A Study On Discretization Techniques

B. Hemada ,
Department of Computer Science and Engineering,
V.R.Siddhartha Engineering College (Autonomous)
Affiliated to JNTUK, Vijayawada,
Andhra Pradesh, India.

K.S.Vijaya Lakshmi
Department of Computer Science and Engineering,
V.R.Siddhartha Engineering College (Autonomous)
Affiliated to JNTUK, Vijayawada,
Andhra Pradesh, India.

Abstract

The task of extracting knowledge from databases is quite often performed by machine learning algorithms. Many algorithms can only process discrete attributes. Real-world databases often involve continuous features. Those features have to be discretized before using such algorithms. Discretization methods can transform continuous features into a finite number of intervals, where each interval is associated with a numerical discrete value. This paper analyzed existing data discretization techniques for data preprocessing. Firstly, the importance and process of discretization is studied. Furthermore, we conduct an experimental study of discretization methods involving the most representative and newest discretizers. It's essential to select proper methods depending on learning environment. At last, the thought of choosing the best discretization methods in association analysis is proposed as future research.

Keywords – *Discretization, Continuous data, Data Mining, Classification.*

1. Introduction.

Data mining can be defined as the non trivial process of identifying valid, novel, potentially useful, ultimately understandable patterns in data. Even though the modeling phase is the core of the process, the quality of the results relies heavily on data preparation which usually takes around 80% of the total time. An interesting method for data preparation is to discretize the input variables. Discretization of continuous attributes plays an important role in knowledge discovery. Many algorithms related to data mining require the training examples that contain only discrete values, and the rules generated by classification algorithms with discrete values are normally shorter and more understandable. Suitable discretization is useful to increase the generalization and accuracy of discovered knowledge.

Discretization is the process of dividing the range of the continuous attribute into intervals. Every interval is labeled a discrete value, and then the original data will be mapped to the discrete values.

Discretization of the continuous attributes is an important preprocessing approach for data mining and machine learning algorithm. An effective discretization method not only can reduce the demand of system memory and improve the efficiency of data mining and machine learning algorithm, but also make the knowledge extracted from the discretized dataset more compact, easy to be understand and used. Research shows that picking the best split points is a NP-complete problem. The result of discrimination is related not only with the discretization algorithm itself but also with the data distribution and the number of split points. When the same discretization algorithm is applied to different dataset, we may get different result. We can only know the effectiveness of the discretization method by the result of post processing. So whether the discretization method is good or not is also related with the induction algorithm adopted later.

There are many advantages of using discrete values over continuous ones: (1) Discretization will reduce the number of continuous features' values, which brings smaller demands on system's storage. (2) Discrete features are closer to a knowledge-level representation than continuous ones. (3) Data can also be reduced and simplified through discretization. For both users and experts, discrete features are easier to understand, use, and explain. (4)

Discretization makes learning more accurate and faster. (5) In addition to the many advantages of having discrete data over continuous one, a suite of classification learning algorithms can only deal with discrete data. Successful discretization can significantly extend the application range of many learning algorithms.

2. Categorization of Discretization Approaches.

Discretization algorithms can be categorized into **supervised** and **unsupervised** based on whether the class label information is used. Supervised discretization uses class information to guide the discretization process, while the unsupervised discretization does not. Equal Width and Equal Frequency are two representative unsupervised discretization algorithms. Many supervised discretization techniques have been proposed to date, of which the Entropy-MDLP discretization has been accepted as by far the most effective in the context of both decision tree learning and rule induction algorithms. Compared to supervised discretization, previous research has indicated that unsupervised discretization algorithms have less computational complexity, but may result in much worse classification performance. When classification performance is the main concern, supervised discretization should be adopted.

The usage of Discretization methods can be **dynamic** or **static**. A dynamic method would discretize continuous values when a classifier is being built, such as in C4.5 while static discretization is done prior to classification task.

Another dimension of discretization methods is **local** *vs.* **global**. A local method would discretize in a localized region of instance space (i.e., a subset of instances) while a global discretization method uses the entire instance space to discretize.

Discretization methods can also be grouped in terms of **top-down** or **bottom-up**. Top-down methods start with an empty list of cut-points (or split-points) and keep on adding new ones to the list by 'splitting' intervals as the discretization progresses. Bottom-up methods start with the complete list of all the continuous values of the feature as cut-points and gradually remove some of them by 'merging' intervals as the discretization progresses.

Another dichotomy is **direct** *vs.* **incremental**. Direct methods divide the range of *k* intervals simultaneously (i.e., equal-width, equal-frequency, or K-means), needing an additional input from the user to determine the number of intervals. Incremental methods begin with simple discretization and are followed by an improvement or refinement process, which requires a stopping criterion to halt further discretization.

Discretization can be **univariate** or **multivariate**. Univariate discretization quantifies one feature at a time while multivariate discretization considers simultaneously multiple features.

3. A Typical Discretization Process.

A typical (univariate) discretization process broadly consists of four steps. (1) *Sort* the continuous values of the feature to be discretized, (2) *Evaluate* a cut-point for splitting or adjacent intervals for merging, (3) According to some criterion, *split* or *merge* the intervals of continuous value, and (4) finally *stop* discretization.

3.1. Sorting

The continuous values for a feature are sorted in either ascending or descending order. If sorting is done once and for all at the beginning of discretization, it is global treatment and can be applied when the entire instance space is used for discretization. If sorting is done at each iteration of a process, it is a local treatment in which only a region of entire instance space is considered for discretization.

3.2. Choosing a cut-point

After sorting, the next step in the discretization process is to find the best cut-point to split a range of continuous values or the best pair of adjacent intervals to merge. There are numerous evaluation functions such as entropy measures and statistical measures.

3.3. Splitting / Merging

In the top-down approach, intervals are split while for a bottom-up approach intervals are merged. For splitting, it is required to evaluate cut-points and to choose the best one and split the range of continuous values into two partitions. Discretization continuous with each part (increased by one) until a stopping criteria is satisfied. For merging, adjacent intervals are evaluated to find the best pair of intervals to merge in each iteration. Discretization continuous with the reduced number (decreased by one) of intervals until the stopping criterion is satisfied.

3.4. Stopping Criteria

A stopping criterion specifies when to halt the discretization process. A stopping criterion can be very simple such as fixing the number of intervals at the beginning or a more complex one like evaluating a function.

4. Literature Survey.

The past few decades have seen many researches on discretization for mining association rules. In this paper we study few unsupervised and supervised discretization methods. Equal-Width and Equal-Frequency are two commonly used unsupervised discretization methods. Both Equal-Width and Equal-Frequency methods require a parameter n, indicating the maximum number of intervals in discretizing a feature. The Equal-width [2] discretization technique determines the interval width according to the user-specified number of intervals using the relation

$$i_w = (max\_value - min\_value) / n,$$

where

i_w = interval width

max_value = maximum value of the attribute

min_value = minimum value of the attribute

It then creates the cut points using the relation

$$cut\_point = min\_value + j * i_w$$

where

j=1,2,……n-1

The Equal-frequency [2] discretization technique is similar to equal-width with the exception that the number of unique values (frequency) within each of the user-specified n intervals should be equal. The interval frequency is obtained using the following relation

$$i_f = nb\_unique\_values / n$$

where

i_f = interval frequency

nb_unique_values = number of unique values for a continuous attribute

The two methods are simple but are sensitive to n. For equal-frequency, for instance, many occurrences of a continuous value could cause the occurrences to be assigned into different bins. This can be handled by adjusting boundaries of neighboring bins so that duplicate values should belong to one bin only. Another problem is the presence of outliers that take extreme values. This can be overcome by removing the outliers using a threshold.

Fayyad and Irani proposed a supervised discretization method, Ent-MDLP [3] which uses entropy measure to find a potential cut-point to split a range of continuous values into two intervals. An entropy-based method will use the class information entropy of candidate partitions to select boundaries for discretization. Class information entropy is a measure of purity and it measures the amount of information which would be needed to specify to which class an instance belongs. The entropy measure in the context of classification can be defined as

$$E_e = E_1 + E_2$$

$$E_e = - p_{left} \sum_{i=1}^{k} p_{i,left} \log p_{i,left}$$

$$- p_{right} \sum_{i=1}^{k} p_{i,right} \log p_{i,right}$$

where

E_e = entropy of the cut-point
E_1 = entropy to the left of the cut-point
E_2 = entropy to the right of the cut-point
k = total number of classes
i = a practical class

p_{left} = number of instances to the left of cut-point / total number of instances, N
p_{right} = number of instances to the right of cut-point / total number of instances, N
$p_{i,left}$ = num of instances of class *i* to the left of cut-point / number of instances to the left of cut-point
$p_{i,right}$ = {num of instances of class *i* to the right of cut-point } / { number of instances to the right of cut-point}

It considers one big interval containing all known values of a feature and then recursively partitions this interval into smaller subintervals until the stopping criterion satisfies. The stopping criterion was based on the MDL (Minimum Description Length) principle which is defined as

$$gain > \frac{\log_2 (N-1)}{N} + \frac{\log_2 (3^k-2) - k\,E + k_1\,E_1 + k_2\,E}{N}$$

where

$$E = - \sum_{i=1}^{k} p_i \log p_i$$

$$p_i = \frac{\text{Number of instances of Class } i}{N}$$

gain = $E - E_e$ = information gained by splitting at the cut-point

N = total number of instances in the attribute value list at each recursion

k_1 = number of classes to the left of the cut-point

k_2 = number of classes to the right of the cut-point.

A supervised, static and global discretization method which uses the Gini gain[4] as discretization measure was proposed by Xiao-Hang Zhang, Jun Wu, Ting-Jie Lu and Yuan Jiang. In this discretization method the cut point is chosen based on the criterion, whose Gini gain value is the biggest on attribute A. The Gini gain ΔG is defined as

$$\Delta G(A,b;S) = Gini(S) - \frac{|S_1|}{|S|} Gini(S_1) - \frac{|S_2|}{|S|} Gini(S_2)$$

where

S_1 and S_2 are the subsets of S partitioned by the cut point b

Gini (.) is the Gini measure defined by

$$G(interval) = 1 - \sum_{j=1}^{k} (p_j^{(i)})^2$$

$p_j^{(i)}$ is the *j*th class probability in *i*th interval and satisfies $\sum_{j=1}^{k} p_j^{(i)} = 1$.

|.| denotes the number of instances.

The training set is split into two subsets by the cut point which is chosen using Gini measure. Subsequent cut points are selected by recursively applying the same binary discretization method to one of the generated subsets, which has biggest Gini gain value, until the stopping criterion is achieved. The stopping criterion of the discretization algorithm is defined by

$$G_{n+1} \ln(n+1+p) > G_n \ln(n+p)$$

Where

n denotes the current number of intervals,

p is a positive integer determined by the user,

G_n is the Gini value with n intervals, defined by

$$G_n = \sum_{i=1}^{n} \frac{|S1|}{|S|} Gini(interval\ i)$$

When compared with Ent-MDLP algorithm, the results has shown that in many applications the Gini algorithm has better performance than Ent-MDLP algorithm and original C4.5 algorithm. So it can be a good alternative to the entropy-based discretization methods.

Lukasz A. Kurgan and Krzysztof J. Cios proposed a supervised CAIM (class-attribute interdependence maximization) discretization algorithm [5] that handles continuous and mixed mode attributes. The CAIM algorithm's goal is to find the minimum number of discrete intervals while minimizing the loss of class-attribute interdependency. The algorithm uses class-attribute interdependency information as the criterion for the optimal discretization. The Class-Attribute Interdependency Maximization (CAIM) criterion measures the dependency between the class variable *C* and the discretization variable *D* for attribute *F*, for a given quanta matrix. The CAIM criterion is defined as

$$CAIM(C, D|F) = \frac{\sum_{r=1}^{n} (max_r^2 / M_{+r})}{n}$$

where,

n is the number of intervals,

r iterates through all intervals, i.e. r=1,2,...,n,

max_r is the maximum value among all qir values (maximum value within the rth column of the quanta matrix), i=1,2,...,*S*,

M_{+r} is the total number of continuous values of attribute *F* that are within the interval (dr-1, dr].

The algorithm starts with a single interval that covers all possible values of a continuous attribute, and divides it iteratively. From all possible division points that are tried it chooses the division boundary that gives the highest value of the CAIM criterion. When the algorithm was tested on several well-known datasets and compared with six other state-of the-art discretization algorithms, the comparison showed that the CAIM algorithm generated discretization schemes with, on average, the lowest number of intervals and the highest dependence between class labels and discrete intervals, thus outperforming other discretization algorithms. The execution time of the CAIM algorithm is also much shorter than the execution time of some other supervised discretization algorithms. The analysis of performance of the CAIM algorithm shows that the algorithm that generates small number of intervals helps to reduce the size of the data and improves the accuracy and the number of subsequently generated rules.

The Khiops discretization method proposed by Marc Boulle [6] is a bottom-up method based on the global optimization of chi-square(χ^2). χ^2 is a statistical

measure that conducts a significance test on the relationship between the values of a feature and a class. χ^2 statistic determines the similarity of adjacent intervals based on some significance level. It tests the hypothesis that two adjacent intervals of a feature are independent of the class. If they are independent, they should be merged, otherwise they should remain separate. The formula for computing χ^2 value is

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{p} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

where:

p = number of classes

A_{ij} = number of distinct values in the *i*th interval, *j*th class

R_i = number of examples in *i*th interval = $\sum_{j=1}^{p} A_{ij}$

C_j = number of examples in *j*th class= $\sum_{i=1}^{m} A_{ij}$

N = total number of examples = $\sum_{j=1}^{p} C_j$ and

E_{ij} = expected frequency of A_{ij} = $(R_i * C_j) / N$

The discretization method starts from the elementary single value intervals and then searches for the best merge between adjacent intervals. Two different types of merges are encountered. First, merges with at least one interval that does not meet the constraint and second, merges with both intervals fulfilling the constraint. The best merge candidate (with the highest chi-square value) is chosen in priority among the first type of merges (in which case the merge is accepted unconditionally), and otherwise, if all minimum frequency constraints are respected, among the second type of merges (in which case the merge is accepted under the condition of improvement of the confidence level). The algorithm is reiterated until both all minimum frequency constraints are respected and no further merge can decrease the confidence level. When compared with other chi-square based methods like ChiMerge and ChiSplit methods, this global evaluation carries some intrinsic benefits. The Khiops automatic stopping rule brings both ease of use and high quality discretizations. Its computational complexity is the same as for the fastest other discretization methods.

Quisha Zhu, Lin Lin, Mei-Ling Shyu and Shu-Ching Chen [7] proposed a novel and effective supervised discretization algorithm based on correlation maximization (CM). It is proposed by using multiple correspondence analyses (MCA). MCA is an effective technique to capture the correlations between intervals/items and classes. The one that gives the highest correlation with the classes is selected as a cut-point. The geometrical representation of MCA not only visualizes the correlation relationship between intervals/items and classes, but also presents an elegant way to decide the cut-points. The graphical representation of MCA is called symmetric map. It is used to visualize the intervals of a feature and the classes as points in a two-dimensional map. The correlation between an interval and a class can be represented by the cosine angle between these 2 vectors in the first 2 dimensions. The larger the cosine value of the angle is the stronger the correlation between them.

For a numeric feature F_i, all values of this feature are sorted to form a set of n+1 distinct values. Candidate cut points are the mid points of all adjacent pairs in the set. The cut point with the largest cosine is selected as the first cut-point T1. Then the same strategy can be carried out separately in the left and right intervals in a binary recursive way. The recursion is terminated if the correlation between current intervals and classes is lower than the correlation between their predecessor and their classes. When compared with other discretization algorithms, this algorithm produced relatively small number of intervals and also has a low computational complexity. The drawback of this method is, it cannot discretize the datasets containing more than 2 classes.

5. Conclusion and Future Scope.

Discretization of continuous features plays an important role in data pre-processing. This paper briefly introduces that the generation of the problem of discretization brings many benefits including improving the algorithm's efficiency and expanding their application scope. From the past few decades much work has been done in this area resulting in many different discretization methods. Choosing a suitable discretization method largely depends on the user need for discretization, as well as on the kind of data to be discretized. While a lot of work has been done, there are still many issues that remained unsolved, and new methods are needed to address these issues. In future we expect robust discretization techniques which can overcome the drawbacks of handling huge data and large number of attributes.

References

[1] Salvador Garc´ıa, Juli ´an Luengo, Jos´e A. S´aez, Victoria L´opez, and Francisco Herrera "A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning" *IEEE Transactions On Knowledge And Data Engineering*, Oct 2011.

[2] A.K.C. Wong and D.K.Y. Chiu, "Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 6, pp. 796-805, Nov. 1987.

[3] U.M. Fayyad and K.B. Irani, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning," *Proc. 13th Int'l Joint Conf. Artificial Intelligence*, pp. 1022-1027, 1993.

[4] Xiao-Hang Zhang, Jun Wu, Ting-Jie Lu, Yuan Jiang "A discretization algorithm based on gini criterion" , *Proceedings Of The Sixth International Conference On Machine Learning And Cybernetics, Hong Kong*, 19-22 august 2007.

[5] Lukasz A. Kurgan, and Krzysztof J. Cios, "CAIM Discretization Algorithm" IEEE *Transactions On Knowledge And Data Engineering*, *Vol*. 16, *No*. 2, *February* 2004.

[6] Marc Boulle "*Khiops: A Statistical Discretization Method of Continuous Attributes*" Machine Learning, 55, 53–69, 2004.

[7] Qiusha Zhu, Lin Lin, Mei-Ling Shyu, Shu-Ching Chen " Effective Supervised Discretization for Classification based on Correlation Maximization" *Information Reuse And Integration (Iri)*, 2011 IEEE *International Conference On* 3-5 Aug. 2011.

[8] A.An and N.Cercone, " Discretization of Continuous Attributes for Learning Classification Rules," *Proc. Third Pacific-Asia Conf. Methodologies For Knowledge Discovery And Data Mining, Pp.* 509-514, 1999.

[9] S.Kotsiantis and D.Kanellopoulos "Discretization techniques: A recent survey", *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58, 2006.

[10] H. Liu, F. Hussain, C.L.Tan, and M. Dash, "Discretization: An enabling technique," *Data Mining and Knowledge Discovery*, vol. 6, no. 4, pp. 393‑423, 2002.

[11] J.Y.Ching, A.K.C.Wong, and K.C.C.Chan, "*Class-Dependent Discretization for Inductive Learning from Continuous and Mixed Mode Data*," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 7, pp. 641-651, July 1995.

[12] W. Huang, "*Discretization of Continuous Attributes for Inductive Machine Learning*," master's thesis, Dept. Computer Science, Univ. of Toledo, Ohio, 1996.

[13] J. Dougherty, R. Kohavi, and M. Sahami, "*Supervised and Unsupervised Discretization of Continuous Features*," Proc. 12th Int'l Conf. Machine Learning, pp. 194-202, 1995.

[14] Y. Yang, G. I.Webb, and X.Wu, "Discretization methods," in *Data Mining and Knowledge Discovery Handbook*, 2010, pp. 101‑116.

[15] Fayyad, U. and Irani, K. Discretizing continuous attributes while learning bayesian networks. *In Proc. Thirteenth International Conference on Machine Learning*. Morgan Kaufmann, pp. 157–165, 1996.

[16] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical machine learning tools and techniques.* 3rd Edition. Morgan Kaufmann, 2011.

[17] Khurram Shehzad "EDISC: A Class-Tailored Discretization Technique for Rule-Based Classification" IEEE *Transactions On Knowledge And Data Engineering, Vol*. 24, *No*. 8, *August* 2012.