OBTAINING BEST PARAMETER VALUES FOR ACCURATE CLASSIFICATION

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 we examine the effect that the choice of support and confidence thresholds has on the accuracy of classifiers obtained by Classification Association Rule Mining. We show that accuracy can almost always be improved by a suitable choice of threshold values, and we describe a method for finding the best values. We present results that demonstrate this approach can obtain higher accuracy without the need for coverage analysis of the training data.

Keywords: *Classification, Association Rule Mining*

1 INTRODUCTION

A method of classification that has attracted recent attention is to make use of Association Rule Mining (ARM) techniques to define Classification Rules. Examples of Classification Association Rule Mining (CARM) methods include PRM and CPAR [4], CMAR [2] and CBA [3]. In general, CARM algorithms begin by generating all rules that satisfy threshold values of *support* (the number of instances in the training data for which the rule is found to apply) and *confidence* (the ratio of its support to the total number of instances of the rule's antecedent). The candidate rules are then pruned and ordered using other techniques. CBA ([3]) generates rules which are prioritised, using confidence, support, and rule-length, then pruned by *coverage analysis*, in which each record in the training set is examined to identify a rule that classifies it correctly. The CMAR algorithm ([2]) has a similar general structure to CBA, but uses a different coverage procedure that may generate more than one rule for each case.

The cost of coverage analysis, especially when dealing with large data sets with many attributes, motivated us to consider whether it is possible to generate an accurate set of Classification Rules directly from an ARM process without coverage analysis. In [1] we described an algorithm, TFPC, of this kind. The heuristic applied by TFPC is that once a general rule is found that satisfies the required thresholds of support and confidence, no more specific rules (rules with the same consequent, whose antecedent is a superset) will be considered. This provides a very efficient method for generating a relatively compact set of CRs. Because no coverage analysis is carried out, however, the choice of appropriate support and confidence thresholds is critical in determining the final rule set.

In this paper we examine the effect of varying these thresholds on the accuracy of both TFPC and other algorithms. We show that classification accuracy can be significantly improved, in most cases, by an appropriate choice of values. We describe a *hill climbing* algorithm which aims to find the "best" thresholds from examination of the training data. We show that this procedure can lead to higher classification accuracy at lower cost than methods of coverage analysis.

2 Finding best threshold values

To examine the effect that may result from varying threshold values, we carried out experiments using test data from the UCI Machine Learning Repository, discretized using the LUCS-KDD DN software [1]. Here, for example, the label **glass.D48.N214.C7** refers to the "glass" data set, which includes 214 records in 7 classes, with attributes which have been discretised into 48 binary categories. Using this data, we investigated the classification accuracy that can be achieved using the TFPC algorithm, and also from CMAR and CBA, across the full range of values for the support and confidence thresholds. For each (support, confidence) pair we obtained a classification accuracy from a division of the full data set into a 90% training set and 10% test set. Figure 1 illustrates a selection of results in the form of 3-D plots, in which the X and Y axes represent support

[1] Available at $http://www.csc.liv.ac.uk \sim frans/KDD/Software/LUCS-KDD-DN/lucs-kdd\_DN.html$

and confidence threshold values, and the Z axis the corresponding classification accuracy obtained.

These results demonstrate that the accuracy obtained can be very sensitive to the choice of thresholds. The coverage analysis used in CMAR and CBA usually smooths out some of the influence of this, provided sufficiently low thresholds are chosen. The smoothing is not perfect, however; **ticTacToe**, for example, illustrates a case where a low confidence threshold leads to a selection of poor rules by CMAR, and CBA performs badly for **wine** if a low support threshold is chosen. For CBA, the example of **ionosphere** shows a case where a poor choice of thresholds (even values that appear reasonable) may lead to a dramatically worse result. This is partly because, unlike CMAR, CBA's coverage analysis may sometimes retain a rule that applies only to a single case. This makes the method liable to include spurious rules, especially if the data set is small enough for these to reach the required thresholds.

It is apparent that the accuracy of the classifiers obtained using any of these methods may be improved by a careful selection of these thresholds. To obtain these values, we apply a procedure for identifying the threshold values that lead to the highest classification accuracy from a particular training set. The method applies a "hill-climbing" strategy that makes use of a 3-D playing area measuring $100 \times 100 \times 100$, as visualised in the illustrations discussed above. The procedure commences with initial support and confidence threshold values, describing a current location (cl) in the base plane of the playing area. Using these values, the chosen rule-generation algorithm is applied to the training data, and the resulting classifier applied to the test data, with appropriate cross-validation, to obtain a classification accuracy for cl.

The procedure then moves round the playing area with the aim of improving the accuracy value. To do this it continuously generates data for a set of eight test locations, defined by applying two values, δs and δc as positive and negative increments of the support and confidence threshold values associated with cl. The rule-generation algorithm is applied to obtain a classification accuracy for each of the test locations which is inside the playing area and for which no accuracy value has previously been calculated. The location for which the highest accuracy is obtained is selected as cl for the next iteration. If the current cl has the best accuracy, then the threshold increments are reduced and a further iteration of test locations takes place. The process concludes when no improvement in accuracy can be obtained. The final cl selected will be at worst a local optimum.

3 RESULTS

Table 1 summarises the results of applying the hill-climbing procedure described above to datasets from the UCI repository, for the algorithms TFPC, CMAR and CBA. For each algorithm, the first two columns in the table show the average accuracy obtained from applying the algorithm to (90%, 10%) divisions of the dataset with ten-fold cross-validation. The first of the two columns shows the result for a support threshold of 1% and a confidence threshold of 50% (the values usually chosen in analysis of classification algorithms), and the second after applying the hill-climbing procedure to identify the "best" threshold values. For these experiments δS and δC were set to 6.4 and 8.0 respectively, and $min\delta S$ and $min\delta C$ to 0.1 and 1.0. In each case the threshold values that produced the best accuracy are also tabulated.

Table 1 confirms the picture suggested by the illustrations in Figure 1 (although note the correspondence is not exact, as cross-validation was not used in obtaining the graphical representations). In almost all cases, an improved accuracy can be obtained from a pair of thresholds different from the default (1%, 50%) choice. As would be expected, the greatest gain from the hill-climbing procedure is in the case of TFPC, but a better accuracy is also obtained for CMAR in 21 of the 25 sets, and for CBA in 20. In a number of cases the improvement is substantial. It is apparent that CBA, especially, can give very poor results with the default threshold values. In the cases of **ionosphere** and **wine**, the illustrations reveal the reason to be that a 1% support threshold leads, for these small data sets, to the selection of spurious rules. This is also the case for **zoo** and **hepatitis**, and for **mushroom**, where even a much larger data set includes misleading instances if a small support threshold is chosen. In the latter case the hill-climbing procedure has been ineffective in escaping a poor local optimum. Notice that here the coverage analysis used in CMAR is much more successful in identifying the best rules, although TFPC also does relatively well. CMAR is generally less sensitive than CBA to the choice of thresholds, but both methods give very poor results when, as in the cases of **chess** and **letrecog**, the chosen confidence threshold is too high, and CMAR performs relatively poorly for **led7** for the same reason. The extreme case is **chess**, where both CMAR and CBA (and TFPC) find no rules at the 50% confidence threshold. Notice, also, that for the largest data sets (those with more than 5000 cases) a support threshold lower than 1% almost always produces better results, although the additional candidate rules generated at this level will make coverage analysis more expensive.

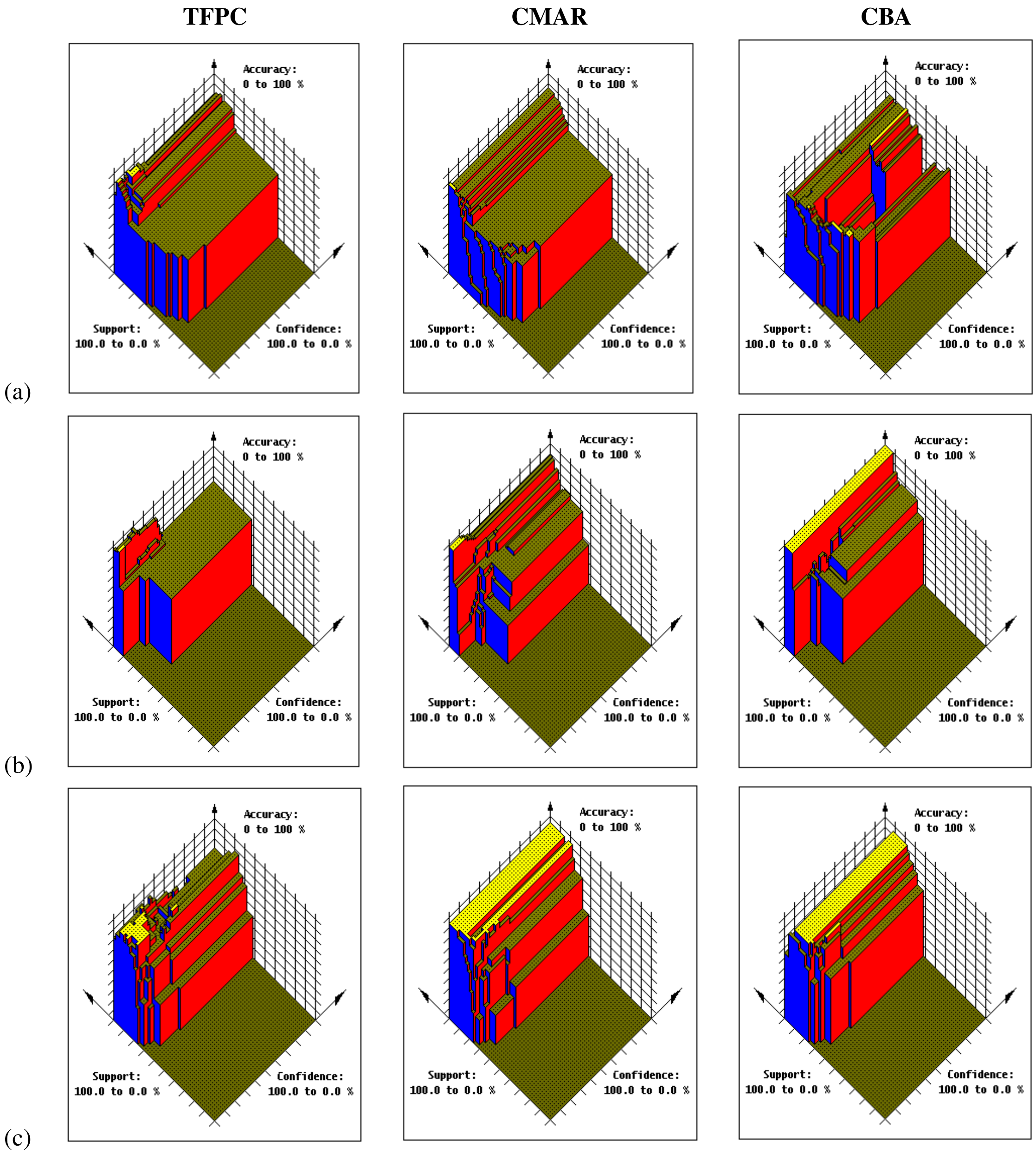

Figure 1 *3-D Plots: (a) ionosphere.D157.N351.C2, (b) ticTacToe.D29.N958.C2 and (c) wine.D68.N178.C3*

In general, the results show that coverage analysis, especially in CMAR, is usually (although not always) effective in minimising any adverse effect from a poor choice of thresholds. Although TFPC with the default threshold values produces reasonably high accuracy in most cases, the lack of coverage analysis generally leads to somewhat lower accuracy than one or both of the other methods. However, the results when the hill-climbing procedure is applied to TFPC show that high accuracy can be obtained without coverage analysis if a good choice of thresholds is made. In 18 of the 25 cases, the accuracy of TFPC after hill-climbing is as good or better than that of CMAR with the default thresholds, and in only one case (**wine**) is it substantially worse, the hill-climbing in this case failing to find the peak of the rather irregular terrain shown in the illustration. Conversely, the result for **penDig** demonstrates a case in which the hill-climbing procedure of TFPC works better than the coverage analysis of CMAR in identifying important low-support, high-confidence rules. The results also improve on CBA in 14 cases, often by a large margin. Overall this suggests that a good choice of thresholds can eliminate the need for coverage analysis procedures.

The significance of this is that coverage analysis is relatively expensive, especially if the data set and/or the number of candidate rules is large, as is likely to be the case if a low support threshold is chosen. The final four columns of Table 1 give a comparison of the total execution times to construct a classifier with ten-fold cross validation, using TFPC, with or without the hill-climbing procedure, and for CMAR and CBA (for the 1%, 50% thresholds). These figures were obtained using our Java 1.4 implementations on a single Celeron 1.2 Ghz CPU with 512 MBytes of RAM.

Data set	TFPC				CMAR				CBA				Execution Time			
	Def. val.	HC	"best" t'hold		Def. val.	HC	"best" t'hold		Def. val.	HC	"best" t'hold		TFPC	TFPC HC	CMAR	CPAR
			S	C			S	C			S	C				
adult.D97.N48842.C2	80.8	81.0	0.2	50.1	80.1	80.9	0.7	50.0	84.2	84.6	0.1	48.4	2.9	20.0	78.0	230.0
anneal.D73.N898.C6	88.3	90.1	0.4	49.1	90.7	91.8	0.4	50.0	94.7	96.5	0.8	46.8	0.5	2.7	2.3	5.8
auto.D137.N205.C7	70.6	75.1	2.0	52.4	79.5	80.0	1.2	50.0	45.5	77.5	2.7	50.8	3.3	61.7	703.9	536.3
breast.D20.N699.C2	90.0	90.0	1.0	50.0	91.2	91.2	1.0	50.0	94.1	94.1	1.0	50.0	0.3	0.3	0.4	0.6
chess.D58.N28056.C18	0.0	38.0	0.1	25.2	0.0	34.6	0.1	11.0	0.0	39.8	0.1	24.0	2.1	46.7	2.0	2.0
cylBds.D124.N540.C2	68.3	74.4	1.2	49.8	75.7	77.8	1.3	49.9	75.7	78.0	1.9	50.0	4.0	163.9	206.9	923.6
flare.D39.N1389.C9	84.3	84.3	1.0	50.0	84.3	84.3	1.0	50.0	84.2	84.2	1.0	50.0	0.4	0.5	1.0	2.4
glass.D48.N214.C7	64.5	76.2	2.6	45.6	75.0	75.0	1.0	50.0	68.3	70.7	3.0	51.6	0.4	1.1	0.8	0.8
heart.D52.N303.C5	51.4	56.0	4.2	52.4	54.4	54.8	1.6	50.0	57.3	60.0	4.2	49.2	0.6	2.7	0.9	1.4
hepatitis.D56.N155.C2	81.2	83.8	1.6	51.6	81.0	82.8	2.9	50.0	57.8	83.8	7.1	48.4	0.6	2.4	2.4	10.0
horseCol.D85.D368.C2	79.1	79.9	1.2	50.2	81.1	81.9	2.9	50.0	79.2	83.9	5.5	49.2	0.5	2.1	10.5	66.5
ion'sph.D157.N351.C2	85.2	92.9	9.8	50.0	90.6	91.5	2.6	50.0	31.6	89.5	10.0	49.2	2.3	16.3	3066.8	2361.1
iris.D19.N150.C3	95.3	95.3	1.0	50.0	93.3	94.7	2.3	50.0	94.0	94.0	1.0	50.0	0.3	0.4	0.3	0.3
led7.D24.N3200.C10	57.3	62.7	2.2	49.4	62.2	67.4	1.3	40.4	66.6	68.0	1.0	46.0	0.4	1.4	0.6	0.7
letRc.D106.N20K.C26	26.4	47.6	0.1	32.3	25.5	45.5	0.1	31.8	28.6	58.9	0.1	13.7	3.7	196.2	17.2	20.5
mush'm.D90.N8124.C2	99.0	99.7	1.8	69.2	100.0	100.0	1.0	50.0	46.7	46.7	1.0	50.0	1.4	30.6	269.0	366.2
nurs'ry.D32.N12960.C5	77.8	89.9	1.0	73.2	88.3	90.1	0.8	62.6	90.1	91.2	1.5	50.0	1.3	21.3	5.8	6.9
pgBlks.D46.N5473.C5	90.0	90.0	1.0	50.0	90.0	90.3	0.2	50.0	90.9	91.0	1.6	50.0	0.3	0.7	0.8	2.2
penD.D89.N10992.C10	81.7	88.5	0.1	62.3	83.5	85.2	0.8	50.0	87.4	91.4	0.1	50.9	3.7	227.8	39.3	43.6
pima.D38.N768.C2	74.4	74.9	2.3	50.0	74.4	74.5	1.6	50.0	75.0	75.7	2.8	50.0	0.3	0.4	0.4	0.7
soyLrg.D118.N683.C19	89.1	91.4	1.1	49.1	90.8	91.8	0.8	51.6	91.0	92.9	0.6	52.2	9.8	644.3	405.6	273.8
ticTacToe.D29.N958.C2	67.1	96.5	1.5	74.2	93.5	94.4	1.6	50.0	100.0	100.0	1.0	50.0	0.4	4.5	1.0	1.6
wavef'm.D101.N5K.C3	66.7	76.6	3.2	64.3	76.2	77.2	0.6	50.0	77.6	78.2	2.6	50.0	3.7	210.8	167.3	93.4
wine.D68.N178.C3	72.1	81.9	4.5	51.2	93.1	94.3	2.3	50.0	53.2	65.5	4.8	50.0	0.3	1.0	7.3	11.7
zoo.D42.N101.C7	93.0	94.0	1.0	49.2	94.0	95.0	1.6	50.0	40.4	93.1	7.4	50.0	0.5	2.5	3.1	3.2

Table 2. *Accuracy and performance results (Default values: confidence = 50%, support = 1%)*

As would be expected, the execution times for TFPC (with default threshold values) are almost always far lower than for either of the other two methods. Less obviously, performing hill-climbing with TFPC is in many cases faster than coverage analysis with CMAR or CBA. In 13 of the 25 cases, this was the fastest procedure to obtain classification rules, and it is only markedly worse in cases such as **chess** and **letRecog**, where the other methods have failed to identify the rules necessary for good classification accuracy. These results suggest that TFPC with hill-climbing is an effective way of generating an accurate classifier which is often less costly than other methods.

4 CONCLUSIONS

In this paper we have shown that the choice of appropriate values for the support and confidence thresholds can have a significant effect on the accuracy of classifiers obtained by CARM algorithms. The coverage analysis performed by methods such as CMAR and CBA reduces this effect, but does not eliminate it. CMAR appears to be less sensitive than CBA to the choice of threshold values, but for both methods better accuracy can almost always be obtained by a good choice. We have also shown that, if threshold values are selected well, it is possible to obtain good classification rules using a simple and fast algorithm, TFPC, without the need for coverage analysis. We describe a procedure for finding these threshold values that will lead to good classification accuracy. Our results demonstrate that this approach can lead to improved classification accuracy, at a cost that is comparable to or lower than that of coverage analysis.

References

[1] Coenen, F.,Leng, P. and Zhang, L (2005). *Threshold Tuning for Improved Classification Association Rule Mining* Proc PAKDD 2005: LNCS 3518, Springer, pp216-225

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

[3] Liu, B. Hsu, W. and Ma, Y (1998). *Integrating Classification and Association Rule Mining.* Proceedings KDD-98, New York, 27-31 August. AAAI. pp80-86.

[4] Yin, X. and Han, J. (2003). *CPAR: Classification based on Predictive Association Rules.* Proc. SIAM Int. Conf. on Data Mining (SDM'03), San Francisco, CA, pp. 331-335.