Discretizing Continuous Attributes Using Information Theory

Chang-Hwan Lee

Department of Information and Communications, DongGuk University,
Seoul, Korea 100-715
chlee@dgu.ac.kr

Abstract. Many classification algorithms require that training examples contain only discrete values. In order to use these algorithms when some attributes have continuous numeric values, the numeric attributes must be converted into discrete ones. This paper describes a new way of discretizing numeric values using information theory. The amount of information each interval gives to the target attribute is measured using Hellinger divergence, and the interval boundaries are decided so that each interval contains as equal amount of information as possible. In order to compare our discretization method with some current discretization methods, several popular classification data sets are selected for discretization. We use naive Bayesian classifier and C4.5 as classification tools to compare the accuracy of our discretization method with that of other methods.

1 Introduction

Discretization is a process which changes continuous numeric values into discrete categorical values. It divides the values of a numeric attribute into a number of intervals, where each interval can be mapped to a discrete categorical or nominal symbol. Most real-world applications of classification algorithm contain continuous numeric attributes. When the feature space of data includes continuous attributes only or mixed type of attributes (continuous type along with discrete type), it makes the problem of classification vitally difficult. For example, classification methods based on instance-based measures are generally difficult to apply to such data because the similarity measures defined on discrete values are usually not compatible with similarity of continuous values. Alternative methodologies such as probabilistic modeling, when applied to continuous data, require an extremely large amount of data.

In addition, poorly discretized attributes prevent classification systems from finding important inductive rules. For example, if the ages between 15 and 25 mapped into the same interval, it is impossible to generate the rule about the legal age to start military service. Furthermore, poor discretization makes it difficult to distinguish the non-predictive case from poor discretization. In most

P. Yolum et al.(Eds.): ISCIS 2005, LNCS 3733, pp. 493–502, 2005.

cases, inaccurate classification caused by poor discretization is likely to be considered as an error originated from the classification method itself. In other words, if the numeric values are poorly discretized, no matter how good our classification systems are, we fail to find some important rules in databases.

In this paper, we describe a new way of discretizing numeric attributes. We discretize the continuous values using a minimum loss of information criterion. Our discretization method is supervised one since it takes into consideration the class values of examples, and adopts information theory as a tool to measure the amount of information each interval contains. A number of typical machine learning data sets are selected for discretization, and these are discretized by both other current discretization methods and our proposed method. To compare the correctness of the discretization results, we use the naive Bayesian classifier and C4.5 as the classification algorithms to read and classify data.

The structure of this paper is as follows. Section 2 introduces some current discretization methods. In Section 3, we explain the basic ideas and theoretical background of our approach. Section 4 explains the brief algorithm and correctness of our approach, and experimental results of discretization using some typical machine learning data sets are shown in Section 5. Finally, conclusions are given in Section 6.

2 Related Work

Although discretization influences significantly the effectiveness of classification algorithms, not many studies have been done because it usually has been considered a peripheral issue. Among them, we describe a few well-known methods in machine learning literature.

A simple method, called equal distance method, is to partition the range between the minimum and maximum values into N intervals of equal width. Another method, called equal frequency method, chooses the intervals so that each interval contains approximately the same number of training examples; thus, if $N = 10$, each interval would contain approximately 10% of the examples. However, with both of these discretizations, it would be very difficult or almost impossible to learn certain concepts.

Some classification algorithms such as C4.5 [11] and PVM [13] take into account the class information when constructing intervals. For example, in C4.5, an entropy measure is used to select the best attribute to branch on at each node of the decision tree. And that measure is used to determine the best cut point for splitting a numeric attribute into two intervals. A threshold value, T, for the continuous numeric attribute A is determined, and the test $A \leq T$ is assigned to the left branch while $A > T$ is assigned to the right branch. This cut point is decided by exhaustively checking all possible binary splits of the current interval and choosing the splitting value that maximizes the entropy measure.

Fayyad [6] has extended the method of binary discretization in and C4.5 [11], and introduced multi-interval discretization, called Entropy Minimization Discretization(EMD), using minimum description length(MDL) technique. In

this method, the data are discretized into two intervals and the resulting class information entropy is calculated. A binary discretization is determined by selecting the cut point for which the entropy is minimal amongst all candidates. The binary discretization is applied recursively, always selecting the best cut point. A minimum description length criterion is applied to decide when to stop discretization. This method is implemented in this paper, and used in our exprimental study.

Fuzzy discretization(FD), proposed by Kononenko [8], initially forms k equal-width intervals using equal width discretization. Then it estimates $p(a_i < X_i \leq b_i | C = c)$ from all training instances rather than from instances that have value of X_i in (a_i, b_i). The influence of a training instances with value v of X_i on (a_i, b_i) is assumed to be normally distributed with the mean value equal to v. The idea behind fuzzy discretization is that small variation of the value of a numeric attribute should have small effects on the attribute's probabilities, whereas under non-fuzzy discretization, a slight difference between two values, one above and one below the cut point can have drastic effects on the estimated probabilities. The number of initial intervals k is a predefined parameter and is set as 7 in our experiments. This method is also implemented and used in our experimental study.

Khiops [3] proposes a discretization method using chi-square statistic. This method optimizes the chi-square criterion in a global manner on the whole discretization domain. It is a bottom-up method which starts with the discretization from the elementary single value intervals. It then evaluates all merges between adjacent intervals and selects the best one based on the chi-square criterion, and iterates.

Even though some algorithms use dynamic discretization methods, it might still be preferable to use static discretization. Using static discretization as a preprocessing step, we can see significant speed up for classification algorithm with little or no loss of accuracy [4]. The increase in efficiency is due to that the dynamic algorithm, such as C4.5/CART, must re-discretize all numeric attributes at every node in the decision tree while in static discretization all numeric attributes are discretized only once before the classification algorithm runs.

3 Hellinger-Based Discretization

It is seldom possible to verify that a given discretization is reasonable because a classification algorithm can hardly distinguish a non-predictive case from a poorly discretized attribute. In general, it is seldom possible to know what the correct or optimal discretization is unless the users are familiar with the problem domain. Another problem which complicates evaluation is that discretization quality depends on the classification algorithms that will use the discretization. Even though it is not possible to have an optimal discretization with which to compare results, some notion of quality is needed in order to design and evaluate a discretization algorithm.

The primary purpose of discretization, besides eliminating numeric values from the training data, is to produce a concise summarization of a numeric attribute. An interval is essentially a summary of the relative frequency of classes within that interval. Therefore, in an accurate discretization, the relative class frequencies should be fairly consistent within an interval(otherwise the interval should be split to express this difference) but two adjacent intervals should not have similar relative class frequencies(otherwise the intervals should be combined to make the discretization more concise). Thus, the defining characteristic of a high quality discretization can be summarized as: maximizing intra-interval uniformity and minimizing inter-interval uniformity.

Our method achieves this notion of quality by using an entropy function. The difference between the class frequencies of the target attribute and the class frequencies of a given interval is defined as *the amount of information* that the interval gives to the target attribute. The more different these two class frequencies are, the more information the interval gives to the target attribute. Therefore, defining an entropy function which can measure the degree of divergence between two class frequencies is crucial in our method and will be explained in the following.

3.1 Measuring Information Content

The basic principle of our discretization method is to discretize numeric values so that each discretized interval has as equal amount of information as possible. In other words, we define the amount of information that a certain interval contains as the degree of divergence between a priori distribution and a posteriori distribution of the target attribute. Therefore, the critical part of our method is to select or define an appropriate measure of the amount of information each interval gives to the target attribute.

In our approach, the interpretation of the amount of information is defined in the following. For a given interval, its class frequency distribution is likely to differ from that of the target attribute. The amount of information an interval provides is defined as the dissimilarity(divergence) between these two class frequencies. We employ an entropy function in order to measure the degree of divergence between these two class frequencies.

Some entropy functions have been used in this direction in machine learning literature. However, the purpose of these functions is different from that of ours. They are designed to decide the most discriminating attributes for generating decision trees [11]. Suppose X is the target attribute and it has k discrete values, denoted as $x_1, x_2, \ldots, x_k$. Let $p(x_i)$ denote the probability of x_i. Assume that we are going to discretize an attribute A with respect to the target attribute X. Suppose $A = a_i$ and $A = a_{i+1}$ are boundaries of an interval, and this interval is mapped into a discrete value a. Then the probability distribution of X under the condition that $a_i \leq A < a_{i+1}$ is possibly different from a priori distribution of X. We will introduce several studies for measuring divergence from machine learning literature and information theory literature.

In machine learning literature, C4.5 [11], which generates decision trees from data, has been widely used for rule induction. It uses the following formula, called information gain, for estimating the information given from $A = a$ about X.

$$H(X) - H(X|a) = \sum_t p(t) \log \left(\frac{1}{p(t)} \right) - \sum_t p(t|a) \log \left(\frac{1}{p(t|a)} \right) \quad . \tag{1}$$

It takes into consideration both a priori and a posteriori probabilities. It calculates the difference between the entropy of a priori distribution and that of a posteriori distribution, and uses the value to determine the most discriminating attribute of decision tree. However, it sometimes fails to calculate the divergence between two distributions correctly. Calculating the average value of each probability, it cannot detect the divergence of the distributions in the case that one distribution is a permutation of the other.

In information theory literature, several studies are done about divergence measure. Kullback [9] derived a divergence measure, called I-measure, defined as

$$\sum_i p(x_i|a) \log \frac{p(x_i|a)}{p(x_i)}. \tag{2}$$

Another group of divergence measure, widely used in information theory, includes Bhattacharyya divergence [2] and Renyi divergence [12].

However, since these measures are originally defined on continuous variables, there are some problems when these are applied to discrete values. These measures are not applicable in case one or more than one of the $p(x_i)$ are zero. Suppose that one class frequency of a priori distribution is unity and the rest are all zero. Similarly, one value of a posteriori distribution is unity and the rest are all zero. Then Kullback divergence, Renyi divergence and Bhattacharyya divergence are not defined in this case, and we cannot apply these directly without approximating the original values.

In this paper, we adopt Hellinger divergence [7] which is defined as

$$\left| \sum_i (\sqrt{p(x_i)} - \sqrt{p(x_i|a)})^2 \right|^{1/2} \quad . \tag{3}$$

It was originally proposed by Beran [1], and unlike other divergence measures, this measure is applicable to any case of probability distribution. In other words, Hellinger measure is continuous on every possible combination of a priori and a posteriori values. It can be interpreted as a distance measure where distance corresponds to the amount of divergence between a priori distribution and a posteriori distribution. It becomes zero if and only if both a priori and a posteriori distributions are identical, and ranges from 0 to $\sqrt{2}$. Therefore, we employ Hellinger divergence as a measure of divergence, which will be used as the information amount of intervals. The entropy of an interval I described above is defined as follows.

Definition 1. *The entropy of an interval I is defined as follows:*

$$E(I) = \left| \sum_i \left(\sqrt{p(x_i)} - \sqrt{p(x_i|I)} \right)^2 \right|^{1/2}. \tag{4}$$

4 Discretizing Algorithm

The algorithm consists of an initialization step and a bottom up combining process. As part of the initialization step, the training examples are sorted according to their values for the attribute being discretized and then each example becomes its own interval. The midpoint between each successive pair of values in the sorted sequence is called a potential *cutpoint*. Each cutpoint associates two adjacent intervals(or point values), and its corresponding entropy is defined as follows.

Definition 2. *The entropy of a cutpoint C, adjacent to interval a and b, is defined as follows.*

$$E(C) = E(a) - E(b). \tag{5}$$

If the class frequency of these two intervals are exactly the same, the cutpoint is called *in-class cutpoint*, and if not, the cutpoint is called *boundary cutpoint*. In other words, if two adjacent point values or intervals have different class frequencies, their midpoint(cutpoint) is defined as boundary cutpoint. Intuitively, discretization at in-class cutpoints are not desirable because it separates examples of one class. Therefore, boundary cutpoint must have high priority to be selected for discretization.

In combining process, the amount of information that each interval gives to the target attribute is calculated using Hellinger divergence. For each pair of two adjacent intervals, the system computes the informational difference between them. The least value of difference will be selected and its corresponding pair of intervals will be merged. Merging process continues until the system reaches the maximum number of intervals(k) usually given by users. The value of k, maximum number intervals, is determined by selecting a desired precision level the user wants. The standard recommended value of k is to set the value between 3 to 10 depending on the domain to prevent an excessive number of intervals from being created. Figure 1 shows the abstract algorithm of the discretization method.

We have the following theorem which shows the correctness of our discretization algorithm.

Theorem 1. *The in-class cutpoints are not to be selected for discretization unless all boundary cutpoints are exhausted for discretization.*

The proof is omitted due to space limit. This theorem implies that in our algorithm discretization keeps occurring only at boundary cutpoints unless it exhausts all boundary cutpoints. By doing so, it prevents the in-class cutpoints from being selected for discretization.

Input : a_1, a_2, ..., a_N (sorted and distinct numeric values)

$a_0 = a_1$; $a_{N+1} = a_N$;
K:=maximum number of interval;
/* Initialization step */
for i=1 **to** N **do**
    INTVL= $\{I_i = (p_i, q_i) | p_i = (a_{i-1} + a_i)/2, q_i = (a_i + a_{i+1})/2\}$;
end
/* Entropy of each interval */
for each $I_i \in$ INTVL **do**
    $E(I_i) = \left| \sum_j (\sqrt{P(a_j)} - \sqrt{P(a_j|I_i)})^2 \right|^{1/2}$;
end
/* Entropy of each cutpoint */
for i=1 **to** N-1 **do**
    $E(p_i) = E(I_i) - E(I_{i+1})$;
end
repeat N-K times **do**
    MERGE=cutpoint with least value of E;
    merge two intervals of MERGE;
end
return INTVL;

Fig. 1. Discretization Algorithm

The computational complexity of our discretization method is given as $O(n)$, where n is the number of examples.

Lemma 1. *Suppose n is the number of examples. The complexity of the proposed discretization method is given as*

$$O(n) \tag{6}$$

The proof of the lemma is trivial based on the pseudo code in Figure 1.

5 Empirical Results

Because our discretization method is not itself a classification algorithm it cannot be tested directly for classification accuracy, but must be evaluated indirectly in the context of a classification algorithm. Therefore, our discretization method will be used to create intervals for two well-known classification systems: naive Bayesian classifier and C4.5 [11].

In our experimental study, we compare our proposed method with Fuzzy Discretization(FD) [8], as a preprocessing step to the C4.5 algorithm and naive-Bayes classifier. C4.5 algorithm is a state-of-the-art method for inducing decision trees. The naive Bayes classifier computes the posterior probability of the classes given the data, assuming independence between the features for each class.

For the test data set, we have chosen eight datasets. Table 1 shows the datasets we chose for our comparison. These datasets are obtained from the

UCI repository [10] such that each had at least one continuous atribute. We used 10-fold cross-validation technique and, for each experiment, the training data are separately discretized into seven intervals by Fuzzy Discretization(FD) [8] and our proposed discretization method, respectively. The intervals so formed are separately applied to the test data. The experimental results are recorded as average classification accuracy that is the percentage of correct predictions of classification algorithms in the test across trials.

Table 2 shows the classification results of naive Bayes classifier using the different discretization methods. As we can see, our discretization method shows

Table 1. Description of datasets

Dataset	Size	Numeric	Categorical	Classes
Anneal	898	6	32	6
Breast	699	10	0	2
Glass	214	9	0	3
Hepatitis	155	6	13	2
Horse-colic	368	8	13	2
Hypothyroid	3163	7	18	2
Iris	150	4	0	3
Vehicle	846	18	0	4

Table 2. Classification results using naive Bayesian method

Dataset	FD	Proposed method
Anneal	92.3	89.2
Breast	96.3	97.2
Glass	64.8	68.1
Hepatitis	87.7	88.3
Horse-colic	81.5	78.4
Hypothyroid	97.2	97.0
Iris	94.7	96.6
Vehicle	59.6	62.8

Table 3. Classification results using C4.5

Dataset	FD	Proposed method
Anneal	89.2	87.3
Breast	91.5	95.8
Glass	69.2	70.1
Hepatitis	85.4	87.2
Horse-colic	81.5	82.7
Hypothyroid	98.8	97.3
Iris	95.6	96.3
Vehicle	62.7	66.4

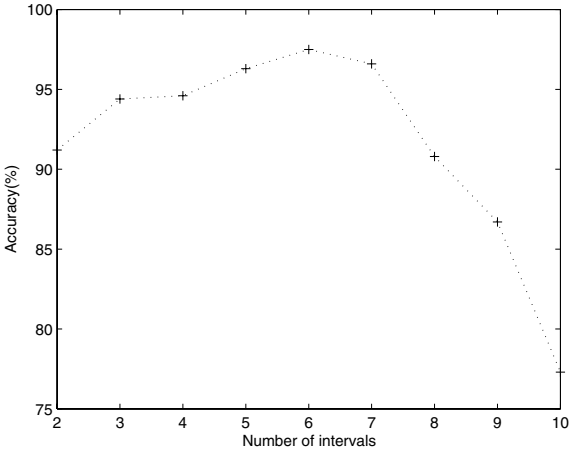

Fig. 2. Classification accuracy versus number of intervals

better results than other method in most data sets. In five cases among eight datasets, our method showed better classification accuracy.

Table 3 shows the results of classification for each data set using C4.5, and we can easily see that our discretization method shows the better classification accuracy in most cases. In six cases among eight datasets, our method showed the better classification accuracy.

Determining the right value of maximum number of intervals significantly effects the correctness of discretization. Too small number of intervals prevents important cutpoints from being discretized while too many cuts produce unnecessary intervals. In order to see the effect of the number of intervals, we applied naive Bayesian classifier to iris data set with different number of intervals, and the results are shown in Figure 2. For iris data set, when the attribute is discretized into 5-7 intervals, its classification result shows better accuracies while the number of intervals is greater than 7 or less than 5, the classification accuracy drops significantly.

6 Conclusion

In this paper, we proposed a new way of discretizing numeric attributes, considering class values when discretizing numeric values. Using our discretization method, the user can be fairly confident that the method will seldom miss important intervals or choose an interval boundary when there is obviously a better choice because discretization is carried out based on the information content of each interval about the target attribute. Our algorithm is easy to apply because all it requires for users to do is to provide the maximum number of intervals.

Our method showed better performance than other traditional methods in most cases. Our method can be applied virtually to any domain, and is applicable

to multi-class learning(i.e. domains with more than two classes–not just positive and negative examples).

Another benefit of our method is that it provides a concise summarization of numeric attributes, an aid to increasing human understanding of the relationship between numeric features and the class attributes.

One problem of our method is the lack of ability to distinguish between true correlations and coincidence. In general, it is probably not very harmful to have a few unnecessary interval boundaries; the penalty for excluding an interval is usually worse, because the classification algorithm has no way of making a distinction that is not in the data presented to it.

References

1. Beran R. J.: Minimum Hellinger Distances for Parametric Models, *Ann. Statistics*, Vol. 5 (1977) 445-463
2. Kadota T., Shepp L. A.: On the Best Finite Set of Linear Observables for discriminating two Gaussian signals, *IEEE Transactions on Information Theory*, Vol. 13 (1967) 278-284
3. Boulle M.: Khiops: A Statistical Discretization Method of Continuous Attributes, *Machine Learning*, Vol. 55 (2004) 53-69
4. Catlett J.: On changing continuous attributes into ordered discrete attributes. In *European Working Session on Learning* (1991)
5. Dougherty J., Kohavi R., Sahami M.: Supervised and Unsupervised Discretization of Continuous Features, 12th Int'l Conf. on Machine Learning (1995)
6. Fayyad U. M., Irani K. B.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, *13th International Joint Conference of Artificial Intelligence* (1993) 1022-1027
7. Ying Z.: Minimum Hellinger Distance Estimation for Censored Data, *The Annals of Statistics*, Vol. 20, No. 3 (1992)
8. Kononenko I.: Inductive and Bayesian Learning in Medical Diagnosis, *Applied Artificial Intelligence*, Vol. 7 (1993) 317-337
9. Kullback S.: *Information Theory and Statistics*, New York: Dover Publications (1968)
10. Murphy P. M., Aha D. W.: UCI repository of machine learning databases. http://www.ics.uci.edu/ mlearn (1996)
11. Quinlan J. R.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publisher (1993)
12. Renyi A.: On Measures of Entropy and Information, *Proceedings of Fourth Berkeley Symposium*, Vol. 1 (1961) 547-561
13. Weiss S. M., Galen R. S., Tapepalli P. V.: Maximizing the predictive value of production rules, *Artificial Intelligence*, Vol. 45 (1990) 47-71