

# A Novel Rule Ordering Approach in Classification Association Rule Mining

Yanbo J. Wang<sup>1</sup>, Qin Xin<sup>2</sup>, and Frans Coenen<sup>1</sup>

<sup>1</sup> Department of Computer Science, The University of Liverpool,  
Ashton Building, Ashton Street, Liverpool, L69 3BX, UK  
{jwang, frans}@csc.liv.ac.uk

<sup>2</sup> Department of Informatics, University of Bergen,  
P.B.7800, N-5020 Bergen, Norway  
xin@ii.uib.no

**Abstract.** A Classification Association Rule (CAR), a common type of mined knowledge in Data Mining, describes an implicative co-occurring relationship between a set of binary-valued data-attributes (items) and a pre-defined class, expressed in the form of an “antecedent  $\Rightarrow$  consequent-class” rule. Classification Association Rule Mining (CARM) is a recent Classification Rule Mining (CRM) approach that builds an Association Rule Mining (ARM) based classifier using CARs. Regardless of which particular methodology is used to build it, a classifier is usually presented as an ordered CAR list, based on an applied rule ordering strategy. Five existing rule ordering mechanisms can be identified: (1) Confidence-Support-size\_of\_Antecedent (CSA), (2) size\_of\_Antecedent-Confidence-Support (ACS), (3) Weighted Relative Accuracy (WRA), (4) Laplace Accuracy, and (5)  $\chi^2$  Testing. In this paper, we divide the above mechanisms into two groups: (i) pure “support-confidence” framework like, and (ii) additive score assigning like. We consequently propose a hybrid rule ordering approach by combining one approach taken from (i) and another approach taken from (ii). The experimental results show that the proposed rule ordering approach performs well with respect to the accuracy of classification.

**Keywords:** Classification Association Rules, Classification Association Rule Mining, Data Mining, Rule Ordering.

## 1 Introduction

Classification Rule Mining (CRM) [15] is a well-known Data Mining technique for the extraction of hidden Classification Rules (CRs) from a given database that is coupled with a set of pre-defined classes, the objective being to build a classifier to classify “unseen” data records. One recent approach to CRM is to employ Association Rule Mining (ARM) [1] techniques to identify the desired CRs, i.e. Classification Association Rule Mining (CARM). In [9], Coenen *et al.* suggest that results presented in [13] and [14] show that CARM seems to offer greater accuracy of classification, in many cases, than other CRM methods such as C4.5 [15]. CARM mines a set of Classification Association Rules (CARs) from a class transaction database (the

well-established transaction database in a class fashion), where a CAR describes an implicative co-occurring relationship between a set of binary-valued data attributes (items in a transaction database) and a pre-defined class, expressed in the form of an “antecedent  $\Rightarrow$  consequent-class” rule. Regardless of which particular methodology is used to generate CARs, a classifier is usually presented as an ordered CAR list, based on an applied rule ordering mechanism. In [7] Coenen and Leng evaluated a number of alternative case satisfaction and rule ordering strategies. They indicate that (1) three common case satisfaction approaches are best first rule, best  $K$  rule, and all rules; and (2) five existing rule ordering mechanisms are Confidence-Support-size\_of\_Antecedent (CSA), size\_of\_Antecedent-Confidence-Support (ACS), Weighted Relative Accuracy (WRA), Laplace Accuracy, and  $\chi^2$  Testing. In this paper, we further divide (2) into two groups: (i) pure “support-confidence” framework like, and (ii) additive score assigning like. We consequently propose a hybrid rule ordering approach by combining one mechanism taken from (i) and another mechanism taken from (ii). The experimental results show good performance regarding the accuracy of classification when using the proposed rule ordering approach with the best first rule case satisfaction.

## 2 Related Work

### 2.1 An Overview of CARM Algorithms

The idea of CARM was first presented in [3]. Subsequently a number of alternative approaches have been described. Broadly CARM algorithms can be categorized into two groups according to the way that the CARs are generated:

- **Two stage algorithms** where a set of CARs are produced first (stage 1), which are then pruned and placed into a classifier (stage 2). Examples of this approach include CBA [14] and CMAR [13]. CBA (Classification Based Associations), developed by Liu *et al.* in 1998, is an Apriori [2] based CARM algorithm, which (1) applies its CBA-GR procedure for CAR generation; and (2) applies its CBA-CB procedure to build a classifier based on the generated CARs. CMAR (Classification based on Multiple Association Rules), introduced by Han and Jan in 2001, is similar to CBA but generates CARs through a FP-tree [11] based approach.
- **Integrated algorithms** where the classifier is produced in a single processing step. Examples of this approach include TFPC<sup>1</sup> [7] [9], and induction systems such as FOIL [16], PRM and CPAR [17]. TFPC (Total From Partial Classification), proposed by Coenen *et al.* in 2004, is a Apriori-TFP [8] based CARM algorithm, which generates CARs through efficiently constructing both P-tree and T-tree set enumeration tree structures. FOIL (First Order Inductive Learner) is an inductive learning algorithm for generating CARs developed by Quinlan and Cameron-Jones in 1993. This algorithm was later developed by Yin and Han to produce the PRM (Predictive Rule Mining) CAR generation

---

<sup>1</sup> TFPC may be obtained from <http://www.csc.liv.ac.uk/~frans/KDD/Software>.

algorithm. PRM was then further developed, by Yin and Han in 2003 to produce CPAR (Classification based on Predictive Association Rules).

## 2.2 Case Satisfaction Approaches

In [7] Coenen and Leng summarized three case satisfaction approaches that have been employed in different CARM algorithms for utilizing the resulting classifier to classify “unseen” data. These three case satisfaction approaches are itemized as follows (given a particular case):

- **Best First Rule:** Select the first best rule that satisfies the given case according to some ordering imposed on the CAR list. The ordering can be defined according to many different ordering mechanisms, including: (1) CSA – combinations of confidence, support and size of antecedent, with confidence being the most significant factor (used in CBA, TFPC and the early stages of processing of CMAR); (2) ACS – an alternative to CSA that considers the size of the rule antecedent as the most significant factor; (3) WRA – which reflects a number of rule “interestingness” measures as proposed in [12]; (3) Laplace Accuracy – as used in PRM and CPAR; (5)  $\chi^2$  Testing –  $\chi^2$  values as used, in part, in CMAR; etc.
- **Best  $K$  Rules:** Select the first best  $K$  rules that satisfy the given case and then select a rule according to some averaging process as used for example, in CPAR. The term “best” in this case is defined according to an imposed ordering of the form described in Best First Rule.
- **All Rules:** Collect all rules in the classifier that satisfy the given case and then evaluate this collection to identify a class. One well-known evaluation method in this category is WCS (Weighted  $\chi^2$ ) testing as used in CMAR.

## 3 Rule Ordering Approaches

As noted in the previous section five existing rule ordering mechanisms are identified to support the best first rule case satisfaction strategy. Each can be further separated into two stages: (1) a rule weighting stage where each CAR is labeled with a weighting score that represents the significance of this CAR indicates a predefined class; and (2) a rule re-ordering stage, which sorts the original CAR list in a descending manner, based on the score assigned in stage (1), of each CAR. Based on (1) we divide these existing rule ordering mechanisms into two groups: (i) pure “support-confidence” framework like, and (ii) additive score assigning like. With regards to both stages of rule weighting and rule re-ordering, each rule ordering mechanism can be described in more detail as follows:

### (i) Pure “support-confidence” framework like

- **CSA:** The CSA rule ordering mechanism is based on the well-established “support-confidence” framework. It does not assign an additive weighting score to any CAR in its rule weighting stage, but simply gathers the value of confidence and support, and the size of the rule antecedent to “express” a

weighting score for each CAR. In the rule re-ordering stage, CSA generally sort the original CAR list in a descending order based on the value of confidence of each CAR. For these CARs that share a common value of confidence, CSA sorts them in a descending order based on their support value. Furthermore for these CARs that share common values for both confidence and support, CSA sorts them in an ascending order based on the size of the rule antecedent.

- **ACS:** The ACS rule ordering mechanism is a variation of CSA. It takes the size of the rule antecedent as its major factor (using a descending order) followed by the rule confidence and support values respectively. This rule ordering mechanism ensures that “specific rules have a higher precedence than more general rules” [7].

(ii) **Additive score assigning like**

- **WRA:** The use of WRA can be found in [12], where this technique is used to determine an expected accuracy for each CAR. In its rule weighing stage, WRA assigns an additive weighting score to each CAR. The calculation of the value of a CAR  $r$ , confirmed in [7], is:  $wra(r) = \text{support}(r.\text{antecedent}) * (\text{confidence}(r) - \text{support}(r.\text{consequent}))$ . In the rule re-ordering stage the original CAR list is simply sorted in a descending order based on the assigned  $wra$  value of each CAR.
- **Laplace Accuracy:** The use of the *Laplace expected error estimate* [5] can be found in [17]. The principle of applying this rule ordering mechanism is similar to WRA. The calculation of the *Laplace* value of a CAR  $r$  is:  $Laplace(r) = (\text{support}(r.\text{antecedent} \cup r.\text{consequent}) + 1) / (\text{support}(r.\text{antecedent}) + c)$ , where  $c$  represents the number of pre-defined classes.
- **$\chi^2$  Testing:**  $\chi^2$  Testing is a well known technique in statistics, which can be used to determine whether two variables are independent of one another. In  $\chi^2$  Testing a set of observed values ( $O$ ) is compared against a set of expected values ( $E$ ) – values that would be estimated if there were no associative relationship between the variables. The value of  $\chi^2$  is calculated as:  $\sum_{[i=1..n]} (O_i - E_i)^2 / E_i$ , where  $n$  is the number of observed/expected values, which is always 4 in CARM. If the  $\chi^2$  value between two variables (the antecedent and consequent-class of a CAR) above a given threshold value (for CMAR the chosen threshold is 3.8415), thus it can be concluded that there is a relation between the rule antecedent and consequent-class, otherwise there is not a relation. After assigning an additive  $\chi^2$  value to each CAR, it can be used to re-order the CAR list in a descending basis.

## 4 The Hybrid Rule Ordering Strategy

In [17] Yin and Han believe that there are only a limited number, say at most  $K$  in each class, of CARs that are required to distinguish between classes and should be thus used to make up a classifier. They suggest a value of 5 as an appropriate value for  $K$ , and employ the Laplace accuracy (a method in the additive score assigning like) to estimate the accuracy of CARs. With respect to the above suggestions, we

propose a hybrid rule ordering strategy by combining one rule ordering mechanism taken from (i) the pure “support-confidence” framework like, and another rule ordering mechanism taken from (ii) the additive score assigning like. We sketch the process of the proposed rule ordering approach in Figure 1 as follows.

```

Procedure HYBRID RULE ORDERING;
Input:  a list of CARs  $\mathcal{R}$  (in a CSA or ACS
        ordering manner);
Output: a re-ordered list of CARs  $\mathcal{R}^+$ ;

(1) begin
(2)   $\mathcal{R}^+ := \{\emptyset\}$ ;
(3)   $\mathcal{R}^0 := \{\emptyset\}$ ;
(4)   $\mathcal{R}^0 \leftarrow$  catch the best  $K$  rules in  $\mathcal{R}$  using
        a method in (ii);
(5)   $\mathcal{R}^0 \leftarrow$  re-order  $\mathcal{R}^0$  based on a method in (i);
(6)   $\mathcal{R}^+ \leftarrow \mathcal{R}^0 + \mathcal{R}$ ;
(7)  return ( $\mathcal{R}^+$ );
(8) end

```

Fig. 1. The HYBRID RULE ORDERING Procedure

Six different schemes can be identified in this hybrid approach:

- **Hybrid CSA/WRA:** Selects the best  $K$  rules in a WRA manner, and re-orders both the best  $K$  CAR list and the original CAR list in a CSA fashion. (Note: we assume that both CAR lists use the same ordering fashion as either CSA or ACS);
- **Hybrid CSA/Laplace:** Selects the best  $K$  rules in a Laplace manner, and re-orders both the best  $K$  CAR list and the original CAR list in a CSA fashion;
- **Hybrid CSA/ $\chi^2$ :** Selects the best  $K$  rules in a  $\chi^2$  manner, and re-orders both the best  $K$  CAR list and the original CAR list in a CSA fashion;
- **Hybrid ACS/WRA:** Selects the best  $K$  rules in a WRA manner, and re-orders both the best  $K$  CAR list and the original CAR list in an ACS fashion.
- **Hybrid ACS/Laplace:** Selects the best  $K$  rules in a Laplace manner, and re-orders both the best  $K$  CAR list and the original CAR list in an ACS fashion; and
- **Hybrid ACS/ $\chi^2$ :** Selects the best  $K$  rules in a  $\chi^2$  manner, and re-orders both the best  $K$  CAR list and the original CAR list in an ACS fashion.

## 5 Experimental Results

In this section, we aim to evaluate the proposed hybrid rule ordering approach with respect to the accuracy of classification. All evaluations were obtained using the TFPC algorithm coupled with the best first rule case satisfaction strategy, although

any other CARM classifier generator, founded on the best first rule strategy, could equally well be used. Experiments were run on a 1.20 GHz Intel Celeron CPU with 256 Mbyte of RAM running under Windows Command Processor.

The experiments were conducted using a range of datasets taken from the LUCS-KDD discretised/normalized ARM and CARM Data Library [6]. The chosen datasets are originally taken from the UCI Machine Learning Repository [4]. These datasets have been discretised and normalized using the LUCS-KDD DN software<sup>2</sup>, so that data are then presented in a binary format suitable for use with CARM applications. It should be noted that the datasets were rearranged so that occurrences of classes were distributed evenly throughout the datasets. This then allowed the datasets to be divided in half with the first half used as the training set and the second half as the test set. Although a “better” accuracy figure might have been obtained using Ten-Cross Validation [10], it is the relative accuracy that is of interest here and not the absolute accuracy.

The first set of evaluations undertaken used a confidence threshold value of 50% and a support threshold value 1% (as used in the published evaluations of CMAR [11], CPAR [17], TFPC [7] [9]). The results are presented in Table 1 where 120 classification accuracy values are listed based on 24 chosen datasets. The row labels describe the key characteristics of each dataset: for example, the label

**Table 1.** Classification accuracy – five existing rule ordering approaches

DATASETS	CSA	ACS	WRA	Laplace	$\chi^2$
<i>adult.D97.N48842.C2</i>	80.83	73.99	81.66	76.07	76.07
<i>anneal.D73.N898.C6</i>	91.09	75.50	87.75	77.51	77.51
<i>auto.D137.N205.C7</i>	61.76	53.92	50.00	47.06	50.00
<i>breast.D20.N699.C2</i>	89.11	89.11	87.68	65.62	65.62
<i>connect4.D129.N67557.C3</i>	65.83	64.83	67.93	65.83	65.83
<i>cylBands.D124.N540.C2</i>	65.93	42.59	64.07	57.78	57.78
<i>flare.D39.N1389.C9</i>	84.44	83.86	84.15	84.44	84.44
<i>glass.D48.N214.C7</i>	58.88	43.93	50.47	52.34	50.47
<i>heart.D52.N303.C5</i>	58.28	28.48	55.63	54.97	54.97
<i>hepatitis.D56.N155.C2</i>	68.83	48.05	71.43	79.22	79.22
<i>horseColic.D85.N368.C2</i>	72.83	40.76	79.89	79.89	63.04
<i>ionosphere.D157.N351.C2</i>	85.14	61.14	86.86	64.57	64.57
<i>iris.D19.N150.C3</i>	97.33	97.33	97.33	97.33	97.33
<i>led7.D24.N3200.C10</i>	68.38	61.38	63.94	63.88	65.56
<i>letRecog.D106.N20000.C26</i>	31.13	26.21	26.33	26.33	28.52
<i>mushroom.D90.N8124.C2</i>	99.21	65.76	98.45	98.45	49.43
<i>nursery.D32.N12960.C5</i>	80.35	55.88	70.17	70.17	70.17
<i>pageBlocks.D46.N5473.C5</i>	90.97	90.97	90.20	89.80	89.80
<i>pima.D38.N768.C2</i>	73.18	71.88	72.92	65.10	65.10
<i>soybean-large.D118.N683.C19</i>	86.22	79.77	36.36	36.07	77.42
<i>ticTacToe.D29.N958.C2</i>	71.61	36.12	68.06	65.34	65.34
<i>waveform.D101.N5000.C3</i>	61.56	47.96	56.24	57.84	57.28
<i>wine.D68.N178.C3</i>	56.18	37.08	80.90	73.03	70.79
<i>zoo.D42.N101.C7</i>	80.00	42.00	56.00	42.00	42.00
<i>Average</i>	<b>74.13</b>	<b>59.10</b>	<b>70.18</b>	<b>66.28</b>	<b>65.34</b>

*adult.D97.N48842.C2* denotes the “adult” dataset, which includes 48,842 records in 2 pre-defined classes, with attributes that for the experiments described here have been discretised and normalized into 97 binary categories.

From Table 1 it can be seen that with a 50% confidence threshold and a 1% support threshold the CSA rule ordering mechanism worked better than other alternative approaches. When applying the CSA rule ordering mechanism, the average accuracy of classification throughout the 24 datasets is 74.13%, whereas using ACS is 59.10%, WRA is 70.18%, Laplace is 66.28%, and  $\chi^2$  is 65.34%.

The second set of evaluations undertaken used a confidence threshold value of 50%, a support threshold value of 1%, and a value of 5 as an appropriate value for *K* when selecting the best *K* rules (as suggested by Yin and Han in [17]). The results are presented in Table 2 where 144 classification accuracy values are listed based on 24 chosen datasets.

**Table 2.** Classification accuracy – six hybrid rule ordering schemes

DATASETS	CSA/ WRA	CSA/ Laplace	CSA/ $\chi^2$	ACS/ WRA	ACS/ Laplace	ACS/ $\chi^2$
<i>adult.D97.N48842.C2</i>	83.33	79.95	79.95	78.56	83.76	80.14
<i>anneal.D73.N898.C6</i>	91.09	91.54	91.54	80.40	80.62	88.20
<i>auto.D137.N205.C7</i>	59.80	58.82	53.92	55.88	54.90	52.94
<i>breast.D20.N699.C2</i>	89.11	88.54	89.11	89.11	88.54	89.11
<i>connect4.D129.N67557.C3</i>	67.67	65.83	65.83	64.88	64.88	64.88
<i>cylBands.D124.N540.C2</i>	67.04	69.26	57.78	61.11	70.00	53.33
<i>flare.D39.N1389.C9</i>	84.29	84.44	84.44	83.86	83.86	83.86
<i>glass.D48.N214.C7</i>	66.36	66.36	66.36	65.42	65.42	68.22
<i>heart.D52.N303.C5</i>	55.63	56.95	58.94	52.32	50.33	50.33
<i>hepatitis.D56.N155.C2</i>	84.42	84.42	84.42	63.64	71.43	68.83
<i>horseColic.D85.N368.C2</i>	83.15	83.15	79.89	75.00	83.15	71.20
<i>ionosphere.D157.N351.C2</i>	90.29	89.71	88.00	90.29	89.71	88.00
<i>iris.D19.N150.C3</i>	97.33	97.33	97.33	97.33	97.33	97.33
<i>led7.D24.N3200.C10</i>	68.19	68.19	68.38	62.06	62.06	62.31
<i>letRecog.D106.N20000.C26</i>	31.49	31.49	31.56	27.39	27.39	28.41
<i>mushroom.D90.N8124.C2</i>	98.45	98.82	98.45	98.45	98.82	98.45
<i>nursery.D32.N12960.C5</i>	78.86	78.86	78.86	66.73	66.73	66.73
<i>pageBlocks.D46.N5473.C5</i>	90.97	90.97	90.97	90.97	90.97	90.97
<i>pima.D38.N768.C2</i>	73.18	73.18	72.66	73.18	73.18	72.66
<i>soybean-large.D118.N683.C19</i>	80.94	80.94	82.11	75.66	75.66	78.01
<i>ticTacToe.D29.N958.C2</i>	74.95	74.74	72.65	60.75	70.35	67.22
<i>waveform.D101.N5000.C3</i>	57.96	57.96	60.60	59.20	59.20	60.60
<i>wine.D68.N178.C3</i>	77.53	77.53	77.53	77.53	77.53	77.53
<i>zoo.D42.N101.C7</i>	84.00	90.00	72.00	80.00	80.00	80.00
<b>Average</b>	<b>76.50</b>	<b>76.62</b>	<b>75.14</b>	<b>72.07</b>	<b>73.58</b>	<b>72.47</b>

From Table 2 it can be seen that with a 50% confidence threshold, a 1% support threshold, and 5 as the value of *K*, the approach hybrid CSA/Laplace preformed better

than other alternative hybrid schemes. When applying the hybrid CSA/Laplace, the average accuracy of classification throughout the 24 datasets is 76.62%. Let CSA and Laplace be the “parents” of the hybrid CSA/Laplace, we indicate that the classification accuracy obtained using the hybrid CSA/Laplace is significantly higher than using its “parents”, where CSA is 74.13% and Laplace is 66.28%. Furthermore we identify:

- The classification accuracy of the hybrid CSA/WRA is significantly higher than the accuracies of its “parents”, where the average accuracy of the hybrid CSA/WRA is 76.50% whereas CSA is 74.13% and WRA is 70.18%.
- The classification accuracy of the hybrid CSA/ $\chi^2$  is significantly higher than the accuracies of its “parents”, where the accuracy of the hybrid CSA/ $\chi^2$  is 75.14% whereas CSA is 74.13% and  $\chi^2$  is 65.34%;
- The accuracy of the hybrid ACS/WRA is significant higher than the accuracies of its “parents”, where the hybrid ACS/WRA is 72.07% whereas ACS is 59.10% and WRA is 70.18%;
- The accuracy of the hybrid ACS/Laplace is significantly higher than its “parents”, where the hybrid ACS/Laplace is 73.58% whereas ACS is 59.10% and Laplace is 66.28%; and
- The accuracy of the hybrid ACS/ $\chi^2$  is significantly higher than its “parents”, where the hybrid ACS/ $\chi^2$  is 72.47% whereas ACS is 59.10% and  $\chi^2$  is 65.34%.

## 6 Conclusion

This paper is concerned with an investigation of CARM. An overview of alternative CARM algorithms was provided in Section 2.1, and three current case satisfaction strategies were reviewed in Section 2.2. In Section 3 with regards to both stages of rule weighting and rule re-ordering, we described the existing rule ordering mechanisms in groups (the “support-confidence” framework like vs. the additive score assigning like). A hybrid rule ordering approach was proposed in Section 4, which combines an approach taken from the “support-confidence” framework like, and another approach taken from the additive score assigning like. Subsequently six hybrid rule ordering schemes were introduced. From the experimental results (see Section 5), all six hybrid schemes presented good classification accuracy – the accuracy is significantly higher than the accuracies obtained by their “parent” rule ordering approaches. Further research is suggested to identify the improved rule ordering approach to give a better performance.

## Acknowledgments

The authors would like to thank Prof. Paul Leng and Dr. Robert Sanderson of the Department of Computer Science at the University of Liverpool for their support with respect to the work described here.

## References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Buneman, P., Jajodia, S. (eds.) SIGMOD-93. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, May 1993, pp. 207–216. ACM Press, New York (1993)
2. Agrawal, R., Srikant, R.: Fast Algorithm for Mining Association Rules. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) VLDB-94. Proceedings of the 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, September 1994, pp. 487–499. Morgan Kaufmann Publishers, San Francisco (1994)
3. Ali, K., Manganaris, S., Srikant, R.: Partial Classification using Association Rules. In: Heckerman, D., Mannila, H., Pregibon, D., Uthurusamy, R. (eds.) KDD-97. Proceedings of the Third International conference on Knowledge Discovery and Data Mining, Newport Beach, California, August 1997, pp. 115–118. AAAI Press, Menlo Park (1997)
4. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. Department of Information and Computer Science, University of California, Irvine, CA, United States (1998), <http://www.ics.uci.edu/mllearn/MLRepository.html>
5. Clark, P., Boswell, R.: Rule Induction with CN2: Some Recent Improvement. In: Kodratoff, Y. (ed.) Machine Learning - EWSL-91. LNCS, vol. 482, pp. 111–116. Springer, Heidelberg (1991)
6. Coenen, F.: The LUCS-KDD Discretised/Normalised ARM and CARM Data Library. Department of Computer Science, The University of Liverpool, UK (2003), <http://www.csc.liv.ac.uk/frans/KDD/Software/LUCS-KDD-DN>
7. Coenen, F., Leng, P.: An Evaluation of Approaches to Classification Rule Selection. In: ICDM-04. Proceedings of the 4th IEEE International Conference on Data Mining, Brighton, November 2004, pp. 359–362. IEEE Computer Society Press, Los Alamitos (2004)
8. Coenen, F., Leng, P., Ahmed, S.: Data Structure for Association Rule Mining: T-trees and P-trees. IEEE Transactions on Knowledge and Data Engineering 16(6), 774–778 (2004)
9. Coenen, F., Leng, P., Zhang, L.: Threshold Tuning for Improved Classification Association Rule Mining. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 216–225. Springer, Heidelberg (2005)
10. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, Heidelberg (2002)
11. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: Chen, W., Naughton, J.F., Bernstein, P.A. (eds.) SIGMOD-00. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, May 2000, pp. 1–12. ACM Press, New York (2000)
12. Lavrac, N., Flach, P., Zupan, B.: Rule Evaluation Measures: A Unifying View. In: Džeroski, S., Flach, P.A. (eds.) Inductive Logic Programming. LNCS (LNAI), vol. 1634, pp. 174–185. Springer, Heidelberg (1999)
13. Li, W., Han, J., Pei, J., CMAR.: CMAR: Accurate and Efficient Classification based on Multiple Class-association Rules. In: Cercone, N., Lin, T.Y., Wu, X. (eds.) ICDM-01. Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, November 29 –December 2, 2001, pp. 369–376. IEEE Computer Society Press, Los Alamitos (2001)
14. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: Agrawal, R., Stolorz, P.E., Piatetsky-Shapiro, G. (eds.) KDD-98. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York City, August 1998, pp. 80–86. AAAI Press, Menlo Park (1998)

15. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)
16. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A Midterm Report. In: Brazdil, P.B. (ed.) Machine Learning: ECML-93. LNCS, vol. 667, pp. 3–20. Springer, Heidelberg (1993)
17. Yin, X., Han, J.: CPAR: Classification based on Predictive Association Rules. In: Barbara, D., Kamath, C. (eds.) SDM-03. Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, May 2003, SIAM, Philadelphia pp. 331–335 (2003)