataset

Hepatitis PKDD 2005					
Methods	EWD	EH	EWG	EB	EIB
EWD	-	7	7	3	0
EH	0	-	0	0	0
EWG	0	0	-	0	0
EB	5	10	10	-	0
EIB	15	24	24	15	-

Table 5: The number of cases in which the method of discretization in row indicates significant improvement over the other discretization methods in column for the Mutagenesis dataset

Hepatitis PKDD 2005					
Methods	EWD	EH	EWG	EB	EIB
EWD	-	12	14	9	3
EH	3	-	1	0	0
EWG	3	0	-	0	0
EB	2	3	4	-	0
EIB	4	11	11	8	-

Table 6: Hepatitis Datasets: The percentage of significant improvement for each discretization method over the other methods (Won = percentage of significant improvement of the discretization method in row over the method in column)

Hepatitis							
		EWD	EH	EWG	EB	EIB	Average
Discretizations:							
EWD	Won		29.2	29.2	12.5	00.0	17.7
	Ties		70.8	70.8	66.7	37.5	61.5
	Loss		00.0	00.0	20.8	62.5	20.8
EH	Won	00.0		00.0	00.0	00.0	00.0
	Ties	93.0		100.0	100.0	00.0	73.3
	Loss	29.2		00.0	00.0	100.0	32.3
EWG	Won	00.0	00.0		00.0	00.0	00.0
	Ties	70.8	100.0		100.0	00.0	67.7
	Loss	29.2	00.0		00.0	100.0	32.3
EB	Won	20.8	41.7	41.7		00.0	26.1
	Ties	66.7	58.3	58.3		37.5	55.2
	Loss	12.5	00.0	00.0		62.5	18.8
EIB	Won	62.5	100.0	100.0	62.5		81.3
	Ties	37.5	00.0	00.0	37.5		18.8
	Loss	00.0	00.0	00.0	00.0		00.0

From Table 4 and 5, the percentages of significant improvement performed by each discretization method are computed in Table 6 and 7. For the hepatitis dataset, the entropy-instance-based discretization method has the highest average percentage of significant improvement. In contrast, for the mutagenesis dataset, both the equal-width interval discretization and the Entropy-Instance-Based (EIB) discretization methods show high average percentage of ties (no significant improvement). However, both methods show reasonable high average percentage of significant improvement over the other discretization methods, with EIB having the lowest percentage of loss.

Table 7: Mutagenesis Datasets: The percentage of significant improvement for each discretization method over the other methods (Won = percentage of significant improvement of the discretization method in row over the method in column)

Hepatitis							
		EWD	EH	EWG	EB	EIB	Average
Discretizations:							
EWD	Won		50.0	58.2	37.5	12.5	39.6
	Ties		37.5	29.3	54.2	70.8	48.0
	Loss		12.5	12.5	8.3	16.7	12.5
EH	Won	12.5		4.2	00.0	00.0	4.2
	Ties	37.5		95.8	87.5	54.2	68.8
	Loss	00.0		00.0	12.5	45.8	27.1
EWG	Won	12.5	00.0		00.0	00.0	3.1
	Ties	29.2	95.8		83.3	54.2	65.6
	Loss	58.3	4.2		16.7	45.8	31.3
EB	Won	8.3	12.4	16.7		00.0	9.4
	Ties	54.2	87.6	83.3		66.7	73.0
	Loss	37.5	00.0	00.0		33.3	17.7
EIB	Won	16.7	45.8	45.8	33.3		35.4
	Ties	70.8	54.2	54.2	66.7		61.5
	Loss	12.5	00.0	00.0	00.0		3.1

DISCUSSION

In general, based on these experiments we can conclude that the Entropy-Instance-Based (EIB) discretization method helps one to achieve higher percentage of accuracy. This should come as no surprise, as the EIB is more precise in choosing the optimal numeric cut points. In other words, EIB splits the data better based on the class information and also the individual information entropy. As a result, each object in the target table can be described more accurately since each object possesses more consistent patterns used for clustering.

It is also found that the partitional clustering technique often performs much better compared to the hierarchical clustering technique in summarizing data with multiple occurrences stored in the non-target relation. Fig. 3 shows the comparison of the average performance accuracy (%) for the hierarchical and partitional clustering techniques on both the mutagenesis and hepatitis datasets.

In clustering, the frequency of patterns is used to distinguish records of different classes. And most records may have only a subset of all patterns from the complete patterns used to cluster these records and any two records may share many of the same patterns. As a result, two records could often be nearest neighbors without belonging to the same classes. Since, the nearest neighbors of a record are of different classes, hierarchical clustering technique will often put records of different classes in the same cluster, even at the earliest stages of the clustering process. In cases where nearest neighbors are unreliable, partitioning clustering

technique (such as K-means) that relies on more global properties[14] is needed. In partitioning clustering technique, computing the cosine similarity of a record to a cluster centroid is the same as computing the average similarity of the record to all the clusters records[8], the partitioning clustering technique is implicitly making use of such a global property approach. For that reason, this explains why partitioning clustering technique does better compared to the hierarchical clustering technique in the categorical domain, although this is not the case in some other domains.

One of the main problems with Entropy-Based and Entropy-Instance-Based discretization criterion is that they are relatively expensive. For instance, for 2 bins (k = 2), for a continuous attribute, the Eq. 7 and 12 must be evaluated N-1 times for each attribute, where N is the number of attribute values. Therefore, one may use a genetic algorithm-based discretization[11], in order to obtain a multi-interval discretization for continuous attributes in a very large database, using Entropy-Based or Entropy-Instance-Based methods.

CONCLUSION

This study has revealed, through experiments, that the entropy-instance-based discretization method, which is implemented in the DARA algorithm, helps one to achieve higher percentage of accuracy. The entropy-instance-based discretization method is recommended for discretization of attribute values in multi-relational datasets, in which the individual information entropy can be used to improve the discretization process, as it has been shown here. However, when the dataset is too large, one may apply the genetic algorithm-based for the entropy-instance-based discretization method to find the best partitions. It is also found that, from this experiment, the partitional clustering technique produced better performance accuracy compared to the one produced by hierarchical clustering technique.

REFERENCES

1. Knobbe, A.J. and K.Y.H. Eric, 2005. Numbers in multi-relational data mining. Proceeding of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Oct. 2005, Springer Berlin, Heidelberg, pp: 544-551. DOI: 10.1007/11564126
2. Barron, A.R., J. Rissanen and B. Yu, 1998. The minimum description length principle in coding and modeling. IEEE Trans. Inform. Theor., 44: 2743-2768. DOI: 10.1109/ 18.720554
3. Srinivasan, A., S. Muggleton and R. King, 1995. Comparing the use of background knowledge by inductive logic programming systems. Proceeding of the 5th International Workshop on Inductive Logic Programming, (IWILP'95), Department of Computer Science, Katholieke University Leuven, pp: 199-230. ftp://ftp.comlab.ox.ac.uk/pub/Packages/ILP/Papers/TR/prg-tr-9-95.ps.gz
4. Salton, G. and M. McGill, 1984. Introduction to Modern Information Retrieval. 1st Edn., McGraw-Hill Book Company, USA. , ISBN: 0070544840, pp: 448.
5. Witten, I. and E. Frank, 1999. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. 1st Edn., Morgan Kaufman, pp: 416.
6. Dougherty, J., R. Kohavi and M. Sahami, 1995. Supervised and unsupervised discretization of continuous features. Proceeding of the International Conference on Machine Learning (ICML 1995), pp: 194-202. DOI: 10.1.1.47.6141
7. Quinlan, J.R., 1990. C4.5: Programs for Machine Learning. 1st Edn., Morgan Kaufmann Publishers Inc., San Francisco, CA., USA., pp: 302.
8. Steinbach, M., G. Karypis and V. Kumar, 2000. A comparison of document clustering techniques. Proceeding of the Text Mining Workshop, University of Minnesota, pp: 1-34. DOI: 10.1.1.34.1505
9. Lachiche, N. and P. Flach, 2000. A first-order representation for knowledge discovery and bayesian classification on relational data. Proceeding of the Workshop of 4th International Conference on Principles of Data Mining and Knowledge Discovery, Sept. 2000, pp: 49-60. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.8554
10. Alfred, R., 2008. A genetic-based feature construction method for data summarization. Proceeding of the Advanced Data Mining and Applications, Oct. 2008, Springer Berlin, Heidelberg, pp: 39-50. DOI: 10.1007/978-3-540-88192-6
11. Alfred, R. and D. Kazakov, 2007. Clustering approach to generalized pattern identification based on multi-instanced objects with DARA. Proceeding of the 11th East-European Conference on Advances Databases and Information Systems, Sept. 2007, CEUR-WS.org, Technical University of Varna, pp: 38-49. http://ceur-ws.org/Vol-325/paper04.pdf

12. Alfred, R. and D. Kazakov, 2006. Pattern-based transformation approach to relational domain learning using DARA. Proceeding of the 2006 International Conference on Data Mining, July 2006, CSREA Press, Las Vegas, pp: 296-302. http://ww1.ucmss.com/books/LFS/CSREA2006/DMI5526.pdf
13. Kohavi, R. and M. Sahami, 1996. Error-based and entropy-based discretization of continuous features. Proceeding of the 2nd International Conference on Knowledge Discovery and Data Mining, Aug. 1996, AAAI Press, pp: 114-119. http://www.aaai.org/Papers/KDD/1996/KDD96-019.pdf
14. Guha, S., R. Rastogi and K. Shim, 1999. ROCK: A robust clustering algorithm for categorical attributes. Proceeding of the 15th International Conference on Data Engineering, Mar. 1999, IEEE Computer Society Press, Sydney, Australia, pp: 512-521, DOI: 10.1016/S0306-4379(00)00022-3
15. Tsumoto, S., 2000. Knowledge discovery in clinical databases and evaluation of discovered knowledge in outpatient clinic. J. Inform. Sci., 124: 125-137. DOI: 10.1016/S0020-0255(99)00065-1
16. Laer, W.V., L. De Raedt and S. Dzeroski, 1997. On Multi-class problems and discretization in inductive logic programming. Proceeding of the 10th International Symposium on Foundations of Intelligent Systems, Oct. 15-18, Springer-Verlag, London, UK., pp: 277-286. http://portal.acm.org/citation.cfm?id=689618
17. Fayyad, U.M. and K.B. Irani, 1993. Multi-interval discretization of continuous valued attributes for classification learning. Proceeding of the 13th International Joint Conference on Artificial Intelligence, Aug. 28-Sept. 3, Chambéry, France, Morgan Kaufmann, pp: 1022-1027. http://sci2s.ugr.es/keel/pdf/algorithm/congreso/fayyad1993.pdf