Transactions on Machine Learning
and Data Mining
Vol.1, No 1 (2008) 1-15
© ISSN: 1865-6781 (Journal),
IBaI Publishing ISSN 1864-9734

ibai Publishing
www.ibai-publishing.org

Hybrid Rule Ordering in Classification Association Rule Mining

Yanbo J. Wang [1,*], Qin Xin [2], and Frans Coenen [1]

[1] Department of Computer Science, The University of Liverpool,
Ashton Building, Ashton Street, Liverpool, L69 3BX, United Kingdom
{jwang, frans}@csc.liv.ac.uk; Tel.: +44 151 7954253; Fax: +44 151 7954235

[2] Simula Research Laboratory, P.O. Box 134,
NO-1325 Lysaker, Norway
xin@simula.no

Abstract. Classification Association Rule Mining (CARM) is an approach to classifier generation that builds an Association Rule Mining based classifier using Classification Association Rules (CARs). Regardless of which particular CARM algorithm is used, a similar set of CARs is always generated from data, and a classifier is usually presented as an ordered list of CARs, based on a selected rule ordering strategy. Hence to produce an accurate classifier, it is essential to develop a rational rule ordering mechanism. In the past decade, a number of rule ordering strategies have been introduced. Six major ones can be identified: Confidence Support & size-of-Antecedent (CSA), size-of-Antecedent Confidence & Support (ACS), Confidence Support size-of-Antecedent class-distribution-Frequency & Row-ordering (CSAFR), Weighted Relative Accuracy (WRA), Laplace Accuracy (LA), and Chi-square Testing (χ^2). Broadly speaking, these strategies can be categorized into two groups: Support-Confidence (including CSA, ACS and CSAFR) and Rule Weighting (including WRA, LA and χ^2). In this paper, we propose a hybrid rule ordering approach (framework) by combining one strategy taken from Support-Confidence and another strategy taken from Rule Weighting, which consequently develops nine rule ordering mechanisms. The experimental results demonstrate that all developed mechanisms perform well with respect to the accuracy of classification.

Keywords: Classification Association Rules, Classification Association Rule Mining, Confidence, Data Mining, Rule Ordering, Rule Weighting, Support.

* Corresponding author.

1 Introduction

Data mining is an area of current research and development in computer science, which is attracting more and more attention from a wide range of different groups of people. It aims to extract various types (models) of hidden, interesting, previously unknown and potentially useful knowledge (i.e. rules, patterns, regularities, customs, trends, etc.) from databases, where the volume of a collected database can be measured in gigabytes. Association Rule Mining (ARM), first introduced by Agrawal et al. [1], is a well-known data mining research field. It aims to extract a set of Association Rules (ARs) — a common model of mined knowledge — from a given transactional database D_T. Let $I = \{a_1, a_2, \ldots, a_{n-1}, a_n\}$ be a set of (binary-valued) database attributes (items), and $\mathcal{T} = \{T_1, T_2, \ldots, T_{m-1}, T_m\}$ be a set of database records (transactions), D_T is described by $\mathcal{T}$, where each $T_j \in \mathcal{T}$ comprises a set of items $I' \subseteq I$. An AR describes an implicative co-occurring relationship between two sets of items in D_T, expressed in the form of an "antecedent (X) $\Rightarrow$ consequent (Y)" rule, where X, $Y \subseteq I$ and $X \cap Y = \varnothing$. In ARM, two threshold values are usually used to determine the significance of an AR:

1. **Support:** A set of items S is called an itemset. The support of S is the proportion of transactions T in $\mathcal{T}$ for which $S \subseteq T$. If the support of S exceeds a user-supplied support threshold σ, S is defined to be a frequent/significant itemset.
2. **Confidence:** Represents how "strongly" an itemset (rule antecedent) X implies another itemset (rule consequent) Y. A confidence threshold α, supplied by the user, is used to distinguish high confidence ARs from low confidence ARs.

An AR "$X \Rightarrow Y$" is said to be *valid* when the support for the co-occurrence of X and Y exceeds σ, and the confidence of this AR exceeds α. The computation of support is:

$$support(X \cup Y) = count(X \cup Y) / |\mathcal{T}|, \quad (1)$$

where $count(X \cup Y)$ is the number of transactions containing the set $X \cup Y$ in $\mathcal{T}$, and $|\mathcal{T}|$ is the size function (cardinality) of the set $\mathcal{T}$. The computation of confidence is:

$$confidence(X \Rightarrow Y) = support(X \cup Y) / support(X). \quad (2)$$

The most well-known ARM algorithm is the Apriori algorithm, developed by Agrawal and Srikant [2], which has been the basis of many subsequent ARM and/or ARM-related algorithms.

Classification Rule Mining (CRM) [3] is a data mining technique for identifying hidden Classification Rules (CRs) — another common knowledge model — in a given class database D_C, the objective being to build a classifier to categorize "unseen" data instances/records. Generally D_C is described by a relational database table that includes a class attribute — whose values are a set of pre-defined class labels $C = \{c_1, c_2, \ldots, c_{|C|-1}, c_{|C|}\}$. The process of CRM consists of two stages: (i) a training phase where CRs are generated from a set of training data instances $D_R \subset D_C$; and (ii) a test phase where "unseen" instances in a test dataset $D_E \subset D_C$ are assigned into pre-defined class groups. A D_C is established as $D_R \cup D_E$, where $D_R \cap D_E = \varnothing$. Both D_R and D_E share the same database attributes except the class attribute. By

convention the last attribute in each D_R record usually indicates the pre-defined class of this record, noted as the class attribute, while the class attribute is missing in D_E. One approach to CRM is to employ ARM methods to identify the desired CRs, i.e. Classification Association Rule Mining (CARM) [4].

CARM mines a set of Classification Association Rules (CARs) from a class-transactional database $D_{C\text{-}T}$ (i.e. the well-established transactional database in a class fashion). A CAR is a special AR that describes an implicative co-occurring relationship between a set of items and a pre-defined class, expressed in the form of an "antecedent (X) $\Rightarrow$ consequent-class (c_i)" rule. Coenen et al. [5] and Shidara et al. [6] suggest that results presented in the studies of [7–9] show that in many cases CARM offers greater classification accuracy than other traditional CRM methods, such as C4.5 [3] and RIPPER (Repeated Incremental Pruning to Produce Error Reduction) [10]. Coenen and Leng [11] further indicate:

1. *"Training of the classifier is generally much faster using CARM techniques than other classification generation techniques such as decision tree and SVM (Support Vector Machine) approaches"* (particularly when handling multi-class problems as opposed to two-class problems);
2. *"Training sets with high dimensionality can be handled very effectively"*; and
3. *"The resulting classifier is expressed as a set of rules which are easily understandable and simple to apply to unseen data (an advantage also shared by some other techniques, e.g. decision tree classifiers)".*

In the past decade, a number of CARM approaches have been developed. Although these CARM methods employ different ARM techniques to extract CARs from a given $D_{C\text{-}T}$, a similar set of CARs is always generated, usually based on user-supplied support and confidence thresholds. Regardless of which particular method is used to generate CARs, a classifier is usually presented as an ordered list of CARs, based on a selected rule ordering strategy. Hence, it can be observed that the way to produce a more accurate CARM classifier is to develop a better rule ordering approach. Six major CARM rule ordering mechanisms can be identified: Confidence Support & size-of-Antecedent (CSA), size-of-Antecedent Confidence & Support (ACS), Confidence Support size-of-Antecedent class-distribution-Frequency & Row-ordering (CSAFR), Weighted Relative Accuracy (WRA), Laplace Accuracy (LA), and Chi-square Testing (χ^2). In this paper, we divide these mechanisms into two categories: Support-Confidence based which includes CSA, ACS and CSAFR; and Rule Weighting based which includes WRA, LA and χ^2. We subsequently propose a hybrid rule ordering approach (framework) by combining one mechanism taken from Support-Confidence and another mechanism taken from Rule Weighting. As a consequence, nine rule ordering mechanisms are produced. With regard to the Best First Rule case satisfaction approach [12], the experimental results demonstrate that all rule ordering mechanisms developed in this study perform well with respect to the accuracy of classification.

The rest of this paper is organized as follows. Section 2 describes some related work relevant to this study, where the six major (existing) rule ordering strategies in CARM are outlined. The proposed hybrid rule ordering approach is described in section 3. In section 4 we present experimental results. Finally our conclusions and open issues for further research are given in section 5.

2 Related Work

2.1 An Overview of CARM Algorithms

The idea of CARM was first presented by Ali et al. [4]. Subsequently a number of alternative approaches have been developed. Broadly speaking, 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). Typical algorithms of this approach include CBA (Classification Based on Associations) [8] and CMAR (Classification based on Multiple Association Rules) [7]. CBA is an Apriori based CARM algorithm, which: (i) applies its CBA-RG (Rule Generator) procedure for CAR generation; and (ii) applies its CBA-CB (Classifier Builder) procedure to build a classifier based on the generated CARs. CMAR is similar to CBA but generates CARs through a FP-tree [13] based approach.
- **Integrated Algorithms** where the classifier is produced in a single processing step. Algorithms of this kind include TFPC (Total From Partial Classification) [5,11,12], and CPAR (Classification based on Predictive Association Rules)[9]. TFPC is an Apriori-TFP [14,15] based CARM algorithm, which generates CARs through efficiently constructing both P-tree and T-tree set enumeration tree [16] structures. CPAR is based on the PRM (Predictive Rule Mining) algorithm, and PRM is modified from the FOIL (First Order Inductive Learner) algorithm [17].

2.2 Case Satisfaction Mechanisms

Coenen and Leng [12] summarize three case satisfaction mechanisms that have been employed in a variety of CARM algorithms for utilizing the resulting classifier to classify "unseen" data records. These three case satisfaction approaches are itemized as follows (given a particular case):

- **Best First Rule:** Select the first rule that satisfies the given case according to some ordering imposed on the list of generated CARs. The ordering can be defined according to many different ordering strategies including:
  1. CSA (Confidence Support & size-of-Antecedent) where confidence is the most significant factor and size-of-antecedent the least significant factor (used in CBA, TFPC and the early stage of processing of CMAR), where size-of-antecedent is measured by the cardinality of the rule antecedent;
  2. ACS (size-of-Antecedent Confidence & Support), an alternative mechanism to CSA that considers size-of-antecedent the most significant factor and support the least significant factor;
  3. CSAFR (Confidence Support size-of-Antecedent class-distribution-Frequency & Row-ordering), which makes use of two additional factors, based on the CSA ordering;
  4. WRA (Weighted Relative Accuracy), which reflects a number of rule interestingness measures as proposed in [18];
  5. LA (Laplace Accuracy) — as used in CPAR; and

6. χ^2 (Chi-square Testing) — as used, in part, in CMAR; etc.
These approaches are discussed further in section 2.3.

- **Best *K* Rules:** Select the first (top) *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 χ^2) testing as used in CMAR.

2.3 Rule Ordering Approaches

As noted above, rule ordering strategies support the Best First Rule case satisfaction mechanism. The rule ordering is conducted using some scoring mechanism. The nature of the scoring mechanisms can be divided into two groups. The first group includes the following:

- **CSA:** The CSA rule ordering strategy is based on the well-established "Support-Confidence" framework of for instance [19] that was originally introduced for AR interestingness measure. CSA sorts all generated CARs in a descending order based on the value of confidence of each CAR. For those CARs that share a common value of confidence, CSA sorts them in a descending order based on their support value. Furthermore for those 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 strategy is a variant of CSA. It takes the size of the rule antecedent as its major factor (using a descending order — unlike the ascending order used in CSA) followed by the rule confidence and support values respectively. Coenen and Leng [12] state that ACS ensures: "*specific rules have a higher precedence than more general rules*".
- **CSAFR:** The CSAFR rule ordering strategy, introduced by Thabtah et al. [20], is an extension of the CSA strategy. It begins with the CSA ordering followed by two additional factors, so that: for those CARs that share common values for confidence, support and the size of rule antecedent, CSAFR sorts them in a descending order based on their class distribution frequency (class-based support value); and for those CARs that share common values for all the four factors above, CSAFR sorts them in an ascending order based on the ID-number of the transaction (database row number, i.e. 1, 2, …, *m*–1, *m*), in which the rule first-time appears in the training dataset.

The second group of rule ordering strategies is Rule Weighting based where an additive weighting score is assigned to each CAR, based on a particular weighting scheme. Examples include:

- **WRA:** The WRA measure [18] is used to determine the expected accuracy of each CAR. The calculation of the WRA score of a CAR *R* (as "$X \Rightarrow c_i$") confirmed in [12], is:

$$wra\_score(R) = support(X) \times (confidence(R) - support(c_i)) . \quad (3)$$

WRA simply sorts all generated CARs in a descending order, based on the assigned WRA score of each CAR.

- **LA:** The use of the *Laplace Expected Error Estimate* [21] can be found in [9]. The principle of applying this rule ordering mechanism is similar to WRA. The calculation of the LA score of a CAR R is:

$$la\_score(R) = (support(X \cup \{c_i\}) + 1) / (support(X) + |C|) , \quad (4)$$

where $\{c_i\}$ denotes the 1-itemset form of c_i, and $|C|$ denotes the number of pre-defined classes.

- **χ^2:** χ^2 testing is a well-known technique used in statistics (see for example [22]). It can be used to determine whether two variables are independent of one another. In χ^2 testing, a set of observed values O is compared against a set of expected values E — values that would be estimated if there was no dependence between the variables. The value of χ^2 is calculated using:

$$\chi^2\_\text{value} = \sum_{\{j = 1...n\}} (O_j - E_j)^2 / E_j , \quad (5)$$

where n is the number of entries in the confusion matrix, which is always 4 in CARM. If the χ^2 value between two variables (the rule antecedent and consequent-class of a CAR) is greater than a given threshold value (for CMAR the chosen threshold is 3.8415), it can be concluded that there is a dependence between the rule antecedent and consequent-class; otherwise there is no dependence. After assigning a χ^2 score/value to each CAR, it can be used as the basis for ordering CARs into descending order.

3 The Hybrid Rule Ordering Approach

In this section, we describe the proposed hybrid rule ordering approach in detail (also see the initial version of this work in [23]). Yin and Han [9] suggest that there are only a limited number (perhaps 5 in each class) of CARs that are required to distinguish between classes and should thus be used to make up a classifier. Yin and Han employ LA to estimate the accuracy of CARs. Incorporating the K rules concept of Yin and Han a hybrid Support-Confidence & Rule Weighting based ordering approach can be developed. The hybrid approach fuses both the case satisfaction mechanisms of Best First Rule and Best K Rules. The overall procedure of the hybrid rule ordering strategy is outlined as follows:

Algorithm: The Hybrid Rule Ordering Procedure
Input: (a) A list of CARs $\mathcal{R}$ (in CSA, ACS or CSAFR ordering manner);
(b) A desired number (integer value) K of the best rules;

```
Output:  A re-ordered list of CARs ℛ^HYBRID (in a hybrid
         rule ordering manner);
Begin Algorithm
(1)  ℛ^HYBRID ← ∅;
(2)  ℛ^δCORE ← ∅;
(3)  for each CAR ∈ ℛ do
(4)    calculate the additive score (δ) for this CAR
       (in WRA, LA or χ² ordering manner);
(5)    add (CAR ⊕ δ) into ℛ^δCORE; //the ⊕ sign means
       "with" an additive CAR attribute
(6)  end for
(7)  sort ℛ^δCORE in a descending order based on δ;
(8)  ℛ^δCORE ← select the top K CARs (for each pre-
     defined class) ∈ ℛ^δCORE;
(9)  sort ℛ^δCORE in CSA, ACS or CSAFR ordering manner;
     //keep ℛ^δCORE ordering consistent with ℛ ordering
(10) ℛ^HYBRID ← link ℛ^δCORE at front of ℛ;
(11) return (ℛ^HYBRID);
End Algorithm
```

From the foregoing, nine individual hybrid rule ordering mechanisms are produced:

- **Hybrid CSA/WRA:** Selects the Best *K* Rules (for each pre-defined class) in a WRA manner from the given (original) CAR list (that is presented in a CSA manner), and re-orders the best *K* CAR list in a CSA manner. The best *K* CAR list is then linked at the front of the original CAR list.
- **Hybrid CSA/LA:** Selects the Best *K* Rules (for each pre-defined class) in an LA manner from the given (original) CAR list (that is presented in a CSA manner), and re-orders the best *K* CAR list in a CSA manner. The best *K* CAR list is then linked at the front of the original CAR list.
- **Hybrid CSA/χ^2:** Selects the Best *K* Rules (for each pre-defined class) in a χ^2 manner from the given (original) CAR list (that is presented in a CSA manner), and re-orders the best *K* CAR list in a CSA manner. The best *K* CAR list is then linked at the front of the original CAR list.
- **Hybrid ACS/WRA:** Selects the Best *K* Rules (for each pre-defined class) in a WRA manner from the given (original) CAR list (that is presented in an ACS manner), and re-orders the best *K* CAR list in an ACS manner. The best *K* CAR list is then linked at the front of the original CAR list.
- **Hybrid ACS/LA:** Selects the Best *K* Rules (for each pre-defined class) in an LA manner from the given (original) CAR list (that is presented in an ACS manner), and re-orders the best *K* CAR list in an ACS manner. The best *K* CAR list is then linked at the front of the original CAR list.
- **Hybrid ACS/χ^2:** Selects the Best *K* Rules (for each pre-defined class) in a χ^2 manner from the given (original) CAR list (that is presented in an ACS manner), and re-orders the best *K* CAR list in an ACS manner. The best *K* CAR list is then linked at the front of the original CAR list.

- **Hybrid CSAFR/WRA:** Selects the Best *K* Rules (for each pre-defined class) in a WRA manner from the given (original) CAR list (that is presented in a CSAFR manner), and re-orders the best *K* CAR list in a CSAFR manner. The best *K* CAR list is then linked at the front of the original CAR list.
- **Hybrid CSAFR/LA:** Selects the Best *K* Rules (for each pre-defined class) in an LA manner from the given (original) CAR list (that is presented in a CSAFR manner), and re-orders the best *K* CAR list in a CSAFR manner. The best *K* CAR list is then linked at the front of the original CAR list.
- **Hybrid CSAFR/χ^2:** Selects the Best *K* Rules (for each pre-defined class) in a χ^2 manner from the given (original) CAR list (that is presented in a CSAFR manner), and re-orders the best *K* CAR list in a CSAFR manner. The best *K* CAR list is then linked at the front of the original CAR list.

4 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 [1] CARM 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 1.86 GHz Intel(R) Core(TM)2 CPU with 1.00 GB of RAM running under Windows Command Processor.

The experiments were conducted using a range of datasets taken from the LUCS-KDD discretised/normalised ARM and CARM Data Library [24]. The chosen datasets are originally taken from the UCI Machine Learning Repository [25]. These datasets have been discretized and normalized using the LUCS-KDD DN software [2], so that data are then presented in a binary format suitable for use with CARM applications. It should be noted that each chosen dataset has been re-arranged so that occurrences of classes are distributed evenly throughout the dataset. This allows TFPC to be applied to (90% — training set, 10% — test set) divisions of each dataset with Ten-fold Cross Validation (TCV) accuracy setting.

The first set of evaluations undertaken used a confidence threshold value of 50% and a support threshold value of 1%, as used in the published evaluations of CMAR [7], CPAR [9], TFPC [5,11,12], etc. The results are presented in Table 1 where 114 classification accuracy values are listed based on 19 chosen datasets. The row labels describe the key characteristics of each dataset: for example, the label *anneal.D73.N898.C6* denotes the "anneal" dataset, which includes 898 records in 6 pre-defined classes, with attributes that for the experiments described here have been discretized and normalized into 73 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

[1] TFPC software may be obtained from http://www.csc.liv.ac.uk/~frans/KDD/Software/Apriori-TFPC/aprioriTFPC.html.

[2] DN software may be obtained from http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS-KDD-DN/lucs-kdd_DN.html.

Table 1. Classification accuracy — six existing rule ordering mechanisms.

DATASETS	CSA	ACS	CSAFR	WRA	LA	χ^2
(1) ***adult.D97.N48842.C2***	80.80	74.70	80.80	81.65	76.07	76.07
(2) ***anneal.D73.N898.C6***	88.29	75.58	88.29	85.95	76.17	76.17
(3) ***breast.D20.N699.C2***	89.99	89.99	89.99	88.23	65.51	65.51
(4) ***connect4.D129.N67557.C3***	65.83	65.18	65.83	67.74	65.83	65.83
(5) ***flare.D39.N1389.C9***	84.30	84.30	84.30	84.30	84.30	84.30
(6) ***glass.D48.N214.C7***	64.97	50.74	64.02	53.62	47.73	52.74
(7) ***heart.D52.N303.C5***	51.42	39.76	51.42	54.09	54.09	54.09
(8) ***hepatitis.D56.N155.C2***	81.83	48.50	81.83	70.67	79.33	79.33
(9) ***horseColic.D85.N368.C2***	79.07	41.11	79.07	81.34	65.53	63.03
(10) ***ionosphere.D157.N351.C2***	86.34	64.67	86.34	81.48	64.10	64.10
(11) ***iris.D19.N150.C3***	95.33	95.33	95.33	95.33	95.33	96.00
(12) ***led7.D24.N3200.C10***	68.72	64.22	68.72	64.50	63.50	66.81
(13) ***mushroom.D90.N8124.C2***	99.04	64.92	99.04	98.52	98.52	49.43
(14) ***nursery.D32.N12960.C5***	77.75	55.08	77.75	70.97	70.97	70.97
(15) ***pageBlocks.D46.N5473.C5***	89.99	89.99	89.99	90.22	89.77	89.77
(16) ***pima.D38.N768.C2***	74.37	73.85	74.37	74.37	65.10	65.10
(17) ***soybean-large.D118.N683.C19***	88.01	86.10	88.15	36.15	34.98	74.10
(18) ***ticTacToe.D29.N958.C2***	67.10	39.03	67.10	70.50	65.34	65.34
(19) ***wine.D68.N178.C3***	71.51	50.28	70.52	77.44	67.01	68.19
Average	**79.19**	**65.96**	**79.10**	**75.11**	**69.96**	**69.84**

rule ordering mechanism, the average accuracy of classification throughout the 19 datasets is 79.19%, whereas using ACS is 65.96%, CSAFR is 79.10%, WRA is 75.11%, LA is 69.96%, and χ^2 is 69.84%.

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 (K = 5 was also used in [9]). The results are presented in Table 2 where 171 classification accuracy values are listed based on 19 chosen datasets. 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 CSAFR/χ^2 preformed better than other alternative hybrid rule ordering mechanisms. When applying Hybrid CSAFR/χ^2, the average accuracy of classification throughout the 19 datasets is 79.48%. Let CSAFR and χ^2 be the "parents" of Hybrid CSAFR/χ^2, we indicate that the classification accuracy obtained using Hybrid CSAFR/χ^2 is greater than the accuracies obtained by its "parents", where CSAFR is 79.10% and χ^2 is 69.84%. Furthermore we identify:

- The classification accuracy of Hybrid CSA/WRA is greater than the accuracies obtained by its "parents", where the average accuracy of Hybrid CSA/WRA is 79.37% whereas CSA is 79.19% and WRA is 75.11%.
- The classification accuracy of Hybrid CSA/LA is greater than the accuracies obtained by its "parents", where the average accuracy of Hybrid CSA/LA is 79.22% whereas CSA is 79.19% and LA is 69.96%.
- The classification accuracy of Hybrid CSA/χ^2 is greater than the accuracies obtained by its "parents", where the average accuracy of Hybrid CSA/χ^2 is 79.46% whereas CSA is 79.19% and χ^2 is 69.84%.

Table 2. Classification accuracy — nine hybrid rule ordering mechanisms.

DATASETS	CSA WRA	CSA LA	CSA χ^2	ACS WRA	ACS LA	ACS χ^2	CSAFR WRA	CSAFR LA	CSAFR χ^2
(1) *adult.D97.N48842.C2*	81.42	80.08	80.08	78.66	83.86	80.19	81.42	80.08	80.08
(2) *anneal.D73.N898.C6*	88.62	89.76	89.76	79.11	80.57	82.25	88.62	89.76	89.76
(3) *breast.D20.N699.C2*	89.99	89.59	91.00	89.99	89.59	91.00	89.99	89.59	91.00
(4) *connect4.D129.N67557.C3*	67.65	65.83	65.83	65.24	65.24	65.24	67.65	65.83	65.83
(5) *flare.D39.N1389.C9*	84.30	84.30	84.30	84.30	84.30	84.30	84.30	84.30	84.30
(6) *glass.D48.N214.C7*	64.97	64.97	65.45	59.73	60.69	64.02	64.97	64.97	65.45
(7) *heart.D52.N303.C5*	54.76	55.09	51.06	50.24	50.58	50.88	54.76	54.76	51.39
(8) *hepatitis.D56.N155.C2*	81.33	81.17	80.50	72.00	76.83	72.67	81.33	81.17	80.50
(9) *horseColic.D85.N368.C2*	80.78	81.01	81.24	80.78	81.01	78.71	80.78	81.01	81.24
(10) *ionosphere.D157.N351.C2*	85.76	84.90	84.05	85.76	84.90	84.05	85.76	84.90	84.05
(11) *iris.D19.N150.C3*	95.33	95.33	95.33	95.33	95.33	95.33	95.33	95.33	95.33
(12) *led7.D24.N3200.C10*	68.72	68.72	68.72	64.63	64.63	64.66	68.72	68.72	68.72
(13) *mushroom.D90.N8124.C2*	98.52	98.82	98.52	98.52	98.82	98.52	98.52	98.82	98.52
(14) *nursery.D32.N12960.C5*	78.42	78.42	78.42	66.74	66.74	66.74	78.42	78.42	78.42
(15) *pageBlocks.D46.N5473.C5*	89.99	90.06	90.66	89.99	90.06	90.66	89.99	90.06	90.66
(16) *pima.D38.N768.C2*	74.37	74.50	74.63	74.37	74.50	74.63	74.37	74.50	74.63
(17) *soybean-large.D118.N683.C19*	83.44	82.12	87.71	78.48	77.15	82.59	83.59	82.26	87.85
(18) *ticTacToe.D29.N958.C2*	67.94	68.15	67.94	61.19	63.16	57.80	67.94	68.15	67.94
(19) *wine.D68.N178.C3*	71.72	72.31	74.47	71.72	72.31	74.47	71.72	72.31	74.47
Average	**79.37**	**79.22**	**79.46**	**76.15**	**76.86**	**76.77**	**79.38**	**79.21**	**79.48**

- The classification accuracy of Hybrid ACS/WRA is greater than the accuracies obtained by its "parents", where the average accuracy of Hybrid ACS/WRA is 76.15% whereas ACS is 65.96% and WRA is 75.11%.
- The classification accuracy of Hybrid ACS/LA is greater than the accuracies obtained by its "parents", where the average accuracy of Hybrid ACS/LA is 76.86% whereas ACS is 65.96% and LA is 69.96%.
- The classification accuracy of Hybrid ACS/χ^2 is greater than the accuracies obtained by its "parents", where the average accuracy of Hybrid ACS/χ^2 is 76.77% whereas ACS is 65.96% and χ^2 is 69.84%.
- The classification accuracy of Hybrid CSAFR/WRA is greater than the accuracies obtained by its "parents", where the average accuracy of Hybrid CSAFR/WRA is 79.38% whereas CSAFR is 79.10% and WRA is 75.11%.
- The classification accuracy of Hybrid CSAFR/LA is greater than the accuracies obtained by its "parents", where the average accuracy of Hybrid CSAFR/LA is 79.21% whereas CSAFR is 79.10% and LA is 69.96%

The third set of evaluations emphasize on the number of instances of best classification accuracies obtained throughout the 19 datasets. The results are presented in Table 3 which combines the results shown in Tables 1 and 2. From Table 3 it can be seen that both proposed mechanisms Hybrid CSAFR/χ^2 and Hybrid CSA/χ^2 gave the highest number of best classification accuracies (8 out of 19 cases). It can be further indicated that the average of the best classification accuracy (instance) numbers for all hybrid rule ordering mechanisms is: (3 + 5 + 8 + 1 + 2 + 4 + 3 + 4 + 8) / 9 = 4.22, which is greater than the average of the best classification accuracy numbers for the six existing mechanisms that is: (5 + 1 + 6 + 5 + 1 + 2) / 6 = 3.33.

Table 3. Number of best classification accuracies — fifteen rule ordering mechanisms.

DATA SETS	CSA	ACS	CSAFR	WRA	LA	χ^2	CSA WRA	CSA LA	CSA χ^2	ACS WRA	ACS LA	ACS χ^2	CSAFR WRA	CSAFR LA	CSAFR χ^2
(1)	80.80	74.70	80.80	81.65	76.07	76.07	81.42	80.08	80.08	78.66	**83.86**	80.19	81.42	80.08	80.08
(2)	88.29	75.58	88.29	85.95	76.17	76.17	88.62	**89.76**	**89.76**	79.11	80.57	82.25	88.62	**89.76**	**89.76**
(3)	89.99	89.99	89.99	88.23	65.51	65.51	89.99	89.59	**91.00**	89.99	89.59	**91.00**	89.99	89.59	**91.00**
(4)	65.83	65.18	65.83	**67.74**	65.83	65.83	67.65	65.83	65.83	65.24	65.24	65.24	67.65	65.83	65.83
(5)	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**	**84.30**
(6)	64.97	50.74	64.02	53.62	47.73	52.74	64.97	64.97	**65.45**	59.73	60.69	64.02	64.97	64.97	**65.45**
(7)	51.42	39.76	51.42	54.09	54.09	54.09	54.76	**55.09**	51.06	50.24	50.58	50.88	54.76	54.76	51.39
(8)	**81.83**	48.50	**81.83**	70.67	79.33	79.33	81.33	81.17	80.50	72.00	76.83	72.67	81.33	81.17	80.50
(9)	79.07	41.11	79.07	**81.34**	65.53	63.03	80.78	81.01	81.24	80.78	81.01	78.71	80.78	81.01	81.24
(10)	**86.34**	64.67	**86.34**	81.48	64.10	64.10	85.76	84.90	84.05	85.76	84.90	84.05	85.76	84.90	84.05
(11)	95.33	95.33	95.33	95.33	95.33	**96.00**	95.33	95.33	95.33	95.33	95.33	95.33	95.33	95.33	95.33
(12)	**68.72**	64.22	**68.72**	64.50	63.50	66.81	**68.72**	**68.72**	**68.72**	64.63	64.63	64.66	**68.72**	**68.72**	**68.72**
(13)	**99.04**	64.92	**99.04**	98.52	98.52	49.43	98.52	98.82	98.52	98.52	98.82	98.52	98.52	98.82	98.52
(14)	77.75	55.08	77.75	70.97	70.97	70.97	**78.42**	**78.42**	**78.42**	66.74	66.74	66.74	**78.42**	**78.42**	**78.42**
(15)	89.99	89.99	89.99	90.22	89.77	89.77	89.99	90.06	**90.66**	89.99	90.06	**90.66**	89.99	90.06	**90.66**
(16)	74.37	73.85	74.37	74.37	65.10	65.10	74.37	74.50	**74.63**	74.37	74.50	**74.63**	74.37	74.50	**74.63**
(17)	88.01	86.10	**88.15**	36.15	34.98	74.10	83.44	82.12	87.71	78.48	77.15	82.59	83.59	82.26	87.85
(18)	67.10	39.03	67.10	**70.50**	65.34	65.34	67.94	68.15	67.94	61.19	63.16	57.80	67.94	68.15	67.94
(19)	71.51	50.28	70.52	**77.44**	67.01	68.19	71.72	72.31	74.47	71.72	72.31	74.47	71.72	72.31	74.47
# of Bests	**5**	**1**	**6**	**5**	**1**	**2**	**3**	**5**	**8**	**1**	**2**	**4**	**3**	**4**	**8**

5 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 2.3 we described the existing rule ordering mechanisms in two groups (Support-Confidence versus Rule Weighting). A hybrid rule ordering approach was proposed subsequently in section 3, which combines an approach taken from Support-Confidence and another approach taken from Rule Weighting. Consequently nine hybrid rule ordering mechanisms were introduced in this paper. From the experimental results, all nine hybrid mechanisms presented good classification accuracy — the accuracy is greater than the accuracies obtained by their "parent" rule ordering approaches. In Table 4, the average classification accuracies (of 19 chosen datasets) for all fifteen rule ordering mechanisms are presented in rank order. The proposed hybrid mechanisms (as highlighted) were ranked as No.1 ~ No.6 and No.9 ~ No.11. Furthermore the average of the best classification accuracy numbers for all hybrid strategies is greater than the average number for the six existing mechanisms. Table 5 shows the number of instances of best classification accuracies for all fifteen rule ordering mechanisms considered here: the best results are coming from both Hybrid CSAFR/χ^2 and Hybrid CSA/χ^2. Further research is suggested to identify the improved rule ordering approach to give a better performance.

Table 4. Ranked order of classification accuracies for the fifteen rule ordering mechanisms

Rank No.	Rule Ordering Mechanism	Average Accuracy
1	Hybrid CSAFR/χ^2	79.48
2	Hybrid CSA/χ^2	79.46
3	Hybrid CSAFR/WRA	79.38
4	Hybrid CSA/WRA	79.37
5	Hybrid CSA/LA	79.22
6	Hybrid CSAFR/LA	79.21
7	CSA	79.19
8	CSAFR	79.10
9	Hybrid ACS/LA	76.86
10	Hybrid ACS/χ^2	76.77
11	Hybrid ACS/WRA	76.15
12	WRA	75.11
13	LA	69.96
14	χ^2	68.84
15	ACS	65.96

Table 5. Ranked order of the best accuracy number for the fifteen rule ordering mechanisms.

Rank No.	Rule Ordering Mechanism	Number of Bests
1	Hybrid CSAFR/χ^2	8
1	Hybrid CSA/χ^2	8
3	CSAFR	6
4	Hybrid CSA/LA	5
4	CSA	5
4	WRA	5
7	Hybrid CSAFR/LA	4
7	Hybrid ACS/χ^2	4
9	Hybrid CSAFR/WRA	3
9	Hybrid CSA/WRA	3
11	Hybrid ACS/LA	2
11	χ^2	2
13	Hybrid ACS/WRA	1
13	LA	1
13	ACS	1

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.): Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD-93), Washington, DC, United States, May 1993. ACM Press, New York, NY (1993) 207–216
2. Agrawal, R., Srikant, R.: Fast Algorithm for Mining Association Rules. In: Bocca, J. B., Jarke, M., Zaniolo, C. (Eds.): Proceedings of the 20th International Conference on Very Large Data Bases (VLDB-94), Santiago de Chile, Chile, September 1994. Morgan Kaufmann Publishers, San Francisco, CA (1994) 487–499
3. Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco, CA, United States (1993)
4. Ali, K., Manganaris, S., Srikant, R.: Partial Classification using Association Rules. In: Heckerman, D., Mannila, H., Pregibon, D., Uthurusamy, R. (Eds.): Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, California, United States, August 1997. AAAI Press, Menlo Park, CA (1997) 115–118
5. Coenen, F., Leng, P., Zhang, L.: Threshold Tuning for Improved Classification Association Rule Mining. In: Ho, T. B., Cheung, D., Liu, H. (Eds.): Proceedings of the Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-05), Hanoi, Vietnam, May 2005. Springer-Verlag Berlin Heidelberg (2005) 216–225
6. Y. Shidara, A. Nakamura, and M. Kudo, CCIC: Consistent Common Itemsets Classifier, In: P. Perner (Ed.), Proceedings of the Fifth International Conference on Machine Learning and Data Mining (MLDM-07), Leipzig, Germany, July 2007. LNAI 4571, Springer-Verlag Berlin Heidelberg, 2007, p. 490–498
7. Li, W., Han, J., Pei, J.: CMAR: Accurate and Efficient Classification based on Multiple Class-association Rules. In: Cercone, N., Lin, T. Y., Wu, X. (Eds.): Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM-01), San Jose, CA, United States, November–December 2001. IEEE Computer Society (2001) 369–376
8. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: Agrawal, R., Stolorz, P. E., Piatetsky-Shapiro, G. (Eds.): Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, United States, August 1998. AAAI Press, Menlo Park, CA (1998) 80–86
9. Yin, X., Han, J.: CPAR: Classification based on Predictive Association Rules. In: Barbara, D., Kamath, C. (Eds.): Proceedings of the Third SIAM International Conference on Data Mining (SDM-03), San Francisco, CA, United States, May 2003. SIAM, Philadelphia, PA (2003) 331–335
10. Cohen, W. W.: Fast Effective Rule Induction. In: Prieditis, A., Russell, S. J. (Eds.): Machine Learning – Proceedings of the Twelfth International Conference on Machine Learning (ICML-95), Tahoe City, CA, United States, July 1995. Morgan Kaufmann Publishers (1995) 115–123
11. Coenen, F., Leng, P.: The Effect of Threshold Values on Association Rule based Classification Accuracy. Journal of Data and Knowledge Engineering, 60, 2 (2007) 345–360
12. Coenen, F., Leng, P.: An Evaluation of Approaches to Classification Rule Selection. In: Proceedings of the 4th IEEE International Conference on Data Mining (ICDM-04), Brighton, UK, November 2004. IEEE Computer Society (2004) 359–362
13. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: Chen, W., Naughton, J. F., Bernstein, P. A. (Eds.): Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD-00), Dallas, TX, United States, May 2000. ACM Press, New York, NY (2000) 1–12

14. 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 (2004) 774–778
15. Coenen, F., Leng, P., Goulbourne, G.: Tree Structures for Mining Association Rules. Journal of Data Mining and Knowledge Discovery, 8, 1 (2004) 25–51
16. Rymon, R.: Search through Systematic Set Enumeration. In: Nebel, B., Rich, C., Swartout, W. R. (Eds.): Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR-92), Cambridge, MA, United States, October 1992. Morgan Kaufmann Publishers (1992) 539–550
17. Quinlan, J. R., Cameron-Jones, R. M.: FOIL: A Midterm Report. In: Brazdil, R. (Ed.): Proceedings of the 1993 European Conference on Machine Learning (ECML-93), Vienna, Austria, April 1993. Springer (1993) 3–20
18. Lavrac, N., Flach, P., Zupan, B.: Rule Evaluation Measures: A Unifying View. In: Dzeroski, S., Flach, P. A. (Eds.): Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99), Bled, Slovenia, June 1999. Springer (1999) 174–185
19. Delgado, M., Martin-Bautista, M. J., Sanchez, D., Vila, M. A.: Mining Text Data: Special Features and Patterns. In: Hand, D. J., Adams, N. M., Bolton, R. J. (Eds.): Pattern Detection and Discovery – Proceedings of ESF Exploratory Workshop, London, United Kingdom, September 2002. Springer-Verlag (2002) 140–153
20. Thabtah, F., Cowling, P., Peng, Y.: The Impact of Rule Ranking on the Quality of Associative Classifiers. In: Bramer, M., Coenen, F., Allen, T. (Eds.): Research and Development in Intelligent Systems XXII – Proceedings of AI-2005, the Twenty-fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-05), Cambridge, United Kingdom, December 2005. Springer-Verlag London Limited (2006) 277–287
21. Clark, P., Boswell, R.: Rule Induction with CN2: Some Recent Improvement. In: Kodratoff, Y. (Ed.): Proceedings of the Fifth European Working Session on Learning (EWSL-91), Porto, Portugal, March 1991. Springer-Verlag Berlin (1991) 111–116
22. Moore, D. S., McCabe, G. P.: Introduction to the Practice of Statistics (Third Edition). W. H. Freeman and Company, United States (1998)
23. Y. J. Wang, Q. Xin, and F. Coenen, A Novel Rule Ordering Approach in Classification Association Rule Mining, In: P. Perner (Ed.), Proceedings of the Fifth International Conference on Machine Learning and Data Mining (MLDM-07), Leipzig, Germany, July 2007. LNAI 4571, Springer-Verlag Berlin Heidelberg, 2007, p. 339–348
24. 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
25. 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/~mlearn/MLRepository.html

Vitae

Yanbo J. Wang is a Ph.D. candidate in the Department of Computer Science at the University of Liverpool, UK. Recently he has started a postdoctoral position in the School of Computer Science & National Centre for Text Mining at the University of Manchester, UK. In 2003, he was awarded a Bachelor of Administrative Studies with Honours, in Information Technology, by York University, Canada. Currently Yanbo serves as an associate editor for the book series of Advances in Knowledge Communities and Social Networks, IGI Global, Hershey, PA, USA.

Qin Xin graduated with his PhD in the Department of Computer Science at the University of Liverpool. His main research focus on algorithmical aspects of communication problems in Wireless Networks, also consider some combinatorial problems for Data Mining. Currently, he is a postdoc at Simula Research Laboratory, Norway.

Frans Coenen has a general background in AI has been working in the field of Data Mining and Knowledge Discovery in Data for some ten years. He is a member of the IFIP WG12.2 — Machine Learning and Data Mining group and the British Computer Society's specialist group in AI. He has some 160 refereed publications on KDD and AI related research. Frans Coenen is currently a senior lecturer within the Department of Computer Science at the University of Liverpool.