An Evaluation of Approaches to Classification Rule Selection

Frans Coenen and Paul Leng
Department of Computer Science, The University of Liverpool, Liverpool, L69 3BX
{frans,phl}@csc.liv.ac.uk

Abstract

In this paper a number of Classification Rule evaluation measures are considered. In particular the authors review the use of a variety of selection techniques used to order classification rules contained in a classifier, and a number of mechanisms used to classify unseen data. The authors demonstrate that rule ordering founded on the size of antecedent works well given certain conditions. **Keywords***: Classification rule evaluation measures.*

1. Introduction

Classification Association Rule Mining (CARM) is based on the observation that a subset of the Association Rules (ARs) generated by Association Rule Mining (ARM) algorithms can effectively be used for the purpose of classification [2]. ARs [1] are rules of the form $A \Rightarrow C$ where A and C (Antecedent and Consequent) are disjoint subsets of a set of binary valued attributes defined by the input data set. In the case of Classification Rules (CRs) the variable C is a unary subset of a set of classes defined with respect to the input data. The notation $r.A$ and $r.C$ will be used to indicate respectively the antecedent and consequent (class) of a rule r.

Once a classifier has been established (usually presented in the form of a list of rules), regardless of the methodology used to generate it, there are a number of proposed mechanisms for using the resulting classifier to classify unseen data. These can be itemised as follows (given a particular case):

1. **Best Rule:** Select the first "best" rule that satisfies the given case according to some ordering imposed on the rule listing. The ordering can be defined according to many different ordering schemes, including:
   (a) **CSA:** Combinations of confidence, support and size of antecedent, with confidence being the most significant factor. Example CARM systems that use CSA ordering include CBA [7], and (only during the early stages of processing) CMAR [6]).
   (b) **WRA:** The weighted relative accuracy which reflects a number of rule "interestingness" measures as proposed in [5].
   (c) **Laplace Accuracy** Laplace accuracy measures as used in PRM and CPAR [8].
   (d) χ^2 **Testing:** χ^2 values as used, in part, in CMAR [6].

   An alternative to CSA, that has not been considered in the literature to date is to make use of the size of the antecedent as the most significant factor followed by confidence and support, i.e. **ACS** ordering.
2. **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 [8]. "Best" in this case is defined according to an imposed ordering of the form described in 1.
3. **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 *Weighted* χ^2 (WCS) testing as used in CMAR [6].

In this paper we compare the above satisfaction/ordering techniques which are described in some further detail in the following two Sections. The evaluation of the techniques is carried out using a number of sets of CRs, generated using the Apriori-TFPC CARM algorithm which is briefly reviewed in Section 4. The evaluation is discussed in detail in Section 5 and some conclusions offered in Section 6.

2. Rule Ordering

As noted in the introduction to this paper five different rule ordering strategies are considered here, four established strategies and one new strategy. Each is described in more detail in the following five sub-sections.

2.1. CSA ordering

The confidence-Support framework remains very common amongst current ARM algorithms. Given a CR, r, the support for r ($sup(r)$) is the proportion of occurrences of the set $r.A \cup r.C$ in the input data compared to the number of records in the input. The confidence of r ($conf(r)$) is then given as $sup(r)/sup(r.A)$. CSA (Confidence, Support, size of Antecedent) ordering is defined as follows:

1. **Confidence**: A rule r_1 has priority over a rule r_2 if $conf(r_1) > conf(r_2)$.
2. **Support**: A rule r_1 has priority over a rule r_2 if $conf(r_1) = conf(r_2)$ and $sup(r_1) > sup(r_2)$.
3. **Size of antecedent**: A rule r_1 has priority over a rule r_2 if $conf(r_1) = conf(r_2)$, $sup(r_1) = sup(r_2)$ and $|r_1.A| < |r_2.A|$.

Given that confidence values are normally calculated as real numbers it is unusual to have CRs with identical confidence values other than in the case of "100% confidence rules". Where 100% rules are found the associated supported value "comes into play". The size of the antecedent is seldom used in CSA ordering.

It should be noted that CARM algorithms that use the support/confidence framework, usually also make use of user defined minimum support/confidence threshold values during the CR generation process. Consequently rules contained in the final classifier will be such that $sup(r) > min\ support\ threshold$ and $conf(r) > min\ conf\ threshold$.

2.2. Weighted Relative Accuracy (WRA)

The use of WRA (Weighted Relative Accuracy) was proposed in [5] as a unifying mechanism for determining CR expected accuracy. The idea is that the WRA measure is a synthesis of a number of rule "interestingness" measures. The WRA for a rule r is calculated using the identity $wra(r) = weighting \times relative\ accuracy$ Where relative accuracy is equal to $confidence(r) - sup(r.C)$.

The term "relative" is used in the sense that the support for a rule is compared with its expected support — a concept not dissimilar to the idea under-pinning χ^2 testing. A negative relative accuracy indicates that the accuracy associated with r is less than the fixed rule $true \Rightarrow C$. So that rules with a low "generality" (i.e. a low $sup(r.A)$ value) are not given a high accuracy measure the support value for the rule antecedent is used to weight the relative accuracy: $wra(r) = sup(r.A) \times (confidence(r) - sup(r.C))$.

2.3. Laplace Accuracy

The *Laplace expected accuracy estimate*, given a rule r, is defined in [8] as follows:

$$Laplace\ accuracy(r) = \frac{sup(r.A \cup r.C) + 1}{sup(r.A) + k)}$$

where k is the number of classes. Note that in this case support is defined as the actual number of records (in the training or test set) that contain $r.A \cup r.C$ or $r.A$.

2.4. χ^2 Testing

χ^2 testing is a well known statistical technique used to determine whether two variables are independent of one another by comparing a set of *observed values* (O) against a set of *expected values* (E) — values that would be expected if there were no association between the variables. A χ^2 value is calculate using the identity:

$$\chi^2 = \sum_{1 \leq i \leq n} (0_i - E_i)^2/E_i$$

where n is the number of observed/expected values (this is always 4 in the case of CARM). If the result is above a given *critical threshold value* then it can be said that a relationship between the variables exists, otherwise there is no relationship. For CMAR a critical threshold value of 3.8415 was used (this value has also been used in this paper).

2.5. ACS or Specificity ordering

In this paper it is proposed that a good alternative ordering to CSA (as described above) is ACS ordering (size of Antecedent, Confidence and Support) which is defined in a similar manner to CSA (see above) but with size of antecedent placed first. The intuition behind this ordering is that more specific rules should be "triggered" before more general rules are attempted. For example we may have a classifier, ordered using CSA, comprising two rules as follows:

#	Rule	Conf.
1	$\{ab\} \Rightarrow \{y\}$	80%
2	$\{abcd\} \Rightarrow \{x\}$	75%

Given a case $\{abc\}$ this would be classified, using "best first" case satisfaction, as belonging to class y when intuitively class x would be more likely to be the correct class. ACS ordering thus ensures that specific rules have a higher precedence than more general rules so that in the above example the class x would be returned. It should be noted, however, that for ACS to work well a high confidence threshold value should be used. An appropriate mechanism to prevent *overfitting* must also be incorporated.

3 Classification

In the introduction to this paper three alternative case/record classification mechanisms were identified. Two of these, "best k" and "χ^2 testing" are briefly discussed below so as to provide some necessary further detail ("best first" has the obvious interpretation).

3.1. Best K Testing

The intuition behind "best k testing" is that 'one cannot expect that any single rule can perfectly predict the class label for every example satisfying its body' [8]. Given a case n to be classified the best K approach is as follows: (1) obtain all rules that satisfy n; (2) keep only best K rules for each class, or all rules if there are less than K rules for a particular class; (3) for each group determine some average expected value to be maximised (e.g. confidence, size of antecedent, Laplace accuracy, χ^2 value); (4) Select the class associated with the best average. In [8] a value of 5 was suggested as an appropriate value for K.

3.2. Weighted χ^2 Testing

Weighted χ^2 Testing is used in a number of CARM algorithms, such as CMAR [6], to classify data by considering entire groups of rules that satisfy a given case. With respect to CMAR, given a case n to be classified, the procedure commences by first collecting all rules that satisfy n. Then if the consequents of all rules are identical, or only one rule is found, classify case according to the consequents; otherwise group rules according to class and determine the combined effect of the rules in each group (the class associated with the "strongest group" is then selected). The strength of a group is calculate using the WCS (Weighted χ Squared) value. The class associated with the group of rules with the highest WCS value is then selected as the class to be allocated to the case.

4. Apriori-TFPC

The Apriori-TFPC classification rule generation algorithm is founded on the Apriori-TFP (Total From Partial) ARM algorithm [4][1]. This algorithm generates frequent sets that are placed as nodes in a set enumeration tree. To evaluate the above approaches a number of variations of the algorithm were created, each reflecting one of the identified approaches.

Uniquely, in Apriori-TFPC CRs are generated as part of the "frequent set identification process". As the tree is developed nodes in branches whose root represents a classifier are tested for their appropriateness as classification rules using a confidence threshold (to evaluate the different techniques considered in this paper this can equally well be achieved using a χ^2 threshold, Laplace accuracy or a WRA measure). If a node represents a suitable CR, the rule is placed in a list and the node not processed any further. Nodes are also pruned during the generation process according to a user supplied support threshold. This tree pruning is intended to prevent *overfitting*.

The different ordering and case satisfaction techniques considered in this paper can be combined into eleven different variations of Apriori-TFPC (see Table 1). With respect to ACS ordering, note that tree pruning is still carried out according to confidence. In the case of experiments using "best K" techniques K was set to 5.

5. Evaluation

Experiments were conducted using a range of data sets taken from the the UCI Machine Learning Repository [3]. The chosen datasets were discretized using the LUCS-KDD DN software[2], where appropriate continuous attributes were ranged using five sub-ranges. The experiments were run on a 1.2 GHz Intel Celeron CPU with 512 Mbyte of RAM running under Red Hat Linux 7.3.

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 [6] and CBA [7]). The results are presented in Table 1 where the best accuracy obtained for each of the data sets is highlighted in bold print. The row labels describe the key characteristics of each data set: for example, the label $adult.D131.N48842.C2$ denotes the "adult" data set, which includes 48842 records in 2 classes, with attributes that for the experiments described here have been discretised into 131 binary categories.

It should be noted that the datasets were rearranged so that occurences 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, it is the relative accuracy that is of interest here and not the absolute accuracy.

From Table 1 it can be seen that with a 50% confidence threshold the proposed ACS ordering worked reasonably well but not as well as was hoped. It is surmised that this is probably because many specific rules with relatively low confidence were given a high precedence over higher confidence but more general rules. The last four columns of

[1]Apriori-TFP and Apriori-TFPC my be obtained from http://www.csc.liv.ac.uk/ frans/KDD/Software.

[2]The LUCS-KDD DN is available at http://www.csc.liv.ac.uk/ ̃frans/KDD/Software/LUCS-KDD-DN/.

Table 1 show the results of a further set of experiments conducted using a confidence threshold of 75%. In this case best results were obtained using "best first" and ACS. ACS with "best first" also produces the greatest number of best accuracies (10 out of 22). The experiment also illustrated that by reducing the overall number of rules (by increasing the confidence requirement) the "best K" approach deteriorated.

Data Set	$confidence = 50\%$											$confidence = 75\%$			
	Best first					*Best K first*					*All*	*Best first*		*Best K first*	
	CSA	*ACS*	*WRA*	*Lap.*	χ^2	*CSA*	*ACS*	*WRA*	*Lap.*	χ^2	*WCS*	*CSA*	*ACS*	*CSA*	*ACS*
adult.D131.N48842.C2	76.1	76.1	65.2	76.1	32.8	76.0	**76.3**	57.8	76.1	33.7	76.1	80.7	76.1	**80**.9	76.4
anneal.D106.N798.C6	**85.5**	**85.5**	68.4	83.7	59.1	79.4	79.5	69.2	82.2	42.3	80.7	88.0	85.5	**89.5**	**89.5**
auto.D142.N205.C7	12.7	19.6	**54.9**	48.0	19.6	13.7	13.7	32.4	43.1	12.7	20.6	13.7	12.7	12.7	**19.6**
breast.D47.N699.C2	**98.0**	**98.0**	81.1	96.6	83.1	93.4	93.4	72.2	91.7	88.5	**98.0**	**98.0**	**98.0**	**98.0**	97.1
con4.D129.N67557.C3	65.8	65.8	37.3	65.8	61.1	65.8	65.8	62.4	65.8	**65.9**	65.8	65.8	65.8	**65.9**	**65.9**
glass.D52.N214.C7	**49.5**	38.3	34.6	46.7	13.1	42.1	42.1	26.2	43.0	13.1	13.1	32.7	36.4	**46.7**	45.8
heart.D53.N303.C5	49.7	54.3	**57.6**	55.0	55.0	53.0	53.0	56.3	55.0	55.0	55.0	43.0	**54.3**	29.8	29.8
hepatitis.D58.N155.C2	71.4	79.2	67.5	79.2	20.8	**80.5**	**80.5**	70.1	79.2	28.6	61.0	64.9	**79.2**	53.2	53.2
horseCol.D94.D368.C2	67.9	78.2	**83.7**	77.8	62.5	70.7	70.1	78.3	66.8	51.1	62.5	60.9	**77.1**	47.8	58.7
iono.D172.N351.C2	81.7	**91.4**	76.0	87.4	84.6	77.1	88.0	62.9	82.9	69.7	62.9	91.4	91.4	66.3	**94.3**
iris.D23.N150.C3	**94.7**	**94.7**	93.3	93.3	93.3	92.0	92.0	93.3	92.0	89.3	**94.7**	89.3	**94.7**	88.0	88.0
led7.D24.N3200.C10	67.1	57.2	32.9	66.6	26.0	66.2	66.3	31.9	66.8	19.1	**73.8**	63.4	**64.6**	64.1	64.1
letRec.D106.N20000.C26	**44.2**	34.9	16.9	42.4	28.1	41.0	41.2	8.1	41.3	25.6	38.1	28.6	**29.1**	28.5	28.4
mushr'm.D127.N8124.C2	**96.0**	89.1	47.5	46.7	88.4	71.1	69.7	66.9	69.8	85.6	53.1	**96.2**	89.1	82.0	81.9
nursery.D32.N12960.C5	80.0	74.4	70.6	80.0	70.6	69.8	69.9	70.3	70.6	68.6	**85.3**	**89.6**	87.2	87.6	88.3
pageBl'ks.D55.N5473.C5	**89.8**	**89.8**	79.6	**89.8**	3.4	**89.8**	**89.8**	68.5	**89.8**	5.4	2.0	**89.8**	**89.8**	**89.8**	**89.8**
penDig.D90.N10992.C10	**79.5**	40.6	33.1	78.1	49.9	54.4	54.8	35.2	55.0	46.8	70.8	**83.0**	79.3	71.0	71.2
pimaInd.D42.N768.C2	74.2	74.2	**76.0**	74.5	65.1	71.3	73.7	66.9	70.6	62.0	70.6	76.0	**76.8**	73.7	73.7
ticTacToe.D29.N958.C2	66.6	66.0	65.8	66.0	65.8	64.3	51.4	54.9	64.9	65.8	**78.9**	**69.1**	66.2	55.9	55.7
wave.D108.N5000.C3	66.2	61.1	62.5	66.4	62.5	57.8	58.3	60.0	58.7	60.0	**77.6**	**76.4**	67.8	66.0	66.8
wine.D68.N178.C3	70.8	82.0	86.5	**92.1**	**92.1**	67.4	67.4	71.9	80.9	84.3	91.0	70.8	**83.1**	27.0	28.1
zoo.D43.N101.C7	88.0	80.0	54.0	62.0	74.0	70.0	74.0	58.0	62.0	62.0	**92.0**	**86.0**	78.0	66.0	66.0
Average	71.6	69.6	61.1	71.6	55.0	66.7	66.9	57.9	68.6	51.6	64.7	70.8	71.9	63.2	65.1

Table 1 *Classification accuracy* ($minsup = 1\%$)

6. Conclusion

In this paper a number of alternative rule ordering and case satisfaction strategies have been considered. Four established ordering strategies were examined. In addition the authors proposed a fifth strategy, ACS, where more specific rules are given a higher precedence than less specific rules (but using confidence as the most significant factor for tree pruning during the generation process).

The principal findings of the evaluation are as follows: (1) there is no overall best ordering suited to all the data sets used in the experiments, (2) the "best first" case satisfaction mechanism works better than "best k" in all the data sets tested, (3) ACS ordering produces the best result provided that a relatively high confidence threshold is used (a threshold of 75% is suggested), (4) for lower confidence thresholds (50% to 75%) CSA and Laplace ordering coupled with a "best first" case satisfaction produced good results.

References

[1] Agrawal, R. and Srikant, R. (1994). *Fast algorithms for mining association rules*. Proc. VLDB'94, Morgan Kaufman, pp487-499.

[2] Bayardo, R.J. (1997). *Brute-Force Mining of High-Confidence Classification Rules*. Proc. of 3rd Int. Conf. on Knowledge Discovery and Data Mining, AAAI, pp123-126.

[3] Blake, C.L. and Merz, C.J. (1998). *UCI Repository of machine learning databases*. http://www.ics.uci.edu/ mlearn/MLRepository.html, Irvine, CA: University of California.

[4] Coenen, F., Leng, P., Goulbourne, G. (2004). *Tree Structures for Mining Association Rules.* Jo. of Data Mining and Knowledge Discovery, 8(1), pp25-51.

[5] Lavrač, N., Flach, P. and Zupan, B. (1999) *Rule Evaluation Measures: A Unifying View* Proc. 9th Int. Workshop on Inductive Logic Programming (ILP'99), Springer-Verlag, pp174–185.

[6] Li W., Han, J. and Pei, J. (2001). *CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules.* Proc ICDM 2001, pp369-376.

[7] Liu, B. Hsu, W. and Ma, Y (1998). *Integrating Classification and Association Rule Mining.* Proceedings KDD-98, AAAI, pp80-86.

[8] Yin, X. and Han, J. (2003). *CPAR: Classification based on Predictive Association Rules.* Proc. SIAM Int. Conf. on Data Mining (SDM'03), pp331-335.