SPID4.7: Discretization Using Successive Pseudo Deletion at Maximum Information Gain Boundary Points

Somnath Pal* Himika Biswas†

Abstract

Discretization is the process of converting the continuous attributes of the database into discrete ones in order to apply some classification algorithms. This is an important problem in developing generally applicable methods in machine learning and data mining for classification and prediction. This paper intorduces a new technique for discretization based on successive pseudo deletion of instances to reduce the conflicting instances, i.e., by reduction of noise in the database. Such successive pseudo deletions in the database are performed by introducing threshold points on maximum information gain boundary points of the continuous attributes. Our empirical experiments show that the state of the art algorithms for learning, such as CN2, C4.5, Naive-Bayes and RISE give improvement in performances with the discretized output from our method than outputs with other state of the art discretization algorithms.

Keywords: Machine Learning, Continuous attribute, Data pre-processing, Discretization, Classification and Prediction.

1 Introduction.

Many algorithms developed in machine learning, data mining and uncertain reasoning tasks can not handle continuous features [1], [4]. To use them on real-world data sets, continuous attributes must first be discretized into small number of distinct ranges. Also discretization provides a kind of insight into critical values in a continuous attribute. Furthermore, the response time of many classifier inducing algorithms, such as RISE [6], increase if the data is continuous.

Much work has been done in discretization of continuous valued attributes [3], [9], [11], etc. In [7] a valuable systematic reveiw of a number of works on discretization have been carried out. It has been found there that Fayyad and Irani's algorithm based on Minimum Description Length Principle (MDLP) [9] achieved best overall results. The same superiority of MDLP discretizer was further established in [8].

In this work we introduce a discretization method based on a novel concept called successive pseudo deletion (for noise reduction) and show that it achieves a favorably competitive performance compared to MDLP discretization method. Our experimental evaluation is carried on 20 data sets and using CN2 [5], C4.5 [14], Naive-Bayes [7], and RISE [6] algorithms which are used for classification and prediction. In §2, we present our algorithm, SPID4.7 (Selective Pseudo Iterative Deletion 4.7) , that uses the new concept - pseudo deletion. Then in §3 we include details of experimental design for comparative empirical evaluation of our algorithm with MDLP discretizer. Next, in §4, we give comparative results of our discretization with the MDLP and also in-built (local) discretization methods of state of the art machine learning algorithms. This is followed by §5 where a discussion on related works for data pre-processing by discretization and SPID4.7's standing with respect to them is included. The conclusion is summarized finally in §6.

2 Discretization by Pseudo-Deletion.

Our algorithm (SPID4.7), based on successive pseudo deletion at maximum information gain boundary points works on iterative noise reduction of the database. In the next subsection we have presented the algorithm SPID4.7 using the method of successive pseudo deletion.

2.1 SPID4.7 discretization algorithm. We veiw the problem of discretization from a new angle. When there is no discretization of continuous attributes, the whole instance table consists of highest number of conflicting instances, i.e., instances with same set of attribute values but different class values. On the other extreme, if all continuous attributes are discretized at all possible continuous values, then there are minimum number of conflicting instances (actually no conflicting instances if there is no noise in the training data). Our successive pseudo deletion approach of discretization introduces threshold points in different continuous attributes of the database globally at maximum infor-

*Department of Computer Science & Technology, Bengal Engineering & Science University, Shibpur, Howrah - 711103, India. Email: *sp@becs.ac.in*

†Department of Computer Science & Technology, Bengal Engineering & Science University, Shibpur, Howrah - 711103, India. Email: *himika@cs.becs.ac.in*

INPUT: Set of instances (S) with attributes stored in table attb[ ][ ] (where element $attb[i][j]$ is the value of the j^{th} attribute in the i^{th} instance), and table class[ ] (where class[i] stands for the class value of the i^{th} instance). Also m, user-set minimum instances constraint within two discretized ranges.
OUTPUT: Discretized instance set in table attb[ ][ ].

```
begin
    for each continuous attribute in S
            insert threshold point at the boundary point having maximum information gain;
    find pseudo_deletion_count (PDEL_count) by majority vote;
    while (PDEL_count ≠ 0) and (all boundary points have not yet been considered) do
    begin
        for each continuous attribute
              select the boundary point which has maximum information gain and calculate PDEL_reqd;
        find the min_PDEL_reqd among the above selected points;
        if min_PDEL_reqd < PDEL_count
        then begin
                accept the selected boundary point as threshold point;
                PDEL_count ←min_PDEL_reqd;
                remove boundary point(s) violating m on both sides of selected threshold point;
             end;
        else begin
              for each continuous attribute
                    select the bounday point which has minimum information gain and calculate PDEL_reqd;
              find the max_PDEL_reqd among the above selected points;
              if max_PDEL_reqd > PDEL_count then reject the point(s) with max_PDEL_reqd;
              else
                    reject the point(s) with max_PDEL_reqd among the maximum gain points of each continuous attribute;
        end; /*of else*/
    end; /* of while */
end.
```

Table 1:
SPID4.7 algorithm.

mation gain points one by one such that the noise in the database gets reduced with introduction of each threshold point.

Any discretization algorithm has got two parts: (1) (threshold point) selection criterion and (2) stopping criterion. The threshold point selection criterion in SPID4.7 are mainly made to reduce noise at successive steps conditional on information gain (or entropy). The stopping criterion of SPID4.7 are simple: either the noise in the data base is zero or the noise is minimum on the path traversed by the greedy algorithm and there is no more boundary point [9] left to be considered as probable threshold point.

Based on the above, binary threshold points are added to all continuous attributes at the point where information gain of the each attribute is maximum among its boundary points. Then if there are conflicting instances we determine how many instances are to be deleted (not actually deleted) to reduce the conflict in the data set according to majority voting.

After incorporating binary threshold points to all continuous attributes, in case there are conflicting instances, we choose the maximum gain point for each continuous attribute and then temporarily incorporate a threshold point there. Then we calculate the number of instances to be deleted (pseudo deletion required), according to majority voting. If the minimum pseudo deletion required among these selected points is less than the previous pseudo deletion required then the point is accepted as a threshold point. Otherwise, if the maximum pseudo deletion required among the minimum gain points of all continuous attributes is greater than the previous pseudo deletion required then that (those) boundary point(s), where pseudo deletion required is equal to the maximum, is (are) rejected, i.e., not considered as boundary point(s) any more. However, if the maximum pseudo deletion required among minimum gain points is less than or equal to the previous pseudo deletion required then we reject the maximum pseudo deletion required point(s) among the maximum gain points of each continuous attribute. If a boundary point is accepted as a threshold point then the previous pseudo deletion required is replaced by the pseudo deletion required calculated after accepting the

Data set	# E	# A	# CL	M.V.
anneal(ann)	798	$6C, 14D$	6	yes
australian(aus)	690	$6C, 8D$	2	no
credit(cre)	690	$6C, 9D$	2	yes
dermatology(der)	330	$1C, 33D$	6	yes
echocardio(ech)	132	$8C, 3D$	3	yes
ecoli(eco)	336	$7C, 0D$	8	no
glass(gla)	214	$9C, 0D$	7	no
hungary(hun)	294	$5C, 8D$	5	yes
heart-statlog(sta)	270	$5C, 8D$	2	no
switzerland(swi)	123	$4C, 8D$	5	yes
horse-colic(hor)	300	$7C, 15D$	2	yes
imports-85(imp)	201	$15C, 10D$	7	yes
iris(iri)	150	$4C, 0D$	3	no
liver-disorder(liv)	345	$6C, 0D$	2	no
machine(mac)	209	$7C, 0D$	30	no
newthyroid(thy)	214	$5C, 0D$	3	no
pima(pim)	768	$8C, 0D$	2	no
vehicle(veh)	94	$18C, 0D$	4	no
wine(win)	178	$13C, 0D$	3	no
wisconsin(wis)	699	$9C, 0D$	2	yes

Table 2:
Characteristics of Data sets.
E - Examples, A - Attributes, CL - Classes, M.V. - Missing Values.
C - #Continuous Attributes, D - #Discrete Attributes.

selected boundary point. The process is repeated until either the pseudo deletion required becomes zero (i.e., there is no noise in the data set) or there is no boundary point left to be examined.

Some authors like [14] used a constraint on the minimum number of instances in each of the ranges, which means that any given range may include a mixture of class values. In SPID4.7, this minimum instance(s) constraint between ranges is called m and it is an user-set value. When a boundary point is selected as a threshold point, if there are any other unselected boundary point(s) violating the minimum instance(s) constraint at either side of it then the unselected boundary point(s) is (are) rejected. The complete algorithm of SPID4.7 is shown in Table 1.

3 Experimental Design.

To empirically evaluate the performance of SPID4.7 algorithm, experiments are performed on 20 real-world data sets drawn from the University of California at Irvine data repository [2]. The characteristics of the data sets are shown in Table 2.

For empirical evaluation of SPID4.7 we have chosen four well known state of the art classification algorithms: CN2 [5], C4.5 [14], Naive-Bayes [7], and RISE [6]. Missing values for both MDLP and SPID4.7 algorithms are replaced by most frequent value of the attribute.

Each of CN2, C4.5, Naive-Bayes and RISE was

Data Set	in-built discretizer $acc. \pm s.d.$	MDLP $acc. \pm s.d.$	SPID4.7 $acc. \pm s.d.$
ann	86.59 ± 4.39	92.83 ± 2.44	91.67 ± 2.96^{6}
aus	81.79 ± 4.70	81.06 ± 3.97	81.32 ± 4.18^{6}
cre	82.10 ± 4.37	79.78 ± 4.61	80.57 ± 4.75^{5}
der	86.53 ± 7.02	90.83 ± 4.64	90.66 ± 4.37^{6}
ech	62.06 ± 13.03	60.91 ± 13.22	61.51 ± 11.83^{6}
eco	78.99 ± 6.28	79.87 ± 6.17	73.99 ± 6.81^{1}
gla	66.12 ± 8.85	67.07 ± 8.66	68.19 ± 9.54^{6}
hun	62.81 ± 9.45	64.90 ± 8.57	64.71 ± 7.75^{6}
sta	77.28 ± 8.66	76.39 ± 6.61	76.17 ± 7.55^{6}
swi	36.30 ± 12.50	33.39 ± 10.64	38.00 ± 13.20^{4}
hor	75.06 ± 8.14	71.26 ± 7.93	71.20 ± 8.94^{6}
imp	75.80 ± 9.64	74.99 ± 8.84	71.11 ± 10.15^{3}
iri	93.33 ± 5.65	94.39 ± 4.68	96.66 ± 4.06^{2}
liv	66.60 ± 7.59	67.59 ± 7.26	69.57 ± 7.42^{5}
mac	43.45 ± 11.47	38.65 ± 11.32	47.97 ± 11.89^{1}
thy	94.40 ± 5.38	95.60 ± 4.34	95.42 ± 3.88^{6}
pim	74.20 ± 4.58	71.98 ± 5.73	72.95 ± 5.42^{6}
veh	60.20 ± 17.37	72.99 ± 12.27	72.41 ± 12.83^{6}
win	93.25 ± 5.61	94.28 ± 6.28	96.85 ± 4.77^{1}
wis	94.25 ± 2.61	93.76 ± 3.02	94.56 ± 2.76^{5}

Table 3:
Accuracy Comparisons Using CN2 algorithm.
Empirical Results: acc.=average accuracy and s.d.=standard deviation.
Superscripts denote confidence levels comparing MDLP and SPID4.7: 1 is 99.5%, 2 is 99%, 3 is 97.5%, 4 is 95%, 5 is 90%, and 6 is below 90%.

Data Set	in-built discretizer $acc. \pm s.d.$	MDLP $acc. \pm s.d.$	SPID4.7 $acc. \pm s.d.$
ann	93.96 ± 2.58	92.76 ± 2.66	92.98 ± 2.52^{6}
aus	83.95 ± 4.30	86.20 ± 3.66	85.61 ± 4.05^{6}
cre	85.04 ± 4.02	85.85 ± 4.54	85.76 ± 3.79^{6}
der	92.11 ± 5.69	91.99 ± 5.68	92.11 ± 5.69^{6}
ech	64.28 ± 13.35	65.12 ± 11.86	68.62 ± 11.64^{5}
eco	81.84 ± 4.82	80.65 ± 6.11	82.45 ± 6.06^{5}
gla	66.06 ± 8.19	73.67 ± 8.16	72.24 ± 8.20^{6}
hun	62.08 ± 8.42	64.98 ± 8.62	61.05 ± 10.85^{4}
sta	77.80 ± 8.04	78.91 ± 7.33	80.24 ± 7.42^{6}
swi	34.43 ± 15.19	36.24 ± 12.60	40.95 ± 12.81^{4}
hor	83.74 ± 6.03	84.19 ± 5.93	82.13 ± 7.13^{5}
imp	78.67 ± 9.87	77.68 ± 9.85	78.68 ± 9.31^{6}
iri	95.59 ± 5.09	95.46 ± 5.40	96.66 ± 4.06^{6}
liv	66.73 ± 6.96	69.22 ± 7.67	67.37 ± 7.57^{6}
mac	45.23 ± 11.71	43.84 ± 11.44	47.09 ± 11.64^{5}
thy	93.46 ± 5.13	94.68 ± 5.01	94.21 ± 4.79^{6}
pim	74.56 ± 5.06	75.37 ± 5.13	76.26 ± 5.44^{6}
veh	63.50 ± 14.09	68.06 ± 15.51	68.35 ± 16.07^{6}
win	93.49 ± 5.53	96.38 ± 4.47	96.30 ± 4.47^{6}
wis	95.41 ± 2.26	96.07 ± 2.29	95.39 ± 2.71^{5}

Table 4:
Accuracy Comparisons Using C4.5 algorithm.
Empirical Results: acc.=average accuracy and s.d.=standard deviation.
Superscripts denote confidence levels comparing MDLP and SPID4.7: 1 is 99.5%, 2 is 99%, 3 is 97.5%, 4 is 95%, 5 is 90%, and 6 is below 90%.

Data Set	in-built discretizer *acc.* ± *s.d.*	MDLP *acc.* ± *s.d.*	SPID4.7 *acc.* ± *s.d.*
ann	93.86 ± 2.60	92.96 ± 2.91	93.38 ± 3.04[6]
aus	73.88 ± 5.62	85.42 ± 3.76	86.41 ± 3.83[6]
cre	74.03 ± 5.32	86.29 ± 3.90	86.87 ± 3.90[6]
der	97.52 ± 2.07	98.42 ± 1.94	98.48 ± 1.84[6]
ech	57.10 ± 11.98	73.55 ± 10.39	76.46 ± 11.90[6]
eco	66.49 ± 9.05	83.86 ± 5.79	86.84 ± 5.09[1]
gla	40.93 ± 9.83	74.70 ± 8.23	76.61 ± 7.36[6]
hun	59.54 ± 9.04	67.08 ± 9.17	68.31 ± 9.61[6]
sta	70.74 ± 8.55	83.56 ± 7.52	83.56 ± 7.63[6]
swi	24.09 ± 14.04	43.10 ± 12.66	45.67 ± 12.78[6]
hor	77.27 ± 7.01	78.87 ± 6.13	79.07 ± 6.16[6]
imp	73.84 ± 9.76	81.47 ± 7.85	70.94 ± 9.23[1]
iri	91.60 ± 5.30	94.53 ± 4.93	95.07 ± 4.58[6]
liv	58.02 ± 7.61	66.32 ± 7.77	70.34 ± 8.22[2]
mac	63.06 ± 9.24	47.68 ± 11.12	53.94 ± 10.32[1]
thy	91.81 ± 6.91	95.71 ± 4.06	98.14 ± 3.06[1]
pim	65.52 ± 4.39	77.13 ± 4.45	77.94 ± 3.90[6]
veh	44.40 ± 13.98	57.18 ± 14.83	57.40 ± 13.88[6]
win	70.60 ± 9.38	99.33 ± 2.12	99.22 ± 2.23[6]
wis	97.40 ± 1.85	97.14 ± 2.13	97.25 ± 2.06[6]

Table 5:
Accuracy Comparisons Using Naive-Bayes algorithm.
Empirical Results: acc.=average accuracy and s.d.=standard deviation.
Superscripts denote confidence levels comparing MDLP and SPID4.7: 1 is 99.5%, 2 is 99%, 3 is 97.5%, 4 is 95%, 5 is 90%, and 6 is below 90%.

Data Set	in-built discretizer *acc.* ± *s.d.*	MDLP *acc.* ± *s.d.*	SPID4.7 *acc.* ± *s.d.*
ann	98.67 ± 1.29	93.93 ± 2.81	94.79 ± 2.32[5]
aus	83.13 ± 4.45	85.42 ± 4.13	86.46 ± 3.74[5]
cre	83.16 ± 4.34	85.94 ± 3.79	87.01 ± 3.45[5]
der	95.94 ± 3.35	95.39 ± 3.69	96.00 ± 3.50[6]
ech	71.09 ± 11.75	71.96 ± 9.99	72.86 ± 10.34[6]
eco	83.86 ± 5.51	83.62 ± 4.96	83.92 ± 5.87[6]
gla	72.48 ± 8.27	78.78 ± 7.68	79.79 ± 7.44[6]
hun	63.68 ± 7.83	65.22 ± 8.74	67.15 ± 9.73[6]
sta	81.56 ± 7.57	78.74 ± 8.57	79.41 ± 8.29[6]
swi	33.64 ± 14.96	39.99 ± 12.39	47.60 ± 13.50[1]
hor	84.73 ± 6.08	82.40 ± 6.83	83.40 ± 6.78[6]
imp	75.20 ± 8.99	83.25 ± 9.03	80.07 ± 10.04[5]
iri	96.00 ± 5.16	94.67 ± 4.99	96.80 ± 3.83[3]
liv	62.46 ± 8.31	70.26 ± 7.95	71.67 ± 7.44[6]
mac	57.14 ± 10.28	49.38 ± 10.16	57.21 ± 9.77[1]
thy	95.54 ± 4.96	95.89 ± 4.02	97.39 ± 3.51[4]
pim	72.39 ± 4.81	74.84 ± 4.44	75.86 ± 4.85[6]
veh	58.31 ± 15.40	70.78 ± 16.21	73.84 ± 14.02[6]
win	96.75 ± 4.79	96.97 ± 4.37	99.56 ± 1.87[1]
wis	96.51 ± 1.86	96.02 ± 2.31	95.02 ± 2.67[4]

Table 6:
Accuracy Comparisons Using RISE algorithm.
Empirical Results: acc.=average accuracy and s.d.=standard deviation.
Superscripts denote confidence levels comparing MDLP and SPID4.7: 1 is 99.5%, 2 is 99%, 3 is 97.5%, 4 is 95%, 5 is 90%, and 6 is below 90%.

run on each of the original data set with continuous values discretized by respective in-built local discretizer while inducing rules. On the otherhand, the original data sets are first discretized using SPID4.7 and the discretized data was run using CN2, C4.5, Naive-Bayes and RISE. Similarly, for comparisons, the original data sets are discretized using MDLP [9] algorithm and the discretized data were run using the same four algorithms. The accuracy of the induced rules were calculated using 10-fold cross-validation method. Such 10-fold cross-validation experiments were repeated 5 times.

4 Results and Empirical Comparisons.

Table 3 shows empirical results for CN2 algorithm run on its in-built discretizer, with data discretized by MDLP and SPID4.7. Similarly Table 4, Table 5 and Table 6 shows empirical results with C4.5, Naive-Bayes and RISE algorithms respectively in the same format. Although Tables 3-6, showing the accuracies obtained in individual data sets, are interesting in themselves, some interesting features of the results are summarized in Table 7. The results of Table 7 can be interpreted as follows.

Mean: The pre-processed discretized data from SPID4.7 generates higher mean across all domains for all four algorithms (CN2, C4.5, Naive-Bayes, and RISE) compared to their in-built (or locally) discretized data and MDLP discretization algorithm.

Number of wins: An obvious test is simply to compare the number of data sets where discretized output by SPID4.7 achieved higher accuracy compared to MDLP and in-built discretized output. As we can see from Table 7, in case of CN2, its in-built discretizer produces better accuracies for 8 data sets, whereas discretized output of SPID4.7 produces better accuracies for 12 data sets. Again SPID4.7 produces better accuracies for 11 data sets, whereas MDLP produces better accuracies in 9 data sets. Similarly, for C4.5, SPID4.7 wins over its in-built discretizer in 15 out of 20 data sets (1 draw) and over MDLP in 11 out of 20. In case of Naive-Bayes, SPID4.7 wins over its in-built discretizer in 16 out of 20 data sets and over MDLP in 17 out of 20 (1 draw). For RISE, SPID4.7 wins over its in-built discretizer in 16 out of 20 data sets and over MDLP in 18 out of 20.

Significant Win: A better alternative compared to the above is to count only those data sets where the difference was significant at a confidence level of 95% or higher (see Tables 3-6, where superscripts show the results of one-tailed paired t-test performed with MDLP and SPID4.7). This still shows improved results in favor of SPID4.7 discretization for all algorithms except for C4.5 algorithm, where the results of MDLP and SPID4.7 are the same.

Wilcoxon Signed-rank test: In our case, the result of the signed-rank test support that SPID4.7 produces better discretized output than CN2 with confidence of 81.71%. In case of C4.5, SPID4.7 produces better discretization compared to MDLP discretization with confidence 78.52% and in both cases of Naive-Bayes and RISE, SPID4.7 generates better outputs with confidence 99.97%.

Criteria	in-built discretizer	MDLP	SPID4.7
Algorithm CN2			
Mean	73.06	75.13	75.77
in-built(W-D-L)	–	$10-0-10$	$12-0-8$
MDLP(W-D-L)	$10-0-10$	–	$11-0-9$
Significant Win	–	2	4
Wilcoxon Test	–	–	89.71%
Algorithm C4.5			
Mean	74.41	77.87	78.22
in-built(W-D-L)	–	$14-0-6$	$15-1-4$
MDLP(W-D-L)	$6-0-14$	–	$11-0-9$
Significant Win	–	1	1
Wilcoxon Test	–	–	78.52%
Algorithm Naive-Bayes			
Mean	68.68	79.22	80.10
in-built(W-D-L)	–	$17-0-3$	$16-0-4$
MDLP(W-D-L)	$3-0-17$	–	$17-1-2$
Significant Win	–	1	4
Wilcoxon Test	–	–	99.97%
Algorithm RISE			
Mean	75.91	79.67	81.29
in-built(W-D-L)	–	$12-0-8$	$16-0-4$
MDLP(W-D-L)	$8-0-12$	–	$18-0-2$
Significant Win	–	1	5
Wilcoxon Test	–	–	99.97%

Table 7:
Summary of Accuracy Results.
W=Win, D=Draw, L=Loss. X(W-D-L) under the column Y means win-draw-loss of Y compared to X.
Significance Test and Wilcoxon Signed-Rank Test compare MDLP with SPID4.7.

5 Related Works.

SPID4.7, presented in this paper, is a supervised and global method of discretization; whereas MDLP [9] discretization is a supervised and local method of discretization. The ChiMerge system [11], is another local method that provides a statistically justified heuristic method for supervised discretization. Another statistical discretization method Khiops [3], based on chi-square statistics, have recently been published. There, an empirical comparative study of a number of discretization methods based on Naive-Bayes algorithm on 15 data sets was carried out. In that study, Equal Frequency method was ranked higher than ChiMerge and Equal Width, whereas Khiops along with MDLP discretizer was ranked above Equal Frequency method. From the empirical study carried out in the §4 of this paper, we see that SPID4.7 compares favorably with MDLP discretizer.

6 Conclusion.

We have presented a discretization algorithm based on a new concept – successive pseudo iterative deletion at maximum information gain boundary points, that can be used for generating pre-processed data for state of the art machine learning algorithms for data mining tasks. Empirical evaluation of our algorithm SPID4.7 has shown that discretized data generated by it is better than state of the art discretization algorithm MDLP, which has been ranked very high as a discretizer along with Khiops [3] in a recent study.

It is a well-accepted opinion that there is no one superior discretization method that will give best results across all domains. However, it has been observed in [10], that discretization methods based on conditional entropy, perform very well. The success of SPID4.7 can possibly be attributed to the fact that its threshold point selection criteria are conditional (successive noise reduction in database) on maximum information gain boundary points. Furthermore, unlike MDLP and ChiMerge, SPID4.7 is a global discretization algorithm, which may have contributed to its better performance than the others. However, we feel that there is scope for further improvement of SPID4.7.

Future work on this paper involves time complexity analysis of SPID4.7. Other direction of future research includes the applicability of SPID4.7 on large databases. Finally, SPID4.7 can be made available on request to the second author.

Acknowledgement

The authors are grateful to D. Chakraborty for implementation of Naive-Bayes algorithm and A. M. Ghosh for many fruitful discussions on this work. The authors are also grateful to P. Clark, J. R. Quinlan and P. Domingos for making the source codes of CN2, C4.5 and RISE systems respectively available for this work.

References

[1] C. Apte and S. Hong, *Predicting equity returns from security data*, Advances in Knowledge Discovery and Data Mining, AAAI Press and the MIT press, chapter 22, pp. 541–569, 1996.

[2] C. L. Blake and C. J. Merz, *UCI repository of machine learning databases (Machine-readable data repository)*, (http://www.ics.uci.edu/~mlearn/MLRepository.html) Department of Information and Computer Science, University of California, Irvine, 1999.

[3] M. Boulle, *Khiops: A statistical discretization method of continuous attributes*, Machine Learning, 55 (2004), pp. 53–69.

[4] J. Cendrowska, *PRISM: An algorithm for inducing modular rules*, International Journal for Man-Machine Studies, 27 (1987), pp. 349–370.

[5] P. Clark and R. Boswell, *Rule Induction with CN2: some recent improvements*, Machine Learning: Proceedings of the Fifth European Conference, Berlin, pp. 151–163, 1991.

[6] P. Domingos, *Unifying instance-based and rule-based induction*, Machine Learning, 3 (1996), pp. 139–168.

[7] J. Dougherty, R. Kohavi, and M. Sahami, *Supervised and unsupervised discretization of continuous features*, in Proceedings of the Twelfth International Conference on Machine Learning, A. Prieditis and S. Russell, eds., Morgan Kaufmann, San Francisco, pp. 194–202, 1995.

[8] T. Elomaa and J. Rousu, *General and efficient multisplitting of numerical attributes*, Machine Learning, 36/3 (1999), pp. 1–49.

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

[10] J. W. Grzymala-Busse, *Discretization of numerical attributes*, in Handbook of Data Mining and Knowledge Discovery, W. Klösgen and J. M.Zytkow, eds., Oxford University Press, pp. 218–225, 2002.

[11] R. Kerber, *ChiMerge: Discretization of numeric attributes*, in Proceedings of the Tenth National Conference on Artificial Intelligence, MIT Press, pp. 123–128, 1992.

[12] J. R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann, 1993.