Confirmation rule sets

Dragan Gamberger[1] and Nada Lavrač[2]

[1] Rudjer Bošković Institute, Bijenička 54,10000 Zagreb, Croatia
E-mail: gambi@lelhp1.irb.hr
[2] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: nada.lavrac@ijs.si

Abstract. The concept of confirmation rule sets represents a framework for reliable decision making that combines two principles that are effective for increasing the predictive accuracy: consensus in an ensemble of classifiers and indecisive or probabilistic predictions in cases when reliable decisions are not possible. The confirmation rules concept uses a separate classifier set for every class of the domain. In this decision model different rules can be incorporated: either those obtained by applying one or more inductive learning algorithms or even rules representing human encoded expert domain knowledge. The only conditions for the inclusion of a rule into the confirmation rule set are its high predictive value and relative independence of other rules in the confirmation rule set. This paper introduces the concept of confirmation rule sets, together with an algorithm for selecting relatively independent rules from a set of all acceptable confirmation rules and an algorithm for the systematic construction of a set of confirmation rules.

1 Introduction

The concept of confirmation rule sets represents a framework for reliable decision making that combines two principles that are effective for increasing the predictive accuracy: consensus in an ensemble of classifiers and indecisive or probabilistic predictions in cases when reliable decisions are not possible. It is known that ensembles of classifiers generally demonstrate better results than any of their components [2, 3] and that the accuracy and diversity of the components determine the ensemble performance [10]. In most cases classifiers are combined by voting to form a compound classifier. Different classifiers can be obtained either by the application of different learning algorithms on the same training set or by the same learning algorithm on different training (sub)sets. The later approach is used in the well-known *bagging* and *boosting* approaches that employ redundancy to achieve better classification accuracy [4, 5, 16]. For critical applications the predictive accuracy of compound classifiers can be further increased if, instead of voting, the consensus of classifiers' answers is requested. Regardless of the used combination scheme, the fact that it is difficult or impossible to ensure the independence of the ensemble components has the consequence that high prediction reliability of such classifiers can not be ensured in all situations. Another way for achieving reliable predictions is the systematic construction of redundant rules, whose problem is, however, their algorithm and decision complexity.

Independently, there has been significant effort devoted to the development of different other techniques aimed at improving the quality of decision making. Especially in medical domains, some techniques are aimed at the construction of either very sensitive or very specific rules instead of rules with a high overall predictive accuracy.[1] This is however not a general solution because false positive predictions are as inadequate as false negative predictions in many applications. Consequently, some of the techniques applied in medical problems aim at classifiers with high ROC (Receiver Operating Characteristic) curve area (see e.g., [11]).[2]

An alternative approach to reliable predictor construction is the introduction of indecisive predictions. In this approach, in the case of a two-class problem, three different predictions are possible: class positive, class negative, and prediction not possible. The approach follows the concept of *reliable, probably almost always useful learning* defined in [17]. In [15] it was shown how existing machine learning algorithms can be transformed into the form which enables indecisive predictions. In [15] a simple voting based decision model that includes indecisive predictions based on a set of concept descriptions constructed by the FOCL system has been introduced. The achieved predictive accuracies measured on a few medical domains show that significant accuracy improvements are possible. The substantial disadvantage of the approach are however indecisive answers, whose amount has to be kept as low as possible.

The confirmation rule sets concept presented in this work follows the above paradigm of reliable, probably almost always useful learning, allowing for indecisive predictions. However, it represents a more general, consensus based approach to decision making from a set of rules. Its basic characteristic is that it uses separate rules sets for every target class. In this way it is similar to human decision making processes. In addition, like in association rule learning [1], the presented approach utilizes the minimal support requirement as a rule reliability measure which must be satisfied by every rule in order to be accepted as a confirmation rule and included into the confirmation rule set. The approach is introduced in Section 2. The confirmation rule sets concept does not present a novel rule induction approach but rather a decision model in which different rules can be incorporated: either rules induced by (one or more) learning algorithms or even rules representing human encoded expert domain knowledge. High prediction quality is expected only if, besides the predefined predictive

---

[1] Sensitivity measures the fraction of positive cases that are classified as positive, whereas specificity measures the fraction of negative cases classified as negative. Let TP denote true positives, TN true negatives, FP false positives and FN false negative answers, then $Sensitivity = \frac{TP}{TP+FN}$, and $Specificity = \frac{TN}{TN+FP}$.

[2] A ROC curve indicates a tradeoff between the false alarm rate (1 - *Specificity*, plotted on X-axis) that needs to be minimized, and the detection rate (*Sensitivity*, plotted on Y-axis) that needs to be maximized. An appropriate tradeoff, determined by the expert, can be achieved by applying different algorithms, as well as by different parameter settings of a selected data mining algorithm.

value of every included rule, the whole set is as diverse as possible. A simple and general algorithm for the selection of appropriate confirmation rules is presented in Section 2 together with an exhaustive search procedure which can be used for systematic confirmation rules construction. This algorithm has been used to construct rule sets for the coronary arthery disease domain. Prediction results of induced rule sets, measured for different acceptance levels of the consensus scheme, are presented in Section 3, illustrating properties of the confirmation rule sets concept in a real medical domain.

2 Confirmation rules

In the decision model using confirmation rule sets, every diagnostic/prognostic class is treated separately as the target class for which a separate set of confirmation rules is constructed. The basic property of confirmation rules is that they should cover (satisfy) only examples of the given target class. Additionally, in order to ensure high predictive quality of confirmation rule sets, every rule included into the set must be a reliable target class predictor by itself.

Given that a conclusion of a confirmation rule is a target class assignment, and a condition is a conjunction of simple literals, confirmation rules are similar to if-then rules as induced by the AQ algorithms [14], and to association rules [1]. In the confirmation rule concept, however, every *complex* (conjunction) of an AQ rule would constitute a separate and independent rule. Moreover, the main difference with association rules is that confirmation rules have only the target class assignment in the conclusion of a rule whereas a conclusion of an association rule is a conjunction of arbitrary attribute values.

2.1 Properties of confirmation rules

To summarize, confirmation rules are defined by the following properties:

a) The condition of a confirmation rule has the form of a conjunction of simple literals each being a logical attribute value test. The conclusion of a confirmation rule is a target class assignment.
b) Confirmation rules should cover (satisfy) only examples of their target class; since rules cover only target class examples, prediction quality can be estimated by the number of covered target class examples.
c) Acceptable confirmation rules which may be included into confirmation sets are only those rules that cover a sufficient number of target class examples; the *minimal support level* parameter can be used to define the requested minimal number of covered target class examples.[3]

The number of rules in a confirmation rule set is generally not determined and does not have to be equal for all classes. The sets can include all, or only

[3] In our system, the default value of the support level is equal to the second root of the total number of target class examples available.

subset of all, acceptable confirmation rules for the target class. There can be one rule in the set, many of them, but it is also possible that the set is empty if no acceptable confirmation rule is known for the class.

In the defined concept it does not matter how confirmation rule have been induced but rather how they cover the problem space. Every confirmation rule may be induced and used independently of other confirmation rules. When confirmation rule sets are used for prediction, the following outcomes are possible:

a) If no confirmation rule fires for the example, class prediction is indecisive (the example is not classified).[4]
b) If a single confirmation rule fires for the example, class prediction is determined by this rule.
c) If two or more confirmation rules from the same set fire for the example, the target class of the set is predicted with increased reliability.
d) If two or more confirmation rules fire for the example and at least two of these rules are from different sets, class prediction is indecisive.

This indicates that the confirmation rule sets do not give decisive predictions in every situation (cases (a) and (d)), and that predictions of increased reliability are made possible (case (c)). In some situations one may decide to accept only predictions of increased reliability as decisive predictions. How many confirmation rules must cover an example in order to make the decisive classification can be determined by the so-called *acceptance level* parameter. Acceptance level 1 denotes that a single rule coverage is sufficient for example classification (case (b)). Typical values for the acceptance level parameter are 1–3.

2.2 Confirmation rule subset selection algorithm

Typically, there are many acceptable confirmation rules for every class, satisfying the requested minimal number of covered target class examples (defined by the *minimal support level* parameter). Inclusion of all these rules into the rule set is generally not desired because (a) it is difficult to make decisions based on very large sets of rules, and (b) experiments demonstrated that there are subsets of very similar rules which use almost the same attribute values and have similar prediction properties. The second characteristic is especially undesirable in cases when the *acceptance level* greater than one is used, intended at increasing the prediction reliability because in this case very similar rules will cover an example more than once. A solution to this problem is to reduce confirmation rule sets so that they include only a relatively small number of confirmation rules which are as diverse as possible.

It must be noted that a simple increase of the required support level can reduce the number of acceptable confirmation rules, however probably the remaining rules will still be from the same subset of similar rules. This fact was

[4] Alternatively, a probabilistic classification could be proposed for this case, e.g., using a simple Bayesian classifier scheme. In our system this approach has not yet been implemented and tested.

experimentally detected. A better solution is to leave the required support level unchanged so that there are many acceptable confirmation rules and then select among them a small subset of relatively independent rules.

Selecting a subset of independent classifiers for the same target class is known as a complex task which occurs in most multiclassifier decision systems [9]. The problem is difficult because there are many combinations which can make different rules statistically dependent, including (even for domain expert unknown) relations among attribute values reflecting inherent domain properties.

The confirmation rule sets concept is intended to be able to include and combine all the available knowledge without other restrictions than the requested prediction quality of individual confirmation rules. Consequently, our approach accepts as diverse those rules that cover as different sets of target class examples as possible. Obviously the approach can not guarantee statistical independence of the selected rules, but its advantages are the simplicity and robustness concerning all the known and unknown dependences among attribute values.

Algorithm 1: CONFIRMATION RULE SUBSET SELECTION

Input: A set of all acceptable confirmation rules for the target class
P target class examples
Parameter: *number* (number of rules in the selected subset)
Output: subset of *number* relative independent confirmation rules
for the target class
(1) **for** every $e \in P$ **do** $c(e) \leftarrow 1$
(2) **repeat** *number* times
(3) **select** from A the rule with
the highest weight $\sum 1/c(e)$ where summation is over the set
$P' \subseteq P$ of target class examples covered by the rule
(4) **add** the selected rule into the output confirmation rule set
(5) **for** every $e \in P'$ of the selected rule **do** $c(e) \leftarrow c(e) + 1$
(6) **eliminate** the selected rule from A
(7) **end repeat**
(8) **exit** with *number* of selected confirmation rules

Algorithm 1 presents an approach to selecting the subset of a *number* of relative independent confirmation rules. Input is the set of all acceptable confirmation rules A and the set of all target class examples P. For every example $e \in P$ there is a counter $c(e)$. Initially the output set of selected rules is empty and all counter values are set to 1 (step 1). After that in each iteration of the loop (steps 2 to 7) one confirmation rule is added into the output set (step 4). From set A the rule with the highest *weight* value is selected.

For each rule, *weight* is computed such that $1/c(e)$ values are added for all target class examples covered by this rule (step 3). After rule selection, the rule is eliminated from the set A (step 6) and $c(e)$ values for all target class examples covered by the selected rule are incremented by 1 (step 5). This is the central part of the algorithm which ensures that in the first iteration all target class examples contribute the same value $1/c(e) = 1$ to the *weight*, while in following iterations the contributions of examples are inverse proportional to their coverage by previously selected rules. In this way the examples already covered by one or more selected rules can contribute substantially less to the *weight* and the rules covering many yet uncovered target class examples have a greater chance to be selected in the following iterations.

For noisy domains the condition that confirmation rules should not cover any of the non-target class examples may be too strong since it may result in a small total number of target class examples covered by every possible confirmation rule. The requirement may be relaxed by accepting also confirmation rules covering a few non-target class examples as well. Such a modification offers a simple and practical noise handling approach, but it may lead to the reduction of predictive accuracy of induced confirmation rules. Instead of such noise handling, experiments presented in Section 3 are done using a procedure for explicit noise detection and elimination [7]. This procedure is based on the consensus of saturation filters, performing reliable filtering of noisy examples in preprocessing. The characteristic of this approach is that only a small number of examples with high probability of actually being noisy are detected and eliminated from the training set before confirmation rule induction. This is important for the confirmation rules concept which should provide for a high reliability of decisive predictions.

2.3 Confirmation rule set construction

Algorithm 1 builds the set of relative independent confirmation rules by selecting the rules from the input set A, consisting of all acceptable confirmation rules. In some cases there are neither expert knowledge nor rules generated by inductive learning systems that can be used as acceptable confirmation rules. In such situation Algorithm 1 can be modified so that instead of its steps 3 and 6, Procedure 1 is used in every iteration to construct a confirmation rule with highest weight $\sum 1/c(e)$. This procedure is actually an exhaustive search algorithm which uses a set of literals defined for the domain. Such set can be constructed and potentially optimized by algorithms described in [12]. The procedure builds the confirmation rule in the form of logical conjunction of literals so that: **a)** the rule does not cover examples of the non-target class, **b)** the rule covers more than *minimal support level* of target class examples, and **c)** the rule has maximal possible weight $\sum 1/c(e)$. Algorithm 1 with included Procedure 1 is used in experiments presented in Section 3. Computational complexity of the exhaustive search in Procedure 1 is hight what restricts its applicability to problems of with up to few hundred examples.

Procedure 1 needs as its inputs the complete training set E, the appropriate literal set L, and $c(e)$ values imported from Algorithm 1 for all positive training examples. The procedure additionally requires that the parameter $min\_support$ is defined which restricts the space of acceptable confirmation rules. In case when internal variable $best\_weight$ is greater than zero then procedure output is the best acceptable confirmation rule that could be constructed with available literals. If the procedure could not find any acceptable solution then $best\_weight = 0$.

Procedure 1: CONFIRMATION RULE CONSTRUCTION

Input: $E = P \cup N$ (E training set, P positive or target class examples,
N negative or non-target class examples).
L (set of literals, $l \in L$ covers a positive example $e \in P$ if l is true for e,
l covers a negative example $e \in N$ if l is false for e).
$c(e)$ values imported from Algorithm 1.

Parameter: *min_support* (minimal support level for rule acceptance with
default value equal to the second root of $|P|$)

Output: selected confirmation rule in best solution if *best_weight* > 0

(1) **set** *best_weight* $= 0$, loop level pointer $V = 0$
(2) **repeat** literal selection loop
(3) **if** a literal from the literal set at level V covering an uncovered negative example exists **then**
(4) **include** this literal into the present solution at level V
(5) **compute** coverage of positive and negative examples at level V
(6) **compute** *weight* $= \sum 1/c(e)$, $e \in P$ and covered at level V
(7) **if** *weight* $\leq$ *best_weight* or total number of covered positive examples $<$ *min_support* **then** forget this literal at level V and continue the loop at level V
(8) **if** all negative examples are covered at level V **then** copy the present solution into the best solution, *best_weight* $=$ *weight*, forget this literal at level V, and continue the loop at level V
(9) **continue** the loop at level $V + 1$
(10) **else**
(11) **if** level $V = 0$ **then** exit the loop
(12) **else** continue the loop at level $V = V - 1$
(13) **end repeat**

3 Summary of confirmation rule sets application on a medical domain

Application characteristics of confirmation rule sets are illustrated in this section with a summary of measured prediction results for the coronary artery disease diagnosis dataset, collected at the University Medical Center, Ljubljana, Slovenia. Details about the domain and some other machine learning results can be found in [8, 6]. Independently, in [7] the same domain was used to test our noise handling algorithm (the so-called consensus saturation filter). The results were good because the system detected in total 15 noisy examples (out of 327 patient records) out of which the medical doctor who collected the data recognized 14 as being real outliers, either being errors or possibly noisy examples with coronary angiography tests very close to the borderline between the two classes.

In accordance with the standard 10-fold cross-validation procedure, for every training set, 5 confirmation rules were generated for the class *positive* and 5 for the class *negative*. Such experimental setting enabled the testing of generated confirmation rules sets with different acceptance levels. The prediction is correct

accept. level	correct predictions	measured error rate	meas. relative error rate	real relative error rate
a) without noise elimination in preprocessing				
1	72.48%	7.65%	9.54%	4.2%
2	47.71%	2.75%	5.45%	1.8%
3	28.44%	1.22%	4.12%	1.0%
b) with noise elimination in preprocessing				
1	76.15%	5.81%	7.09%	3.2%
2	60.86%	3.06%	4.78%	2.0%
3	47.40%	1.83%	3.73%	0.6%

Table 1. Results of 10-fold cross-validation presenting the percentage of correct predictions, measured error rate, measured relative error rate and real relative error rate for **a)** without and **b)** with noise elimination in preprocessing. For each fold with about 294 training examples and 33 test examples, 5 **confirmation rules** for the class *positive* and 5 **confirmation rules** for the class *negative* were generated. Results are presented for acceptance levels 1–3, where level 3 means that the example must satisfy at least 3 out of 5 rules for decisive prediction. The percentage of correct predictions represents the total number of correct predictions divided by the total number of predictions (327), while the measured error rate is the total number of erroneous predictions divided by the total number of all predictions. The measured relative error rate is equal to the ratio of the number of erroneous predictions and the number of decisive predictions. The real relative error rate is computed so that the number of erroneous predictions is, at first, reduced so that it does not include expert-evaluated domain outliers, and then it is divided by the number of decisive predictions.

(successful) if the example is classified into a single class, which has to be the same as the expert classification. The prediction is erroneous if the example is classified into a single class which is different from the expert classification.

Experimental results are presented in Table 1. The table has two parts: the first presents results obtained *without* and the second *with* noise elimination in preprocessing. In both cases results for three different acceptance levels are reported. The first column of every row is the acceptance level, follows the percentage of correct predictions and the percentage of erroneous predictions. Difference between the numbers is the percentage of indecisive predictions in the corresponding experiment. In the fourth column is the measured relative error computed as the ratio of the number of erroneous predictions and the number of decisive predictions. The numbers in this column are greater than those in the third column because decisive predictions are only a part of all predictions. In this sense values in column four are more realistic from the users point of view. But values in column four (and column three) include noisy cases already detected and evaluated by domain expert in [7]. Misprediction of these cases are not actual errors but expected result of good rules. In order to estimate the real relative error rate, such cases (14 of them for the domain) were eliminated from the measured error sets in which they occur and then the real relative error rate was computed. The values are presented in the last column of Table 1.

Measured error rates in this domain are between 3.7% and 9.5% (column 4) while estimated real error rates are about 0.6% – 4.2% (column 5) what are better results than those obtained both by other machine learning algorithms and medical experts [8]. It must be noted that the elimination of the 'expected' domain noise was extremely conservative, based on the consensus of the saturation filter preprocessor and the domain expert, potentially resulting in overestimation of the real error rate. The least estimated real error rate is detected with acceptance level 3 and noise detection in preprocessing. In this case the number of indecisive predictions is about 50% with only one really wrong prediction in about 150 decisive classifications. This result proves high reliability of the induced confirmation rules.

Results in Table 1 demonstrate the differences in prediction quality of various acceptance levels. As expected, the increased acceptance level reduces the number of correct predictions but it also significantly reduces the number of erroneous predictions, especially the real predictive errors. The observation holds with and without noise elimination in preprocessing. Noise elimination itself is very useful. The comparison of the number of correct predictions for confirmation rules generated without and with noise detection and elimination in preprocessing demonstrates the importance of the use of this (or a similar) noise handling mechanism for effective confirmation rule induction. For example, for acceptance level 3 the increase is from 28% to 47%.

4 Conclusion

This work stresses the importance of reliable decision making and for this purpose the paper elaborates the concept of confirmation rule sets. It is shown that in critical applications where decision errors need to be minimized, confirmation rule sets provide a simple, useful and reliable decision model.

The proposed confirmation rule sets framework is general because it enables the incorporation of results of different machine learning algorithms, as well as the existing expert knowledge. The induced structure of an unordered list of simple rules and the possibility of providing predictions of increased reliability are its main advantages. The main disadvantage of the approach are indecisive answers. In presented experiments the number of indecisive predictions has been high, always greater than 20% with a maximum greater than 70%. In the case of indecisive predictions, a probabilistic classification could be proposed, e.g., using a simple Bayesian classifier scheme. In our system this approach has not yet been implemented and tested. This is planned in further work.

Acknowledgement

This work has been supported in part by Croatian Ministry of Science and Technology, Slovenian Ministry of Science and Technology, and the EU funded project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprize (IST-1999-11495).

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A.I. Verkamo (1996) Fast discovery of association rules. In U.M. Fayyad, G. Piatetski-Shapiro, P. Smyth and R. Uthurusamy (Eds.) *Advances in Knowledge Discovery and Data Mining*, pp. 307–328. AAAI Press.
2. K.M. Ali and M.J. Pazzani (1996) Error reduction through learning multiple descriptions. *Machine Learning*, 24:173–206.
3. P.J.Boland (1989) Majority systems and the Concorcet jury theorem. *Statistician*, 38:181–189.
4. L. Breiman (1996) Bagging predictors. *Machine Learning* 24(2): 123–140.
5. Y. Freund and R.E. Shapire (1996) Experiments with a new boosting algorithm. In *Proc. Thirteenth International Machine Learning Conference ICML'96*, 148–156, Morgan Kaufmann.
6. D. Gamberger, N. Lavrač, C. Grošelj (1999) Diagnostic rules of increased reliability for critical medical applications. In *Proc. Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making AIMDM-99*, 361–365, Springer Lecture Notes in AI 1620.
7. D. Gamberger, N. Lavrač, C. Grošelj (1999) Experiments with noise filtering in a medical domain. In *Proc. of International Conference on Machine Learning ICML-99*, 143–151. Morgan Kaufmann.
8. C. Grošelj, M. Kukar, J.J. Fetich and I. Kononenko (1997) Machine learning improves the accuracy of coronary artery disease diagnostic methods. *Computers in Cardiology*, 24:57–60.
9. T.Kagan and J. Ghosh (1996) Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8:385–404.
10. J. Kittler, M. Hatef, R.P.W. Duin, and J.Matas (1998) On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20:226–239.
11. M. Kukar, I. Kononenko, C. Grošelj, K. Kralj, J.J. Fettich (1998) Analysing and improving the diagnosis of ischaemic heart disease with machine learning. *Artificial Intelligence in Medicine*, special issue on *Data Mining Techniques and Applications in Medicine*, Elsevier.
12. N. Lavrač, D.Gamberger, and P. Turney (1997) A relevancy filter for constructive induction. *IEEE Intelligent Systems & Their Applications*, 13:50–56.
13. B. Liu, W. Hsu and Y. Ma (1998) Integrating classification and association rule mining. In *Proc. Fourth International Conference on knowledge Discovery and Data Mining*, KDD-98, New York, USA, 1998.
14. R.S. Michalski, I. Mozetič, J. Hong, and N. Lavrač (1986) The multi-purpose incremental learning system AQ15 and its testing application on three medical domains. In *Proc. Fifth National Conference on Artificial Intelligence*, pp. 1041–1045, Morgan Kaufmann.
15. M. Pazzani, P. Murphy, K. Ali, and D. Schulenburg (1994) Trading off coverage for accuracy in forecasts: Applications to clinical data analysis. In *Proceedings of the AAAI Symposium on AI in Medicine*, pp. 106–110.
16. J.R. Quinlan (1996) Boosting, bagging, and C4.5 . In *Proc. Thirteenth National Conference on Artificial Intelligence*, 725–730, AAAI Press.
17. R.L. Rivest and R. Sloan (1988) Learning complicated concepts reliably and usefully. In *Proc. Workshop on Computational Learning Theory*, 69–79, Morgan Kaufman.