# Improving the Accuracy of CAR-Based Classifiers by Combining Netconf Measure and Dynamic$-K$ Mechanism

Raudel Hernàndez-Leòn(✉)

Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV),
7a ♯ 21406 e/214 and 216, Rpto. Siboney, Playa, 12200 La Habana, Cuba
rhernandez@cenatav.co.cu

**Abstract.** In this paper, we propose combining Netconf as quality measure and Dynamic$-K$ satisfaction mechanism into Class Association Rules (CARs) based classifiers. In our study, we evaluate the use of several quality measures to compute the CARs as well as the main satisfaction mechanisms ("Best Rule", "Best $K$ Rules" and "All Rules") commonly used in the literature. Our experiments over several datasets show that our proposal gets the best accuracy in contrast to those reported in state-of-the-art works.

**Keywords:** Supervised classification · Satisfaction mechanisms · Class association rules

## 1 Introduction

Associative classification, introduced in [2], integrates Association Rule Mining (ARM) and Classification Rule Mining (CRM). This integration involves mining a special subset of association rules, called Class Association Rules (CARs), using some quality measures (QM) to evaluate them. A classifier based on this approach usually consists of an ordered CAR list $l$, and a satisfaction mechanism for classifying unseen transactions using $l$ [2,3,6].

Associative classification has been applied to many tasks including prediction of consumer behavior [17], automatic error detection [19], breast cancer detection [15], and prediction of protein-protein interaction types [16], among others.

In associative classification, similar to ARM, a set of items $I = \{i_1, \ldots, i_n\}$, a set of classes $C$, and a transactional dataset $T$, are given. Each transaction $t \in T$ is represented by a set of items $X \subseteq I$ and a class $c \in C$. A lexicographic order among the items of $I$ is assumed. The Support of an itemset $X \subseteq I$, denoted as $Sup(X)$, is the fraction of transactions in $T$ containing $X$ (see Eq. 1).

$$Sup(X) = \frac{|T_X|}{|T|} \tag{1}$$

where $T_X$ is the set of transactions in $T$ containing $X$ and $|\cdot|$ represents the cardinality.


A. Pardo and J. Kittler (Eds.): CIARP 2015, LNCS 9423, pp. 603–610, 2015.
DOI: 10.1007/978-3-319-25751-8_72

A CAR is an implication of the form $X \Rightarrow c$ where $X \subseteq I$ and $c \in C$. The most commonly used measures to evaluate CARs are Support and Confidence. The rule $X \Rightarrow c$ is held in $T$ with certain Support $s$ and Confidence $\alpha$, where $s$ is the fraction of transactions in $T$ that contains $X \cup \{c\}$ (see Eq. 2), and $\alpha$ is the probability of finding $c$ in transactions that also contain $X$ (see Eq. 3), which represents how "strongly" the rule antecedent $X$ implies the rule consequent $c$. A CAR $X \Rightarrow c$ satisfies or covers a transaction $t$ if $X \subseteq t$.

$$Sup(X \Rightarrow c) = Sup(X \cup \{c\}) \tag{2}$$

$$Conf(X \Rightarrow c) = \frac{Sup(X \Rightarrow c)}{Sup(X)} \tag{3}$$

However, in [4], the authors analyzed several measures (Conviction, Lift, and Certainty factor), as an alternative to the Support and Confidence measures, for estimating the strength of an association rule.

The rest of the paper is organized as follows. The related work is described in next section. Our proposal is presented in section three. In the fourth section the experimental results are shown. Finally, conclusions are given in section five.

## 2 Related Work

In general, CAR-based classifiers could be divided in two groups according to the strategy used for computing the set of CARs:

1. Two Stage classifiers. In a first stage, all CARs satisfying the Support and Confidence values (or other measures) are mined and later, in a second stage, a classifier is built by selecting a small subset of CARs that fully covers the training set, CBA [2] and CMAR [3] follow this strategy.
2. Integrated classifiers. In these classifiers a small subset of CARs is directly generated using different heuristics, CPAR [6], TFPC [9] and DDPMine [13] follow this strategy.

Regardless of the strategy used for computing the set of CARs, in order to build the classifier we need to sort the CARs. In the literature, there are six main strategies for ordering CARs:

a) CSA (Confidence - Support - Antecedent size): First, the rules are sorted in a descending order according to their Confidence. In case of ties, CARs are sorted in a descending order according to their Support, and if the tie persist, CSA sorts the rules in ascending order according to the size of their rule antecedent. This strategy has been used by the CBA classifier [2].
b) ACS (Antecedent size - Confidence - Support): This strategy is a variation of CSA, but it takes into account the size of the rule antecedent as first ordering criterion followed by Confidence and Support. The classifier TFPC [9] follows this ordering strategy.

c) SrQM (Specific rules (Sr) - Quality Measure (QM)): First, the rules are sorted in a descending order according to the size of the CARs and in case of tie, the tied CARs are sorted in a descending order according to their QM value. CAR-IC classifier follows this ordering strategy [18].
d) WRA (Weighted Relative Accuracy): The WRA rule ordering strategy assigns to each CAR a weight and then sorts the set of CARs in a descending order according to the assigned weights [12,14]. The WRA has been used to order CARs in two versions of the TFPC classifier [12,14]. Given a rule $X \Rightarrow Y$ the WRA is computed as follows:

$$WRA(X \Rightarrow Y) = Sup(X)(Conf(X \Rightarrow Y) - Sup(Y))$$

e) LAP (Laplace Expected Error Estimate): LAP was introduced by Clark and Boswell [1] and it has been used to order CARs in CPAR classifier [6]. Given a rule $X \Rightarrow Y$, in [6] the LAP is defined as follows:

$$LAP(X \Rightarrow Y) = \frac{Sup(X \Rightarrow Y) + 1}{Sup(X) + \mid C \mid}$$

where $C$ is the set of predefined classes.
f) $\chi^2$ (Chi-Square): The $\chi^2$ rule ordering strategy is a well known technique in statistics, which is used to determine whether two variables are independent or related. After computing an additive $\chi^2$ value for each CAR, this value is used to sort the CARs in a descending order in the CMAR classifier [3].

In [18], the authors show that the $SrQM$ rule ordering strategy obtains the best results of all strategies mentioned above. Once the classifier has been built, we need to select a satisfaction mechanism for classifying unseen transactions. Four main satisfaction mechanisms have been reported [2,3,14,21]:

1. **Best Rule**: It selects the first ("best") rule in the order that satisfies the transaction to be classified, and then the class associated to the selected rule is assigned to this transaction [2].
2. **Best K Rules**: It selects the best $K$ rules (per each class) that satisfy the transaction to be classified and then the class is determined using these $K$ rules, according to different criteria [14].
3. **All Rules**: It selects all rules that satisfy the transaction to be classified and use them to determine the class of the new transaction [3].
4. **Dynamic**$-K$: It is similar to the "Best $K$ Rules" mechanism but the value of $K$ may change for each transaction to be classified [21].

Classifiers following the "Best Rule" mechanism could suffer biased classification or overfitting since the classification is based on only one rule. On the other hand, the "All Rules" mechanism includes rules with low ranking for classification and this could affect the accuracy of the classifier. The "Best K Rules" mechanism has been the most used satisfaction mechanism for CAR-based classifiers, reporting the best results. However, in [21] the authors mention some limitations of this mechanism. Also in [21], the authors proposed the Dynamic$-K$

satisfaction mechanism, which does not have the drawbacks of the other three mechanisms (see next section).

In this paper, we propose to combine the Dynamic$-K$ satisfaction mechanism and the Netconf quality measure into CAR-based classifiers. In order to show the suitability of our proposal, we evaluate several quality measures as well as the other reported satisfaction mechanisms. Experiments over several datasets show that our proposal gets the best performance in contrast to those reported in state-of-the-art works.

## 3 Our Proposal

In the next subsections we describe the Dynamic$-K$ satisfaction mechanism (subsection 3.1) as well as the Netconf measure (subsection 3.2). Based on the advantages of Dynamic$-K$ over the other satisfaction mechanisms and based on the characteristics of Netconf measure, we propose in this paper to improve the accuracy of CAR-based classifiers by combining them.

### 3.1 Dynamic$-K$ Satisfaction Mechanism

As we mentioned in related works, the main satisfaction mechanisms reported have limitations that can affect the classification accuracy. In general, the "Best $K$ Rules" mechanism has been the most widely used for CAR-based classifiers, reporting the best results [11]. However, in [21] the authors show that using this mechanism could affect the classification accuracy. Ever more when most of the best $K$ rules were obtained extending the same item, or when there is an imbalance among the numbers of CARs with high measure values, per each class, that cover the new transaction (see some examples in [21]).

In order to overcome these drawbacks, the Dynamic$-K$ mechanism was proposed in [21]. First, Dynamic$-K$ sorts the CARs using the SrQM rule ordering strategy. Later, it selects, for each class $c \in C$, the set of rules $X \Rightarrow c$ covering the new transaction $t$ and satisfying the following conditions:

- $X \Rightarrow c$ is a maximal rule.
- for all $i \in I$, with $i$ lexicographically greater than all items of $X$, $QM(X \cup \{i\} \Rightarrow c) < QM(X \Rightarrow c)$ holds.

Thereby they included more large rules with high quality measure values in the classification, avoiding redundancies and including more different items in the antecedents of the selected CARs.

Let $N_i$ be the set of maximal CARs of class $c_i$ that were selected for Dynamic$-K$ mechanism. After selecting all $N_i$ (for $i = 1$ to $|C|$), Dynamic$-K$ assigns the class $c_j$ such that the QM average of all rules of $N_j$ is greater than the QM average of the top $|N_j|$ rules of each $N_i$, with $i \neq j$ and $|N_i| \geq |N_j|$. In case of tie among classes with different number of CARs, the class with less number of CARs is preferred because the CARs are sorted in descendent order according

to their sizes (SrQM rule ordering strategy); in case of tie among classes with equals number of CARs, the class with greater Support is selected, if the tie persist the class is selected randomly.

Resuming, the Dynamic$-K$ mechanism does not have the drawbacks of the other existent mechanisms since:

- It selects the maximal rules with high QM values, avoiding redundancies and allowing the inclusion of more different items in the antecedents of the selected CARs, thereby CARs of low quality are not included for classifying.
- The result is not affected when there is an imbalance among the numbers of CARs with high QM values, for each class, that cover the new transaction, this happens because to classify a new transaction, Dynamic$-K$ considers the average of the same amount of CARs.
- It considers all good quality CARs that cover the new transaction and not only the best one. Thereby, Dynamic$-K$ does not fall on the mistake of assuming that the best rule is going to classify correctly all transactions that it covers.

### 3.2 Main Quality Measures

In [4], the authors analyzed several measures (Conviction, Lift, and Certainty factor), as an alternative to the Confidence measure, for estimating the strength of an association rule. As an important result, they show that some of these measures overcome the drawbacks of the Confidence. However, in case of Lift and Certainty factor, they have other limitations.

The Lift measure (see Eq. 4) has a not bounded range [4], therefore differences among its values are not meaningful and for this reason, it is difficult to define a Lift threshold.

$$Lift(X \Rightarrow Y) = \frac{Sup(X \Rightarrow Y)}{Sup(X)Sup(Y)} \tag{4}$$

On the other hand, Certainty factor is defined by Eq. 5.

$$CF(X \Rightarrow Y) = \begin{cases} \frac{Conf(X \Rightarrow Y) - Sup(Y)}{1 - Sup(Y)} & if\ Conf(X \Rightarrow Y) > Sup(Y) \\ \frac{Conf(X \Rightarrow Y) - Sup(Y)}{Sup(Y)} & if\ Conf(X \Rightarrow Y) < Sup(Y) \\ 0 & otherwise \end{cases} \tag{5}$$

Negative values of Certainty factor mean negative dependence, while positive values mean positive dependence and 0 means independence. However, the value that Certainty factor takes depends on the Support of the consequent (the class in our case). When $Conf(X \Rightarrow Y)$ is close to $Sup(Y)$, even if the difference of $Conf(X \Rightarrow Y)$ and $Sup(Y)$ is close to 0 but still positive, the Certainty factor measure shows a strong positive dependence when $Sup(Y)$ is high (close to 1).

In [7], the authors introduced a measure to estimate the strength of an association rule, called Netconf. This measure, defined in equation 6, has among its main advantages that it detects misleading rules produced by the Confidence.

$$Netconf(X \Rightarrow Y) = \frac{Sup(X \Rightarrow Y) - Sup(X)Sup(Y)}{Sup(X)(1 - Sup(X))} \quad (6)$$

As a simple example, suppose that $Sup(X) = 0.4$, $Sup(Y) = 0.8$ and $Sup(X \Rightarrow Y) = 0.3$, therefore $Sup(\neg X) = 1 - Sup(X) = 0.6$ and $Sup(\neg X \Rightarrow Y) = Sup(Y) - Sup(X \Rightarrow Y) = 0.5$. If we compute $Conf(X \Rightarrow Y)$ we obtain 0.75 (a high Confidence value) but $Y$ occurs in 80 % of the transactions, therefore the rule $X \Rightarrow Y$ does worse than just randomly guessing, clearly, $X \Rightarrow Y$ is a misleading rule [4]. For this example, $Netconf(X \Rightarrow Y) = -0.083$ showing a negative dependence between the antecedent and the consequent.

## 4 Experimental Results

In this section, we present the result of combining the Netconf measure and the Dynamic$-K$ satisfaction mechanism into a CAR-based classifier. These results are compared with those obtained by the other three satisfaction mechanisms: "Best Rule" [2], "Best $K$ Rules" [14] and "All Rules" [3]. Additionally, we show the result of combining different measures and the four satisfaction mechanisms.

For the experiment showed in Table 1, the four satisfaction mechanisms were implemented inside the CAR-IC classifier [18], using the Netconf threshold set to 0.5, as it was reported in other works [20]. All our experiments were done using ten-fold cross-validation reporting the average over the ten folds, the same folds were used for all satisfaction mechanisms. All the tests were performed on a PC with an Intel Core 2 Duo at 1.86 GHz CPU with 1 GB DDR2 RAM. Similar to other works [2,3,8,13,20], we used several datasets, specifically 20. The chosen datasets were originally taken from the UCI Machine Learning Repository [10], and their numerical attributes were discretized by Frans Coenen using the LUCS-KDD [5] discretized/normalized CARM Data Library.

For the experiment showed in Table 2, we used the Confidence threshold set to 0.5, the Certainty Factor threshold set to 0 and for Lift and Conviction, the threshold set to 1, as their authors suggested. It is important to highlight that both Lift and Conviction are not bounded range [4], therefore differences among its values are not meaningful. Therefore, the authors suggest to use for these measures, the threshold set to 1; values greater than 1 mean positive dependence between antecedent and consequent. In case of Certainty Factor, positive dependence is obtained for values greater than 0.

In Table 1, we can see that the combination of Dynamic$-K$ mechanism and Netconf measure yields an average accuracy higher than the combination of Netconf and all other reported mechanisms, having a difference of 2.4 % with respect to the mechanism in the second place ("Best $K$ Rules" with $K$ set to 5, the same value used in other works [6,11,12,14]). Additionally, Dynamic$-K$ wins in 19 of the 20 datasets and ties in the other one.

From the results show in Table 2, we can conclude that the Dynamic$-K$ mechanism obtains the best results independent of the quality measure used to compute the set of CARs, being Netconf the best of all evaluated measures.

**Table 1.** Classification accuracy using Netconf and the different mechanisms.

| Dataset | Best rule | All rules | Best $K$ rules | Dynamic $K$ |
|---|---|---|---|---|
| adult | 83.17 | 82.15 | 84.50 | **87.33** |
| anneal | 92.74 | 91.89 | 95.38 | **96.42** |
| breast | 85.48 | 84.58 | 85.43 | **87.65** |
| connect4 | 56.95 | 55.95 | 62.18 | **67.09** |
| dermatology | 79.48 | 78.28 | 79.66 | **80.39** |
| ecoli | 83.01 | 81.40 | 84.01 | **86.92** |
| flare | 87.03 | 86.44 | 86.45 | **88.58** |
| glass | 69.07 | 68.23 | 68.92 | **72.13** |
| heart | 54.26 | 53.20 | 57.34 | **61.92** |
| hepatitis | 85.51 | 84.60 | 87.02 | **87.60** |
| horseColic | 83.51 | 82.81 | 83.56 | **86.41** |
| ionosphere | 85.03 | 83.96 | 86.02 | **86.93** |
| iris | 97.10 | 97.04 | 96.67 | **97.72** |
| led7 | 73.67 | 72.37 | 75.88 | **78.18** |
| letRecog | 74.20 | 73.56 | 73.42 | **75.70** |
| mushroom | 99.48 | 98.80 | **99.52** | **99.52** |
| pageBlocks | 92.88 | 92.19 | 94.93 | **97.81** |
| penDigits | 78.80 | 77.36 | 78.32 | **84.03** |
| pima | 76.38 | 75.65 | 78.53 | **79.67** |
| waveform | 74.11 | 73.18 | 75.22 | **79.07** |
| Average | 80.59 | 79.68 | 81.65 | **84.05** |

**Table 2.** Average accuracy of different quality measures over the tested datasets, for different satisfaction mechanisms.

| Measure | Best rule | All rules | Best $K$ rules | Dynamic$-K$ |
|---|---|---|---|---|
| Certainty Factor | 74.28 | 73.39 | 72.08 | 77.68 |
| Lift | 75.49 | 74.64 | 73.21 | 77.88 |
| Conviction | 79.53 | 78.32 | 79.21 | 81.16 |
| Confidence | 79.59 | 78.68 | 80.60 | 81.52 |
| Netconf | **80.59** | **79.68** | **81.65** | **84.05** |

## 5 Conclusions

In this paper, we have proposed to improve the accuracy of CAR-based classifiers by combining Netconf measure and Dynamic$-K$ satisfaction mechanism. From the experimental results, we can conclude that the Dynamic$-K$ satisfaction mechanism obtains the best results independent of the quality measure used to compute the set of CARs, being Netconf the best of all evaluated measures.

## References

1. Clark, P., Boswell, R.: Rule induction with CN2: some recent improvments. In: Proc. of European Working Session on Learning, pp. 151–163 (1991)
2. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proc. of the KDD, pp. 80–86 (1998)
3. Li, W., Han, J., Pei, J.: CMAR: accurate and efficient classification based on multiple class-association rules. In: Proc. of the ICDM, pp. 369–376 (2001)
4. Berzal, F., Blanco, I., Sánchez, D., Vila, M.A.: Measuring the accuracy and interest of association rules: A new framework. Intelligent Data Analysis **6**(3), 221–235 (2002)

5. Coenen, F.: The LUCS-KDD discretised/normalised ARM and CARM Data Library (2003). http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS-KDD-DN
6. Yin, X., Han, J.: CPAR: Classification based on predictive association rules. In: Proc. of the SIAM International Conference on Data Mining, pp. 331–335 (2003)
7. Ahn, K.I., Kim, J.Y.: Efficient Mining of Frequent Itemsets and a Measure of Interest for Association Rule Mining. Information and Knowledge Management **3**(3), 245–257 (2005). Hanoi, Vietnam
8. Wang, J., Karypis G.: HARMONY: Efficiently mining the best rules for classification. In: Proc. of SDM, pp. 205–216 (2005)
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. Asuncion, A., Newman D.J.: UCI Machine Learning Repository (2007). http://www.ics.uci.edu/~mlearn/MLRepository.html
11. Wang, Y.J., Xin, Q., Coenen, F.: A novel rule weighting approach in classification association rule mining. In: International Conference on Data Mining Workshops, pp. 271–276 (2007)
12. Wang, Y.J., Xin, Q., Coenen, F.: A novel rule ordering approach in classification association rule mining. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 339–348. Springer, Heidelberg (2007)
13. Cheng, H., Yan, X., Han, J., Yu, P.S.: Direct discriminative pattern mining for effective classification. In: Proc. of the ICDE, pp. 169–178 (2008)
14. Wang, Y.J., Xin, Q., Coenen, F.: Hybrid Rule Ordering in Classification Association Rule Mining. Trans. MLDM **1**(1), 1–15 (2008)
15. Karabatak, M., Ince, M.C.: An expert system for detection of breast cancer based on association rules and neural network. Expert Syst. Appl. **36**, 3465–3469 (2009)
16. Park, S.H., Reyes, J.A., Gilbert, D.R., Kim, J.W., Kim, S.: Prediction of protein-protein interaction types using association rule based classification. BMC Bioinformatics **10**(1) (2009)
17. Bae, J.K., Kim, J.: Integration of heterogeneous models to predict consumer behavior. Expert Syst. Appl. **37**, 1821–1826 (2010)
18. Hernández, R., Carrasco, J.A., Fco, M.J., Hernández, J.: Classifying using specific rules with high confidence. In: Proc. of the MICAI, pp. 75–80 (2010)
19. Malik, W.A., Unwin, A.: Automated error detection using association rules. Intelligent Data Analysis **15**(5), 749–761 (2011)
20. Hernández, R., Carrasco, J.A., Fco, M.J., Hernández, J.: CAR-NF: A Classifier based on Specific Rules with High Netconf. Intelligent Data Analysis **16**(1), 49–68 (2012)
21. Hernández-León, R.: Dynamic $K$: a novel satisfaction mechanism for CAR-based classifiers. In: Ruiz-Shulcloper, J., Sanniti di Baja, G. (eds.) CIARP 2013, Part I. LNCS, vol. 8258, pp. 141–148. Springer, Heidelberg (2013)