A Multiple Scanning Strategy for Entropy Based Discretization

Jerzy W. Grzymala-Busse[1,2]

[1] Department of Electrical Engineering and Computer Science,
University of Kansas, Lawrence, KS 66045, USA
[2] Institute of Computer Science,
Polish Academy of Sciences, 01-237 Warsaw, Poland
jerzy@ku.edu

Abstract. We present results of experiments performed on 14 data sets with numerical attributes using a novel technique of discretization called multiple scanning. Multiple scanning is based on scanning all attributes of the data set many times, during each scan the best cut-points are found for all attributes. Results of our experiments show that multiple scanning enhances successfully, in terms of the error rate, an ordinary discretization technique based on conditional entropy.

Keywords: Rough sets, multiple scanning, entropy based discretization, LEM2 rule induction algorithm, LERS data mining system.

1 Introduction

Mining data sets with numerical attributes requires a special technique called discretization, i.e., converting numerical values into intervals [7]. Discretization is usually performed before the main process of knowledge acquisition. There are many techniques of discretization, however discretization based on conditional entropy is considered to be one of the most successful techniques [1,3,4,5,7,9,10,11,13,14]. Therefore, in this paper, we selected for our experiments a method of discretization based on conditional entropy [3,5].

Let us start from some fundamental ideas of discretization. For a numerical attribute a with an interval $[a, b]$ as a range, a partition of the range into k intervals

$$\{[a_0, a_1), [a_1, a_2), ..., [a_{k-2}, a_{k-1}), [a_{k-1}, a_k]\},$$

where $a_0 = a$, $a_k = b$, and $a_i < a_{i+1}$ for $i = 0, 1, ..., k-1$, defines a discretization of a. The numbers a_1, a_2,..., a_{k-1} are called *cut-points*. Our discretization system denotes such intervals as $a_0..a_1$, $a_1..a_2$,..., $a_{k-1}..a_k$.

Discretization methods in which attributes are processed one at a time are called *local* [3,7] (or *static* [4]). On the other hand, if all attributes are considered in selection of the best cut-point, the method is called *global* [3,7] (or *dynamic* [4]). Additionally, if information about the expert's classification of cases is taken into account during the process of discretization, the method is called *supervised* [4].

J. Rauch et al. (Eds.): ISMIS 2009, LNAI 5722, pp. 25–34, 2009.

2 Entropy Based Discretization

An entropy of a variable v (attribute or decision) with values v_1, v_2,..., v_n is defined by the following formula

$$E_v(U) = -\sum_{i=1}^{n} p(v_i) \cdot \log p(v_i),$$

where U is the set of all cases in a data set and $p(v_i)$ is a probability (relative frequency) of value v_i in the set U, $i = 0, 1, ..., n$. All logarithms in this paper are binary.

Table 1. An example of a data set with numerical attributes

Case	Attributes			Decision
	A	B	C	D
1	0.3	−0.2	12.2	a
2	0.3	0.4	12.2	b
3	0.3	0.4	14.6	c
4	1.1	−0.2	14.6	c
5	1.1	1.8	14.6	c
6	1.5	1.8	14.6	d
7	1.5	1.8	20.2	e

A conditional entropy of the decision d given an attribute a is

$$E(d|a) = -\sum_{j=1}^{m} p(a_j) \cdot \sum_{i=1}^{n} p(d_i|a_j) \cdot log\ p(d_i|a_j),$$

where $a_1, a_2, ..., a_m$ are all values of a and $d_1, d_2, ..., d_n$ are all values of d. There are two fundamental criteria of quality based on entropy. The first is an *information gain* associated with an attribute a, denoted by $I(a)$, and equal to

$$E_d(U) - E(d|a)$$

the second is *information gain ratio*, for simplicity called *gain ratio*, defined by

$$\frac{I(a)}{E_a(U)}.$$

For a cut-point q for an attribute a the conditional entropy, defined by a cut-point q that splits the set U of all cases into two sets, S_1 and S_2 is defined as follows

$$E_a(q, U) = \frac{|S_1|}{|U|} E_a(S_1) + \frac{|S_2|}{|U|} E_a(S_2),$$

where $|X|$ denotes the cardinality of the set X. The cut-point q for which the conditional entropy $E_a(q, U)$ has the smallest value is selected as the best cut-point.

2.1 Starting from One Attribute

We will discuss two basic discretization techniques based on entropy. The first discretization technique is called *starting from one attribute*. Initially, we identify the best attribute (i.e., the attribute with the largest information gain or the attribute with the largest gain ratio). For the best attribute, we are looking for the best cut-point, i.e., the cut-point with the smallest conditional entropy. The best cut-point divides the data set into two smaller data sets, S_1 and S_2. We apply the same strategy for both smaller data sets separately. However, we need to take into account that discretization of one of the smaller data sets may affect the other. We will illustrate this method by discretizing the data set from Table 1. We will use the information gain as the criterion to select the best attribute.

The conditional entropy $E(D|A)$ is

$$\frac{3}{7}(3)(-\frac{1}{3} \cdot \log \frac{1}{3}) + \frac{2}{7} \cdot 0 + \frac{2}{7}(2)(-\frac{1}{2} \cdot \log \frac{1}{2}) = 0.965.$$

Similarly, the conditional entropies $E(D|B) = 1.250$ and $E(D|C) = 0.749$. The minimal conditional entropy is associated with attribute C. The next question is what is the best cut-point for attribute C. This attribute has two potential cut-points (averages between sorted values of the attribute C): 13.4 and 17.4. The conditional entropy $E_C(13.4, U)$ is

$$\frac{2}{7}(2)(-\frac{1}{2} \cdot \log \frac{1}{2} + \frac{5}{7}(-\frac{3}{5} \cdot \log \frac{3}{5} - \frac{1}{5}(2) \cdot \log \frac{1}{5}) = 1.265,$$

similarly, the conditional entropy $E_C(17.4, U) = 1.536$. Thus we will select the cut-point 13.4. Obviously, the current discretization of attribute C into two intervals 12.2..13.4 and 13.4..20.2 is not sufficient, since if we will use only discretized attribute C our data set will be inconsistent, i.e., there will be conflicting cases. The current discretization partitions Table 1 into two subtables, Tables 2 and 3.

Table 2. The first subtable of Table 1

Case	Attributes			Decision
	A	B	C	D
1	0.3	−0.2	12.2	a
2	0.3	0.4	12.2	b

Table 3. The second subtable of Table 1

Case	Attributes			Decision
	A	B	C	D
3	0.3	0.4	14.6	c
4	1.1	−0.2	14.6	c
5	1.1	1.8	14.6	c
6	1.5	1.8	14.6	d
7	1.5	1.8	20.2	e

It is also obvious that for Table 2 the only attribute that may be discretized is B, with the cut-point equal to 0.1. Table 4 presents the current situation: discretized are attributes B and C, with cut-points 0.1 and 13.4, respectively.

Table 4. Table 1 with discretized attributes B and C once

Case	Attributes		Decision
	B	C	D
1	−0.2..0.1	12.2..13.4	a
2	0.1..1.8	12.2..13.4	b
3	0.1..1.8	13.4..20.2	c
4	−0.2..0.1	13.4..20.2	c
5	0.1..1.8	13.4..20.2	c
6	0.1..1.8	13.4..20.2	d
7	0.1..1.8	13.4..20.2	e

Table 4 is not consistent, so Table 1 needs further discretization. However, by analysis of Table 4 we may easily discover that all what we need to do is to distinguish cases 3 and 5 from cases 6 and 7 and that cases 3 and 4 do not need to be distinguished. Thus, our next table to be discretized is presented as Table 5 (note that Table 5 is simpler than Table 3). We will continue discretization by recursion. Our final choice of cut-points is 1.3 for A, 0.1 for B, and 13.4 and 17.4 for C.

2.2 Multiple Scanning Strategy

The second discretization technique is based on scanning the set of attributes some fixed number of times and selecting for each attribute the best cut-point during each scan. After such scanning, if the discretized decision table needs more discretization, the first technique (starting from one attribute) is used. We will illustrate this technique by scanning all attributes, A, B, and C once. First

Table 5. Table that still need discretization

Case	Attributes			Decision
	A	B	C	D
3	0.3	0.4	14.6	c
5	1.1	1.8	14.6	c
6	1.5	1.8	14.6	d
7	1.5	1.8	20.2	e

we are searching for the best cut-point for attributes A, B, and C. The best cut-points are 1.3, 1.1, and 13.4, respectively. The discretized table is presented as Table 6.

Table 6. Table 1 discretized by scanning all attributes once

Case	Attributes			Decision
	A	B	C	D
1	0.3..1.3	−0.2..1.1	12.2..13.4	a
2	0.3..1.3	−0.2..1.1	12.2..13.4	b
3	0.3..1.3	−0.2..1.1	13.4..20.2	c
4	0.3..1.3	−0.2..1.1	13.4..20.2	c
5	0.3..1.3	1.1..1.8	13.4..20.2	c
6	1.3..1.5	1.1..1.8	13.4..20.2	d
7	1.3..1.5	1.1..1.8	13.4..20.2	e

Table 6 is not consistent, we need to distinguish cases 1 and 2, and, separately, cases 6 and 7. Therefore we need to us *starting from one attribute* technique for two tables, first with two cases, 1 and 2, and second with also two cases, 6 and 7. As a result we will select cut-points 0.1 and 17.4 for attributes B and C, respectively.

2.3 Stopping Condition for Discretization

In experiments discussed in this paper, the stopping condition was the level of consistency [3], based on rough set theory introduced by Z. Pawlak in [12]. Let U denote the set of all cases of the data set. Let P denote a nonempty subset of the set of all variables, i.e., attributes and a decision. Obviously, set P defines an equivalence relation $\wp$ on U, where two cases x and y from U belong to the same equivalence class of $\wp$ if and only if both x and y are characterized by the same values of each variable from P. The set of all equivalence classes of $\wp$, i.e., a partition on U, will be denoted by P^*.

Equivalence classes of $\wp$ are called *elementary sets* of P. Any finite union of elementary sets of P is called a *definable set* in P. Let X be any subset of U. In general, X is not a definable set in P. However, set X may be approximated by two definable sets in P, the first one is called a *lower approximation of X in P*, denoted by $\underline{P}X$ and defined as follows

$$\bigcup \{Y \in P^* | Y \subseteq X\}.$$

The second set is called an *upper approximation of X in P*, denoted by $\overline{P}X$ and defined as follows

$$\bigcup \{Y \in P^* | Y \cap X \neq \emptyset\}.$$

The lower approximation of X in P is the greatest definable set in P, contained in X. The upper approximation of X in P is the least definable set in P containing X. A *rough set of X* is the family of all subsets of U having the same lower and the same upper approximations of X.

A *level of consistency* [3], denoted L_c, is defined as follows

$$L_c = \frac{\sum_{X \in \{d\}^*} |\underline{A}X|}{|U|}.$$

Practically, the requested level of consistency for discretization is 100%, i.e., we want the discretized data set to be *consistent*.

2.4 Interval Merging

The next step of discretization was merging intervals, to reduce their number and, at the same time, preserve consistency. Merging of intervals begins from *safe merging*, where, for each attribute, neighboring intervals labeled by the same decision value are replaced by their union. The next step of merging intervals was based on checking every pair of neighboring intervals whether their merging will result in preserving consistency. If so, intervals are merged permanently. If not, they are marked as un-mergeable. Obviously, the order in which pairs of intervals are selected affects the final outcome. In our experiments, we selected two neighboring intervals with the smallest total conditional entropy, taking all attributes into account. Using interval merging we may eliminate the cut-point 1.1 for attribute B, computed as a result of scanning Table 1 once.

2.5 Rule Induction: LEM2 Algorithm

The data system LERS (Learning from Examples based on Rough Sets) [6] induces rules from incomplete data, i.e., data with missing attribute values, from data with numerical attributes, and from inconsistent data, i.e., data with conflicting cases. Two cases are conflicting when they are characterized by the same values of all attributes, but they belong to different concepts (classes). LERS uses rough set theory to compute lower and upper approximations for concepts involved in conflicts with other concepts [12].

Rules induced from the lower approximation of the concept *certainly* describe the concept, hence such rules are called *certain*. On the other hand, rules induced from the upper approximation of the concept describe the concept *possibly*, so these rules are called *possible*.

The LEM2 algorithm (Learning from Examples Module, version 2) of LERS is most frequently used for rule induction. LEM2 explores the search space of attribute-value pairs. Its input data set is a lower or upper approximation of a concept, so its input data set is always consistent. In general, LEM2 computes a local covering and then converts it into a rule set [2,6]. Recently, a new, improved version of LEM2, called MLEM2, was developed [8].

Table 7. Data sets

Data set	Number of		
	cases	attributes	concepts
Australian	690	14	2
Bankruptcy	66	5	2
Bupa	345	6	2
Connectionist Bench	208	60	2
Echocardiogram	74	7	2
Ecoli	336	8	8
Glass	214	9	6
Image Segmentation	210	19	7
Ionosphere	351	34	2
Iris	150	4	3
Pima	768	8	2
Wave	512	21	3
Wine	178	13	3
Yeast	1484	8	9

3 Experiments

Our experiments were conducted on 14 data sets, summarized in Table 7. All of these data sets, with the exception of *bankruptcy*, are available on the University of California at Irvine *Machine Learning Repository*. The bankruptcy data set is a well-known data set used by E. Altman to predict a bankruptcy of companies.

Every discretization method was applied to every data set, with the level of consistency equal to 100%. For a choice of the best attribute, we used gain ratio. Rule sets were induced using the LEM2 algorithm of the LERS data mining system.

Table 8 presents results of ten-fold cross validation, for all 14 data sets, using increasing number of scans. Obviously, for any data set, after some fixed number of scans, an error rate is stable (constant). For example, for *Australian* data set, the error rate will be 15.65% for the scan number 4, 5, etc. Thus, and data set from Table 8 is characterized by two error rates: minimal and stable. For a given data set, the smallest error rate from Table 8 will be called *minimal* and the last entry in the row that corresponds to the data set will be called *stable*. For example, for the *Australian* data set, the minimal error rate is 14.93% and the stable error rate is 15.65%. For some data sets (e.g., for *bankruptcy*), minimal and stable error rates are identical.

Table 8. Error rates for discretized data sets

Data set	Error rate for scan number						
	0	1	2	3	4	5	6
Australian	34.49	15.22	14.93	15.65			
Bankruptcy	3.03	9.09	1.52				
Bupa	31.30	29.28	30.14	26.67			
Connectionist Bench	29.33	27.88					
Echocardiogram	24.32	16.22					
Ecoli	19.64	20.54	18.75	20.83	21.43	20.54	20. 83
Glass	24.77	34.58	20.56	25.70	24.77	25.70	26.64
Image Segmentation	29.52	19.52	16.19	17.14			
Ionosphere	10.83	6.27	9.69	7.12			
Iris	5.33	2.67	4.67				
Pima	27.21	26.04	25.65	26.30	26.82	26.69	26.43
Wave	27.10	19.53	20.70	19.53	24.77	19.53	
Wine	11.24	2.81					
Yeast	56.74	50.47	48.99	48.92	51.28	52.83	

It is clear from Table 8 that the minimal error rate is never associated with 0 scans (i.e., with the method *starting from one attribute*). Using the Wilcoxon matched-pairs signed-ranks test, we conclude that the following two statements are statistically highly significant (i.e., the significance level is equal to 1% for a two-tail test):

- the minimal error rate is associated with scanning the entire attribute set at least once,
- the stable error rate is smaller than the error rate associated with the *starting from one attribute* discretization technique.

Additionally, effects of scanning during discretization are presented in Table 9. Note that some data sets, e.g., *Australian*, have binary attributes. For such data

Table 9. Number of intervals for scanning data set *bankruptcy*

Attribute	Number of scans					
	0		1		2	
	before merging	after merging	before merging	after merging	before merging	after merging
a1	9	9	4	3	3	2
a2	1	1	2	2	3	2
a3	2	2	2	1	3	2
a4	1	1	2	2	2	2
a5	1	1	2	2	2	1

sets, scanning will not change the number of intervals for binary attributes. We selected *bankruptcy* data set not only because its all attributes are numerical with real numbers as values but also since it has only five attributes. For 0 scans, i.e., for *starting from one attribute*, it is clear that attribute $a1$ was selected as the best and that during discretization eight cut-points were selected. After the single scan, the same attribute was selected as the best attribute, hence two additional cut-points were selected for $a1$. With two scans, for the first three attributes two cut-points were selected, as expected, for the last two attributes, $a4$ and $a5$, only single cut-points were found since the discretized table was already consistent, second cut-points for $a4$ and $a5$ would be redundant.

4 Conclusions

Our paper presents results of experiments in which scanning was used during discretization of 14 data sets with numerical attributes. Our discretization techniques were combined with rule induction using the LEM2 rule induction algorithm. As a result, we conclude that results of discretization based on scanning the attribute set at least once are significantly better (with a significance level of 1%, two-tailed test) than the results of discretization based on starting from one attribute. Thus, we proved that there exists an additional technique for improving discretization.

References

1. Blajdo, P., Grzymala-Busse, J.W., Hippe, Z.S., Knap, M., Mroczek, T., Piatek, L.: A comparison of six approaches to discretization—A rough set perspective. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 31–38. Springer, Heidelberg (2008)
2. Chan, C.C., Grzymala-Busse, J.W.: On the attribute redundancy and the learning programs ID3, PRISM, and LEM2. Department of Computer Science, University of Kansas, TR-91-14 (1991)

3. Chmielewski, M.R., Grzymala-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. Int. Journal of Approximate Reasoning 15, 319–331 (1996)
4. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: 12th International Conference on Machine Learning, pp. 194–202. Morgan Kaufmann Publishers, San Francisco (1995)
5. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. Machine Learning 8, 87–102 (1992)
6. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. Fundamenta Informaticae 31, 27–39 (1997)
7. Grzymala-Busse, J.W.: Discretization of numerical attributes. In: Klösgen, W., Zytkow, J. (eds.) Handbook of Data Mining and Knowledge Discovery, pp. 218–225. Oxford University Press, New York (2002)
8. Grzymala-Busse, J.W.: MLEM2: A new algorithm for rule induction from imperfect data. In: 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 243–250. ESIA Annecy, France (2002)
9. Grzymala-Busse, J.W., Stefanowski, J.: Three discretization methods for rule induction. Int. Journal of Intelligent Systems 16, 29–38 (2001)
10. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. Data Mining and Knowledge Discovery 6, 393–423 (2002)
11. Nguyen, H.S., Nguyen, S.H.: Discretization methods for data mining. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 451–482. Physica-Verlag, Heidelberg (1998)
12. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
13. Pensa, R.G., Leschi, C., Besson, J., Boulicaut, J.F.: Assessment of discretization techniques for relevant pattern discovery from gene expression data. In: Proc. of the 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics, pp. 24–30 (2004)
14. Stefanowski, J.: Handling continuous attributes in discovery of strong decision rules. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 394–401. Springer, Heidelberg (1998)