CloNI: Clustering of $\sqrt{N}$ -Interval discretization

Chotirat "Ann" Ratanamahatana
Department of Computer Science,
University of California, Riverside, USA.

Abstract

It is known that naive Bayesian classifier typically works well on discrete data. All continuous attributes then need to be discretized beforehand for such applications. Inappropriate range of discretization intervals may result in degrading its performance. In this paper, we review previous work on continuous feature discretization and conduct an empirical evaluation of an improved method called Clustering of $\sqrt{N}$ -Interval Discretization (CloNI). CloNI tries to reduce the number of $\sqrt{N}$ intervals in the datasets by iteratively combining two consecutive intervals together, according to their median distance until a stopping criteria is met. We also show that even though C4.5 decision trees can handle continuous features, we can significantly improve its performance in some domains if those features were discretized in advance. In our empirical results, using discretized instead of continuous features in C4.5 never significantly degrades its accuracy. Our results indicate that CloNI reliably performs as well as or better than the Proportional k-interval Discretization (PKID) on all domains, and gives a competitive classification performance for both smaller and larger dataset.

1 Introduction

Discretization techniques are used to reduce the number of values for a given continuous attribute, by dividing the range of the attribute into intervals. Reducing the number of values for an attribute is especially beneficial if decision-tree-based methods of classification are to be applied to the pre-processed data. The reason is that these methods are typically recursive, and a large amount of time is spent on sorting the data at each step. Hence, the smaller the number of distinct values to sort, the faster these methods should be.

A simple and popular form of discretization is to use binning to approximate data distributions. One partitioning rule is *equiwidth* [3,4,9]; the width of each interval is uniform (10-bin is one example). Another is *equidepth*, where the intervals are created so that, roughly, each interval contains the same number of contiguous data samples. Proportional k-Interval Discretization (PKID) proposed by Yang & Webb [12] utilizes such equidepth method. It divides the values in an attribute into $\sqrt{N}$ intervals, each with approximately $\sqrt{N}$ data instances.

Fayyad & Irani has introduced an entropy minimization heuristic [6] in the context of decision tree learning. They applied a *Minimum Description Length* criterion to control the number of intervals produced over the continuous space. Entropy-based or any other types of supervised discretization that use class information has seemed to be superior in that the interval boundaries are defined to occur in places that may help improve classification accuracy. However, this paper will show that using a simpler "Unsupervised" discretization also has a competitive performance, and can yield high classification accuracy without any knowledge on the class distribution of the instances.

In this paper, we introduce CloNI (**Cl**ustering **o**f $\sqrt{N}$ - **I**nterval discretization) that tries to minimize the number of intervals obtained from PKID algorithm for each attribute while maintaining the high accuracy of the classification. Centroid-based Clustering technique is used to merge appropriate contiguous pair of intervals together, such that instances within an interval are "similar" or clustered together as much as possible.

To evaluate our new method, we separately implement the simple 10-bin discretization, Fayyad & Irani's discretization (Entropy), PKID, and our CloNI as pre-processing steps to train both Naive Bayesian classifiers and C4.5 decision trees. We compare the classification accuracies of the resulting classifiers according to the methods applied. We hypothesize that using CloNI for discretization will lead to improvement in classification accuracy of Naive Bayesian classifier and C4.5 decision trees for both smaller and larger datasets.

We give an overview and present work related to attribute discretization in Section 2, including our CloNI algorithm in detail. Experimental design and evaluation are presented in Section 3. Section 4 and 5 provide a discussion and conclusion of this work.

2 Discretization Methods

In this experiment, we consider four different discretization methods: equal width intervals, Proportional k-interval method (PKID) proposed by Yang & Webb [12], Entropy minimization heuristic proposed by Fayyad & Irani [6], and our proposed discretization method (CloNI).

2.1 Equal Width Interval Binning

Due to its mere simplicity, equal width interval binning is very popular and usually implemented in practice. The algorithm needs to first sort the attribute

according to its values, and then find the minimum value, x_{min}, and the maximum value, x_{max} of that attribute. Interval width, w, is then computed by

$$w = \frac{x_{max} - x_{min,}}{k},$$

where k is the user-defined parameter as the total number of intervals needed. The interval boundaries are specified as $x_{min} + w_i$, where $i = 1,...,k-1$. In our experiment, we use $k = 10$, representing the 10-bin discretization.

2.2 Entropy Minimization Heuristic

Fayyad & Irani's heuristic approach was developed in the context of decision tree learning that tries to identify a small number of intervals, each dominated by a single class. They first suggested binary discretization, which discretizes values of continuous attribute into two intervals. The training instances are first sorted in an increasing order, and the midpoint between each successive pair of attribute values is evaluated as a potential cut point. The algorithm selects the best cut point from the range of values by evaluating every cut point candidate. For each evaluation of a candidate, the data is discretized into two intervals and the entropy of the resulting discretization is computed. Given a set of instances S, a feature A, and a partition boundary T, the class information Entropy of the partition, $E(A, T; S)$, is given by:

$$E(A,T;S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2).$$

The boundary T_{min} that minimizes the entropy function over all possible partition boundaries is selected as a binary discretization boundary. This binary discretization is applied recursively until a stopping condition, *minimal description length,* is achieved.

For decision tree learning, minimizing the number of values of an attribute is important in avoiding the fragmentation problem [10]. If an attribute has many values, a split on this attribute will result in many branches, each of which receives relatively few training instances, making it difficult to select appropriate subsequent tests [12].

2.3 Proportional k-interval Discretization (PKID)

Yang & Webb [12] has proposed an algorithm that adjusts the number and size of discretized intervals to be proportional to the number of training instances as follows:

Given a continuous attribute A that has N training instances with *known* values for the attribute, PKID performs the following steps:

1. Sort a continuous attribute in an increasing order, then

2. Discretize the sorted values into $\sqrt{N}$ intervals, with $\sqrt{N}$ instances within each interval, according to the following rules.
   2.1. Any unknown values are to be included within a single interval. Thus, the actual number of instances in each interval may be more than $\sqrt{N}$ instances, and the number of intervals may be smaller than $\sqrt{N}$, depending on how many identical values there are in that attribute, and
   2.2. Smaller than $\sqrt{N}$ interval size is not allowed. The larger interval size is only allowed when identical values are present, or to accommodate the last interval.

2.4 CloNI: Clustering of $\sqrt{N}$ -Interval Discretization

When we are training the classifier that usually has a fixed number of training instances, the larger the size of the interval, the smaller the number of intervals, and vice versa. According to Kohavi and Wolpert [8], decreasing the number of intervals will increase bias while decrease variance, and increasing the number of intervals will decrease bias while increase variance. It would be best if we can find a good compromise between the two aspects according to each continuous attribute in the dataset, which is our purpose in this experiment.

In general, PKID algorithm will only perform well with smaller datasets or continuous attributes that have *many* identical and/or narrow ranges of values, which in turn produces much fewer intervals than $\sqrt{N}$ as specified. Otherwise, PKID tends to produce larger number of intervals when the dataset size increases, and this is the problem we are trying to alleviate.

We propose **Cl**ustering **o**f $\sqrt{N}$ -**I**nterval discretization (CloNI) that tries to minimize the number of intervals for each attribute while maintaining the high accuracy of the classification. Clustering technique is used to partition the objects into groups of clusters, so that instances within a cluster are "similar" to one another as much as possible.

2.4.1 Algorithm

1. Sort a continuous attribute in an increasing order.
2. Discretize the sorted values into $\lfloor\sqrt{N}\rfloor$ intervals, with $\lfloor\sqrt{N}\rfloor$ instances within each interval.
3. Calculate the median for each interval.
4. Merge an adjacent pair of interval together if they satisfy the given criteria, and then recompute the median for the new merged interval.
5. Repeat Steps 3 and 4 until no pair of intervals can be merged.

Figure 1: CloNI Algorithm

Figure 1 shows the algorithm for CloNI. We first sort the continuous attribute according to its values in an increasing order. We then discretize these sorted values into $\lfloor\sqrt{N}\rfloor$ intervals, with $\lfloor\sqrt{N}\rfloor$ instances within each interval, where N represents the number of instances within the attribute. If identical values are found overlapping between any two intervals, all the spillovers must be included in the former interval. In other words, each attribute value must belong to exactly one interval. As a result, the final number of intervals may be fewer than $\lfloor\sqrt{N}\rfloor$, or more than $\lfloor\sqrt{N}\rfloor$ instances may contain in a single interval. The next step is to compute the median for each interval, then merge any adjacent pair with their median distance, d_{median}, that satisfies the following criteria.

$$d_{median}(I_i, I_j) = |m_i - m_j| \leq \frac{median(A)}{l},$$

where m_i is the median for interval I_i, l denotes the number of *distinct* observed values for each attribute, A. The new median value for any merged interval is to be recomputed. The process continues until either no more d_{median} satisfies the criteria, or the number of intervals reaches $max\{1, 2 \cdot log\ l\}$. The $max\{1, 2 \cdot log\ l\}$ heuristic was chosen based on examining S-plus' histogram binning algorithm by Spector [11].

3 Experimental Evaluation

3.1 The Datasets

We have selected 15 natural datasets from the UCI KDD Archive [1] and the UCI machine-learning repository [2]. Table 1 shows the description of the datasets, including the number of continuous and nominal attributes.

Table 1. Description of domains used

Dataset	Size	Continuous	Nominal	Classes
Adult	48,842	6	8	2
Annealing	798	6	32	6
Census Income	299,285	7	33	2
Forest Covertype	581,012	10	44	7
German Credit	1,000	7	13	2
Glass	214	9	0	7
Heart Disease (Cleveland)	303	7	6	2
Hypothyroid	3,163	7	18	2
Ionosphere	351	34	0	2
Iris	150	4	0	3
Multiple Features	2,000	3	3	10

Dataset	Size	Continuous	Nominal	Classes
Pima Indians Diabetes	768	8	0	2
Postoperative Patient	90	1	7	3
Breast Cancer (Wisconsin)	699	9	0	2
Wine	178	13	0	3

3.2 Experimental Design

1. Each dataset is shuffled randomly to make sure that the class distribution in the training and test data are not biased or clustered in any form.
2. For each dataset, perform a 5-fold cross validation to train and test for Naive Bayesian Classifier and C4.5 decision tree, using the following discretization methods:
   - CloNI
   - PKID
   - Entropy, and
   - 10-bin

To evaluate the result in each dataset, the performance is measured by the average (mean) error (percentage in correct classification) in the testing data across all iterations. Note that all the discretization algorithms are only performed on training data. Discretization of attributes in testing data is done only according to the intervals created in the training process.

3.3 Experimental Results

Table 2 and 3 show the experimental results for CloNI, PKID, Entropy, and 10-bin discretization methods for Naive Bayesian Classifier and C4.5, respectively. The last rows of the tables give the mean accuracies for each discretization algorithm. The figures reported in boldface reflect the winning method on each dataset. The last column of table 3 shows the accuracies for each dataset when no discretization method is performed. The continuous attributes are used and discretized by C4.5. Note that all the C4.5 accuracies considered in this experiment are based on the simplified decision tree (with pruning). This accuracy is usually higher on the test (unseen) data, in comparison to the accuracy based on decision trees with no pruning. Table 4 shows the mean number of intervals produced by CloNI, PKID, and Entropy discretization methods.

Table 2. Discretization on Naive Bayesian Classifier

Dataset	Size	CloNI	PKID	Entropy	10-Bin
Adult	48,842	**85.1**	83.0	82.8	80.9
Annealing	798	97.0	95.3	**97.2**	92.4
Census Income	299,285	**77.1**	76.9	76.6	75.9
Forest Covertype	581,012	**68.7**	68.5	68.1	67.6
German Credit	1,000	**75.4**	74.9	74.9	74.6
Glass	214	**75.9**	75.9	74.9	74.7
Heart Disease (Cleveland)	303	82.7	82.7	82.8	**83.0**
Hypothyroid	3,163	**98.5**	98.1	98.4	97.5
Ionosphere	351	**89.9**	89.5	88.7	89.9
Iris	150	92.5	92.5	**93.4**	92.4
Multiple Features	2,000	**69.1**	68.8	67.2	68.1
Pima Indians Diabetes	768	**75.1**	74.0	74.1	74.3
Postoperative Patient	90	**64.1**	**64.1**	63.8	64.0
Breast Cancer (Wisconsin)	699	97.3	97.3	97.1	**97.5**
Wine	178	**98.2**	98.0	97.5	98.0
Mean	-	**83.1**	82.6	82.5	82.0

Table 3. Discretization methods on C4.5 Decision Trees

Dataset	Size	CloNI	PKID	Entropy	10-Bin	Continuous
Adult	48,842	**81.6**	81.5	81.2	80.6	80.7
Annealing	798	91.1	91.1	**91.6**	90.4	**91.6**
Census Income	299,285	**75.0**	74.9	74.5	74.0	73.8
Forest Covertype	581,012	**66.7**	66.5	65.0	64.3	65.2
German Credit	1,000	**73.5**	73.2	73.9	70.6	72.2
Glass	214	69.1	69.0	**69.2**	59.8	65.7
Heart Disease (Cleveland)	303	78.5	78.4	**79.1**	**77.5**	**73.7**
Hypothyroid	3,163	99.0	98.9	99.0	96.5	**99.1**
Ionosphere	351	**89.6**	89.5	88.7	89.9	87.8
Iris	150	94.4	94.4	94.3	**95.4**	94.5
Multiple Features	2,000	**67.9**	67.8	67.8	66.1	67.2
Pima Indians Diabetes	768	73.8	72.9	73.2	**74.1**	70.8
Postoperative Patient	90	**63.8**	**63.8**	62.6	**63.8**	62.4
Breast Cancer (Wisconsin)	699	93.4	93.3	93.4	92.8	**93.9**
Wine	178	**96.0**	95.9	95.9	94.7	95.7
Mean	-	**80.9**	80.7	80.6	79.4	79.6

Table 4. The mean number of intervals produced by different discretization methods

Dataset	Size	CloNI	PKID	Entropy
Adult	48,842	18	49	5
Annealing	798	3	5	3
Census Income	299,285	23	80	5
Forest Covertype	581,012	56	264	6
German Credit	1,000	6	9	2
Glass	214	4	8	3
Heart Disease (Cleveland)	303	5	8	2
Hypothyroid	3,163	12	27	4
Ionosphere	351	6	12	4
Iris	150	5	7	4
Multiple Features	2,000	11	35	6
Pima Indians Diabetes	768	5	16	3
Postoperative Patient	90	2	2	2
Breast Cancer (Wisconsin)	699	6	7	4
Wine	178	5	9	4
Mean	-	11.13	35.87	3.80

4 Discussion

Our experimental results reveal that our CloNI method leads to average increase in accuracy. Specifically, the best method, CloNI, improves performance on all but four relatively small datasets for Naive Bayesian Classifier. C4.5's performance was significantly improved on some datasets using CloNI method and did not significantly degrade on *any* dataset. Even though C4.5 can handle continuous attributes by doing its own discretization within, our experiment suggests that pre-discretizing the datasets before providing them to C4.5 actually can improve its classification accuracies. Especially, having fewer attribute values for the decision trees will make it learn faster. As Dougherty et. al. [5] pointed out, C4.5 induction algorithm itself may not take full advantage of possible local discretization that could be performed on the data or its local discretization could not help much in its induction process on the datasets we use in our experiment. From both Table 2 and 3, it is apparent that even though CloNI is not a winning method for every single dataset, it gives improvement on accuracies over PKID on *all* datasets, and gives the best classification accuracies on larger datasets. In addition, from Table 4, we can see that the number of intervals produced by CloNI is about 3 times fewer than what PKID would produce, and yet give higher accuracies that PKID. And even though the number of intervals produced by the Entropy method is relatively small, it does not always guarantee high classification accuracy. In particular, Entropy only beats CloNI in 2 out of 15 datasets for Naïve Bayesian Classifier and 3 out of 15 datasets for C4.5 decision trees.

5 Conclusion

In this paper, we reviewed some discretization approaches for Naive Bayesian Classifier and C4.5 decision trees, i.e. PKID, Entropy, and 10-bin. We then proposed a simple and improved discretization method, CloNI, which applies clustering method to the $\lfloor \sqrt{N} \rfloor$ intervals and tries to reduce the number of intervals down. We are persuaded that CloNI is more appropriate than PKID, Entropy, and 10-bin for Naive Bayesian Classifier and C4.5 in general because it gives importance to both the number of intervals as well as the number of instances within. It also adjusts them according to the training instances provided. CloNI algorithm tries to decrease the variance as dataset size increases, which reflects in improved performance on all larger datasets. Having fewer intervals is also very beneficial to the decision trees learning in that it provides faster learning time. Our experiment suggests that in comparison to its alternatives, CloNI provides both Naive Bayesian Classifiers and C4.5 decision trees improved performance especially on larger datasets.

References

[1] Bay, S. D. *The UCI KDD Archive [http://kdd.ics.uci.edu]*, Department of Information and Computer Science, University of California, Irvine, 1999.

[2] Blake, C. L., Merz, C. J. *UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]*, Department of Information and Computer Science, University of California, Irvine, 1998.

[3] Catlett, J. *On Changing Continuous Attributes into Ordered Discrete Attributes*. Proceedings of the European Working Session on Learning, pp. 164-178, 1991.

[4] Chmielewski, M.R., and Grzymala-Busse, J.W. *Global Discretization of Continuous Attributes as Preprocessing for Machine Learning*, 3rd International Workshop on Rough Sets and Soft Computing, pp. 294-301, 1994.

[5] Dougherty, J., Kohavi, R., and Sahami, M. *Supervised and unsupervised discretization of continuous features*, Proceedings of the 12th International Conference (ML '95). San Francisco, CA, Morgan Kaufmann, 1995.

[6] Fayyad, U.M. and Irani, K.B. *Multi-interval discretization of continuous-valued attributes for classification learning*, Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI '93), Morgan Kaufmann, pp.1022-1027, 1993.

[7] Hsu, C., Huang, H., and Wong, T. *Why Discretization Works for Naive Bayesian Classifiers*, In Proceedings of the 17th International Conference on Machine Learning (ICML-2000), Stanford, CA, USA. pp. 399-406, 2000.

[8] Kohavi, R., Wolpert, D. *Bias Plus Variance Decomposition for Zero-One Loss Functions*, Proceedings of the 13th International Conference on Machine Learning, pp.275-283, 1996.

[9] Pfahringer, B. *Compression-Based Discretization of Continuous Attributes*, Proceedings of the 12th International Conference on Machine Learning, 1995.

[10] Quinlan, J.R. *C4.5: Programs for Machine Learning*, Morgan Kaufmann, Los Altos, CA, 1993.

[11] Spector, P. *An Introduction to S and S-PLUS*, Duxbury Press, 1994.

[12] Yang, Y. and Webb, G. *Proportional k-Interval Discretization for Naive-Bayes Classifiers*. 12th European Conference on Machine Learning (ECML01), 2001.