

Improving Classification Accuracy with Discretization on Datasets Including Continuous Valued Features

Mehmet Hacıbeyoglu, Ahmet Arslan, Sirzat Kahramanli

Abstract—This study analyzes the effect of discretization on classification of datasets including continuous valued features. Six datasets from UCI which containing continuous valued features are discretized with entropy-based discretization method. The performance improvement between the dataset with original features and the dataset with discretized features is compared with k-nearest neighbors, Naive Bayes, C4.5 and CN2 data mining classification algorithms. As the result the classification accuracies of the six datasets are improved averagely by 1.71% to 12.31%.

Keywords—Data mining classification algorithms, entropy-based discretization method

I. INTRODUCTION

THERE are several applications for machine learning, the most significant of which is data mining. People are often prone to making mistakes when trying to establish relationships between multiple features. This situation makes it difficult to solve particular problems. Data mining classification algorithms developed to produce a solution to this difficulty and improving the efficiency of systems includes a lot of data. Classification is a widely used technique in various fields, including data mining whose goal is classify a large dataset into predefined classes. A dataset can be represented as $S = \{O, C \cup D\}$, where, $O = \{O_i\}_{i=1}^M$ is a finite set of objects, $C = \{A_j\}_{j=1}^N$ is a finite set of condition features and D is a decision feature. In a dataset the features may be continuous, categorical and binary. However, most features in real world are in continuous form. The abundance of continuous features constitutes a serious obstacle to the efficiency of most data mining classification algorithms. Because many of classification algorithms focus on to learn only in nominal feature spaces. Discretization is a typically pre-processing step for machine learning algorithms that transformed continuous-valued feature to discrete one [1]. The goal of discretization is to reduce the number of possible values a continuous attribute takes by partitioning them into a number of intervals [2]. After a discretization process, we can

achieve some important effects such as: the transformed feature is becoming a more meaningful figure; the necessary time for a classification algorithm is reducing and the performance of the classification is becoming more effective.

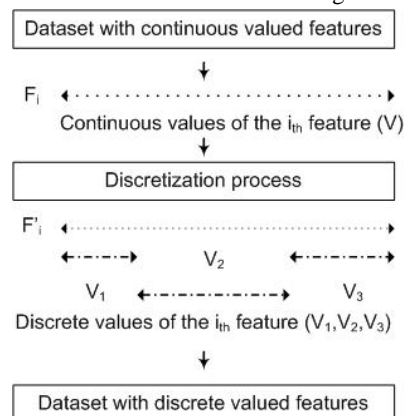


Fig. 1 Transformation of continuous valued features into discrete ones with discretization process

The idea of discretization is to divide the range of a numeric or ordinal attribute into intervals according to given cut points [3]. The cut points can be given directly by the user or can be computed. Fig.1 illustrates the transformation of continuous feature F_i into discrete feature F'_i with values $\{V_1, V_2$ and $V_3\}$. The Fig.2 describes the general steps of the discretization process.

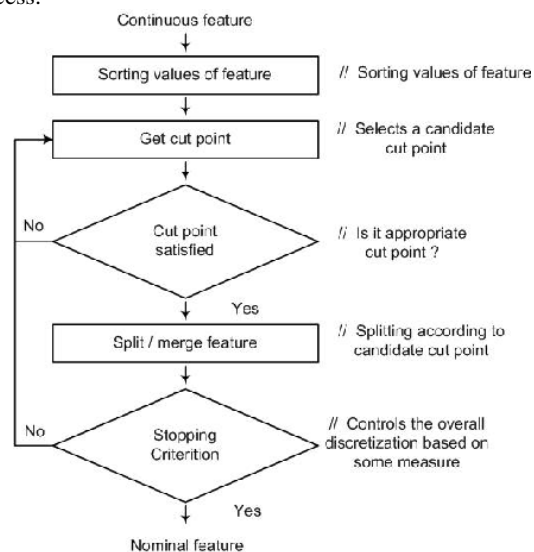


Fig. 2 The general steps of the discretization process

M. H. Author is with the Computer Engineering Department, University of Selcuk, Konya, TURKEY (phone: +90332-223-3333; fax: +90332-241-0665; e-mail: hacibeyoglu@selcuk.edu.tr).

A. A. Author is with the Computer Engineering Department, University of Selcuk, Konya, TURKEY (phone: +90332-223-2000; fax: +90332-241-0665; e-mail: ahmetarslan@selcuk.edu.tr).

S. K. Author is with the Computer Engineering Department, University of Mevlana, Konya, TURKEY (phone: +90332-444-4243; fax: +90332-241-1111; e-mail: sirzat@selcuk.edu.tr).

II. DISCRIMINATION METHODS

In the machine learning literature discretization methods have been categorized into two groups: supervised and unsupervised discretization. The first of the unsupervised discretization method is equal interval width discretization, where the range of observed values is divided into k internals of equal length as follows:

$$\forall i, j: V_i - V_{i-1} = V_j - V_{j-1} \quad (1)$$

The second unsupervised discretization method is equal frequency interval, where the range of observed values is divided into k bins such that the count all bins are equal as follows:

$$\forall i, j: V_i = V_j \quad (2)$$

The supervised discretization methods handle the class label repartition to achieve the different cuts and find the more appropriate intervals. Fayyad and Irani's [4] entropy-based discretization algorithm is arguably the most commonly used supervised discretization approach.

A. Entropy Based Discretization

The potential problems with the unsupervised discretization methods is the loss of classification information because of the resulting discretized feature values that are strongly associated with different classes in the same interval [5]. The supervised discretization methods handle sorted feature values to determine the potential cut points such that the resulting cut point has the strong majority of one particular class. The cut point for discretization is selected by evaluating the favorite disparity measure (i.e., class entropies) of candidate partitions. In entropy based discretization, the cut-point is selected according to the entropy of the candidate cut-points. Entropies of candidate cut-points are defined by following formulas:

$$E(F, T; X) = \frac{|X_1|}{|X|} Ent(X_1) - \frac{|X_2|}{|X|} Ent(X_2) \quad (3)$$

$$Ent(X_i) = -\sum_{i=1}^Z p(C_i, X_i) \log_2(p(C_i, X_i)) \quad (4)$$

In the formula (3), given a set of examples X is partitioned into two intervals X_1 and X_2 using the cut point T on the value of feature F . The entropy function Ent for a given dataset is calculated based on the class distribution of the samples in the set. The entropy of subsets X_1 and X_2 is calculated according to the formula 4, where $p(C_i, X_i)$ is the proportion of examples lying in the class C_i and Z is the total number of the classes. Among all the candidate cut points for $E(F, T; X)$, the best cut point T_F is selected, which has the minimum value of the entropy [6]. After this selection the values of the continuous-valued feature are splitting into two parts. This splitting procedure is recursively continued until a stopping criterion is reached. In entropy-based discretization method, the stopping criterion is defined by following formulas:

$$Gain(F, T; X) > \frac{\log_2(N-1)}{N} + \frac{\Delta(F, T; X)}{N} \quad (5)$$

$$Gain(F, T; X) = Ent(F) - E(F, T; X) \quad (6)$$

$$\Delta(F, T; X) = \log_2(3^Z - 2)$$

$$-[Z \cdot Ent(X) - Z_1 \cdot Ent(X_1) - Z_2 \cdot Ent(X_2)] \quad (7)$$

where, F is the feature which is going to be discretized, T is candidate cut point, X is the set of examples, X_1 and X_2 are the subsets of the split samples for the left and right part of X , respectively, N is the number of the samples in X , Z is the number of the classes in X , Z_1 and Z_2 are the numbers of the classes present in X_1 and X_2 , respectively.

III. EXPERIMENTAL SETTINGS

For experiments we choose 6 datasets from UCI with different characteristics such as: the number of attributes, the number of classes, the number of continuous values of the attributes and the number of examples.

TABLE I
THE PROPERTIES OF USED DATASETS

Dataset Name	Number of		
	Features	Examples	Classes
Statlog(Australian Credit Approval)	14	690	2
Statlog (Heart)	13	270	4
Ionosphere	34	351	2
Iris	4	150	3
Wine	13	178	3
Diabet	8	768	2

As classification algorithms, we used the algorithms K-nn [7] with 7 neighbors, Naive Bayes [8], C4.5 [9] and CN2 [10] without pruning. At the experimental stage, as the experimental methodology, we used cross-validation to estimate the accuracy of the classification algorithms [11]. More specifically, we used ten-fold cross-validation in which the dataset to be processed is permuted and partitioned equally into ten disjoint sets D_1, D_2, \dots, D_{10} . In each phase of a cross-validation, one of the yet unprocessed sets was tested, while the union of all remaining sets was used as training set for classification by the algorithms K-nn, C4.5, Naive Bayes and CN2.

A. K-nearest neighbor

K-nearest neighbor algorithm (K-nn) is a supervised learning algorithm that has been used in many applications in the field of data mining, statistical pattern recognition, image processing and many others. K-nn is a method for classifying objects based on closest training examples in the feature space. The k-neighborhood parameter is determined in the initialization stage of K-nn. The k samples which are closest to new sample are found among the training data. The class of the new sample is determined according to the closest k -samples by using majority voting [7]. Distance measurements like

Euclidean, Hamming and Manhattan are used to calculate the distances of the samples to each other.

B. Naïve Bayes classifier

A Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem with strong independence assumptions. The naive Bayes model is simple but effective and has been used in numerous applications of information processing including image recognition, natural language processing, and information retrieval. This model assumes conditional independence among features; it is possible to estimate its parameters from a limited amount of training data [8]. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule. The corresponding classifier is the function classify defined as follows:

$$\text{classify}(f_1, \dots, f_n) = \arg \max_c p(C = c) \prod_{i=1}^n p(F_i = f_i | C = c) \quad (8)$$

C. C4.5 classifier

C4.5 is a supervised learning classification algorithm used to construct decision trees from the data using the concept of information entropy [9]. Decision trees are composed of nodes, branches and leaves. Nodes are defined as features, branches are defined as values of features and leaves are defined as values of decision feature. Learned trees can also be represented as sets of *if-then* rules to improve human readability. In this method, the feature has maximum gain is determined as root node. Nodes having maximum gain within each subset are determined as sub nodes [12]. When each branch denotes a class, the creation of the tree is finished. Entropy and gain are defined as follows:

$$\text{Ent}(S) = - \sum_{i=1}^c p_i \log_2 p_i \quad (9)$$

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \quad (10)$$

where S is the set of samples, p_i is the proportion of S belonging to class i , $\text{Values}(A)$ is the set of all possible values

for attribute A , and S_v is the subset of S for which attribute A has value v .

D. CN2 classifier

Task of knowledge acquisition for expert systems is needed inducing concept descriptions from examples. CN2 [10] is a learning algorithm and developed for rule induction. CN2 can deal with the problems with poor described and/or noisy data. It creates a rule set like the way AQ[13] algorithm and deal with noisy data like ID3 [14] algorithm. The disadvantage of the AQ algorithm is it needs specific examples. The CN2 algorithm removes this dependence of AQ algorithm and expands the spaces of rules searched. This lets statistical techniques which used for tree pruning to be applied in the if-then rule creation phase and leads simpler induction algorithm.

IV. EXPERIMENTAL RESULTS

To estimate the performance of the discretization method, we compared the results generated entropy based discretization method with the results generated by original sets of attributes for chosen datasets. In the experiments, we used a target machine with an Intel Core2Quad@2.83 GHz processor and 2 GB memory, running on Microsoft Windows 7 OS. The datasets with original features and discretized form of the dataset are classified with k-NN, Naive Bayes, C4.5 and CN2 data mining classification algorithms. Both of the obtained classification results are compared.

We obtained the classification accuracy for a certain dataset as average of the accuracies of the mentioned ten phases. The average percentage of the accuracy rate increase ψ per example provided by the proposed method was obtained by the formula (11).

$$\psi = \frac{\sum_{i=1}^P \phi_i N_i}{\sum_{i=1}^P N_i} \times 100\% \quad (12)$$

where, ϕ_i is the accuracy increase for the dataset W_i , N_i is the number of examples in the dataset W_i and P is the number of the datasets used in the experiments. In Table 2 are given the results of classification of these datasets by using the original feature set and the discretized form of the dataset separately.

TABLE II
CLASSIFICATION RESULTS

Dataset	Classification Accuracy								The increase ϕ_i in the classification accuracy			
	With the original features				With the discretized features							
	K-nn	Naive Bayes	C4.5	CN2	K-nn	Naive Bayes	C4.5	CN2	ϕ_i^{K-nn}	ϕ_i^{NB}	$\phi_i^{C4.5}$	ϕ_i^{CN2}
Statlog(Australian Credit Approval)	0.575	0.868	0.814	0.842	0.800	0.862	0.813	0.852	0,225	-0,006	-0,001	0,01
Statlog (Heart)	0.570	0.804	0.763	0.759	0.830	0.833	0.841	0.830	0,260	0,029	0,078	0,071
Ionosphere	0.863	0.878	0.892	0.909	0.923	0.934	0.875	0.932	0,060	0,056	-0,017	0,023
Iris	0.960	0.920	0.940	0.947	0.947	0.933	0.953	0.933	-0,013	0,013	0,013	-0,014
Wine	0.765	0.972	0.922	0.927	0.983	0.978	0.950	0.978	0,218	0,006	0,028	0,051
Diabet	0.681	0.756	0.737	0.742	0.698	0.780	0.763	0.776	0,017	0,024	0,026	0,034

The average percentage of the accuracy rate increase ψ achieved by the algorithms K-nn, Naive Bayes, C4.5 and CN2 for the datasets given in Table 2 are as follows:

$$\psi_{K-nn} = \frac{\sum_{i=1}^{10} \phi_i^{K-nn} N_i}{\sum_{i=1}^{10} N_i} \times 100\% = 12.314;$$

$$\psi_{NB} = \frac{\sum_{i=1}^{10} \phi_i^{NB} N_i}{\sum_{i=1}^{10} N_i} \times 100\% = 1.861;$$

$$\psi_{C4.5} = \frac{\sum_{i=1}^{10} \phi_i^{C4.5} N_i}{\sum_{i=1}^{10} N_i} \times 100\% = 1.716;$$

$$\psi_{CN2} = \frac{\sum_{i=1}^{10} \phi_i^{CN2} N_i}{\sum_{i=1}^{10} N_i} \times 100\% = 2.793.$$

V.CONCLUSION

In this paper, entropy -based discretization method is used for improving the classification accuracy for datasets including continuous valued features. In the first phase, the continuous valued features of the given dataset are discretized. Second phase, we tested the performance of this approach with the popular algorithms such as K-nn, Naive Bayes, C4.5 and CN2. The discretization approach increased the classification ability of K-nn algorithm approximately 12.3%. Unfortunately, this approach cannot significantly improve the classification ability of Naive Bayes, C4.5 and CN2 algorithms like K-nn.

REFERENCES

- [1] J. L. Lustgarten, V. Gopalakrishnan, H. Grover, and S. Visweswaran, "Improving Classification Performance with Discretization on Biomedical Datasets," in *AMIA Annu Symp Proc.*, 2008, pp.445–449.
- [2] K. J. Cios, W. Pedrycz, R. Swiniarski and L. Kurgan, "Data Mining A Knowledge Discovery Approach," *Springer*, 2007.
- [3] A. Kumar, D. Zhang, "Hand-Geometry Recognition Using Entropy-Based Discretization," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 2, 2007, pp. 181-187.
- [4] U. M. Fayyad, K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proc. 13th International Joint Conference on Artificial Intelligence*, San Francisco, CA, Morgan Kaufmann, 1993, pp. 1022–1027.
- [5] D. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. 12th Int. Conf. Machine Learning*, Tahoe City, CA, 1995, pp. 194–202.
- [6] I. H. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations," San Mateo, CA: Morgan Kaufman, 1999.
- [7] G. Shakhnarovich, T. Darrell and P. Indyk, "Nearest-Neighbor Methods in Learning and Vision," *MIT Press*, 2005.
- [8] Y. Tsuruoka and J. Tsujii, "Improving the performance of dictionary-based approaches in protein name recognition," *Journal of Biomedical Informatics*, vol. 37, no. 6, December, 2004, pp. 461-470
- [9] J. R. Quinlan, "C4.5: Programs for machine learning," San Francisco, CA: Morgan Kaufman, 1993.
- [10] P. Clark and T. Niblett, "The CN2 induction algorithm," *Machine Learning*, 1989, vol. 3, pp. 261-284.
- [11] N. Mastrogiannis, B. Boutsinas and I. Giannikos, "A method for improving the accuracy of data mining classification algorithms," *Computers & Operations Research*, 2009, vol. 36 no.10, pp. 2829-2839.
- [12] J. R. Quinlan, "Induction of C4.5 Decision trees," *Machine Learning*, vol. 1, 1986, pp. 81–106.
- [13] R. S. Michalski, "On the quasi-minimal solution of the general covering problem," in *Proceedings of the Fifth International Symposium on Information Processing*, 1969, Bled, Yugoslavia, pp. 125-128.
- [14] J. R. Quinlan, "Learning efficient classification procedures and their application to chess end games," *Machine learning: An artificial intelligence approach*, 1983, Los Altos, CA: Morgan Kaufmann.