Enhancing Rule Interestingness for Neuro-fuzzy Systems

Thomas Wittmann, Johannes Ruhland, and Matthias Eichholz

Lehrstuhl für Wirtschaftsinformatik,
Friedrich Schiller Universität Jena
Carl-Zeiß-Str. 3, 07743 Jena, Germany
t.wittmann@wiwi.uni-jena.de, j.ruhland@wiwi.uni-jena.de, w6eima@wiwi.uni-jena.de

Abstract. Data Mining Algorithms extract patterns from large amounts of data. But these patterns will yield knowledge only if they are interesting, i.e. valid, new, potentially useful, and understandable. Unfortunately, during pattern search most Data Mining Algorithms focus on validity only, which also holds true for Neuro-Fuzzy Systems. In this Paper we introduce a method to enhance the interestingness of a rule base as a whole. In the first step, we aggregate the rule base through amalgamation of adjacent rules and eliminiation of redundant attributes. Supplementing this rather technical approach, we next sort rules with regard to their performance, as measured by their evidence. Finally, we compute reduced evidences, which penalize rules that are very similar to rules with a higher evidence. Rules sorted on reduced evidence are fed into an integrated rulebrowser, to allow for manual rule selection according to personal and situation-dependent preference. This method was applied successfully to two real-life classification problems, the target group selection for a retail bank, and fault diagnosis for a large car manufacturer. Explicit reference is taken to the NEFCLASS algorithm, but the procedure is easily generalized to other systems.

1 Introduction

Data Mining Algorithms extract patterns from large amounts of data. But patterns are only interesting, if they are valid, new, potentially useful, and understandable. Unfortunately, most Data Mining Algorithms only refer to validity in search for patterns. This also holds for Neuro-Fuzzy Systems, a promising new development for classification learning. Empirical studies (e.g. [11]) have proven their ability to combine automatic learning, as attributed to neural networks, with classification quality comparable to other data mining methods. But when it comes to analyzing real-life problems the main advantage over pure neural networks, the ease of understandability and interpretation, remains a much-acclaimed desire rather than a proven fact, though.

Hence, pattern post-processing is necessary to identify interesting rules in the rule base output of a Neuro-Fuzzy System. This can also be called „data mining of second order“ [6]. Different methods can be used to enhance interestingness of rules

J.M. Zytkow and J. Rauch (Eds.): PKDD'99, LNAI 1704, pp. 242-250, 1999.

according to the four criteria mentioned above. The aim is to concentrate on the most valid, new and useful rules and thus aggregate the rule base. Finally, a lean rule base is easier to understand.

In this paper we refer especially to the NEFCLASS algorithm, but the procedure we develop can be generalized easily. In the first step, we aggregate the rule base with respect to adjacent rules and redundant attributes, in the second step we order the rules with regard to their performance. These sorted rules are fed into an integrated rulebrowser, which gives the user the opportunity to select rules according to his/her interests. As a first test, we successfully applied this method to two real-life classification problems, the target group selection for a bank, and the fault diagnosis for a big car manufacturer.

2 The Neuro-fuzzy System NEFCLASS

In *NEFCLASS (NEuroFuzzyCLASSification)* a fuzzy system is mapped on a neural network, a feed-forward three-layered multilayer perceptron. The (crisp) input pattern is presented to the neurons of the first layer. The fuzzification takes place when the input signals are propagated to the hidden layer, because the weights of the connections are modeled as membership functions of linguistic terms. The neurons of the hidden layer represent the rules. They are fully connected to the input layer with connection weights being interpretable as fuzzy sets. A hidden neuron's response is connected to one output neuron only. With output neurons being associated with an output class on a 1:1 basis, each hidden neuron serves as a classification detector for exactly one output class. Hidden to output layer connections do not carry connection weights reflecting the lack of rule confidences in the approach. The learning phase is divided into two steps. In the first step the system learns the number of rule units and their connections, i.e. the rules and their antecedents, and in the second step it learns the optimal fuzzy sets, that is their membership functions [9]. NEFCLASS is available as freeware from the Institute of Knowledge Processing and Language Engineering, University of Magdeburg, Germany, http://fuzzy.cs.uni-magdeburg.de/welcome.html.

3 Rule Aggregation

Next, we describe a prototype for rule post-processing, called *Rulebrowser.* In the first step, *Rulebrowser* aggregates the rule base with respect to redundant rules and attributes. It joins adjacent rules, i.e. those containing the same conclusion while being based on adjacent fuzzy sets, and eliminates attributes becoming irrelevant afterwards (i.e. all fuzzy sets of variable joined). Figure 1 shows the basic idea for a simple case with two attributes (each containing three fuzzy sets) and nine rules classifying the cases into two classes (class1: c1 and class 2: c2). In figure 1 we are able to join the rules r_1 to r_3, thus making attribute x_1 superfluous for this rule r_a. Furthermore, e.g. r_5 and r_8 can be joined, but this will not allow us to eliminate any attribute.

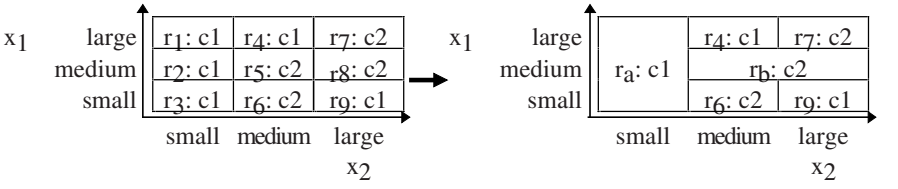

Fig. 1. Rule Aggregation. Example

As seen from figure 1 joining rules in one dimension can inhibit rule aggregation along another attribute. Hence, when confronted with a high-dimensional rule space, rule aggregation is no trivial task, but a complex search problem [7]. Different algorithms can be used to search for suitable candidate rules for aggregation. *Rulebrowser* resorts on a similarity measure to be defined below. For each rule we determine the most similar rule and check the position of both rules in rule space. If we can join them, the first rule's premise is extended to the range of the second rule, eliminating the second rule. If premises can not be joined, we examine the next similar rules, until we find two adjacent rules or all rules are examined.

An attribute becomes superfluous, if after rule aggregation the antecedent containing this attribute covers all possible terms of this attribute, e.g. r_a: 'x1 = small or medium or large' in figure 1. In pseudo-code notation this is :

```
program Rule Aggregation
determine similarities between rules
for all classes do begin
  for all rules in a class do begin
    eliminate superfluous attributes in rules
    repeat for all rules in a class in order of
    similarity to the rule in focus
      if both rules can be joined then
        extend first rule's premise
        eliminate second rule
    until adjacent rule found
  end
end
```

The similarity of two rules $sim_{r1,r2}$ has been defined in [3] as

$$sim_{r1,r2} = 1 - \frac{\sum_{i=1}^{n} dist_i}{n} \quad . \tag{1}$$

$r_1, r_2 =$ compared rules,
$n =$ total number of attributes in r_1 and r_2,
$dist_i =$ distance measure for attribute i:

$$dist_i = \begin{cases} |x_{r1}, x_{r2}| & \text{if attribute i exists in both rules} \\ 1 & \text{else} \end{cases},$$

$|x_{r1}, x_{r2}|$ = distance between the fuzzy sets in r_1 and r_2, measured by their rank, i.e. for an attribute with 5 fuzzy sets {very small, small, medium, large, very large} |large, very small| = 4-1 = 3. For compound fuzzy sets like 'low or medium' the mean of the ranks is used for computation.

4 Rule Sorting and Browsing

Rule aggregation is a more or less technical approach, aiming only at understandability (lean rule base) and utility of rules (relevant attributes). In the second step, *Rulebrowser* sorts the rules with regard to a user-defined performance criterion and gives the user the opportunity to select the rules that comply with his/her interests. According to the user's data mining target different foci on the same rule base may be realized by such a system. Possible aims of analysis are, for example:

1. In database marketing the user looks for a specified number of addresses, which have a high potential for becoming customers. The goal is a number of rules with a high coverage, ordered by decreasing validity. Not a single justification of a decision is important but reliable selection of a high number of potential customers.
2. In checking creditworthiness the user searches for rules with a high validity, but not necessarily a high coverage, that tells him whether the applicant is a high or a low risk customer.

This is to illustrate our belief that there is no single 'most interesting' rule base, but that the user's interests are the prime criteria for interestingness [1][2]. In detail, this approach takes into consideration the single rule confidence, as well as extended utility and novelty aspects.

4.1 Selecting High-Performance Rules

First, we determine the high-performance rules and sort the rules according to their power. This will be operationalized in various ways. Besides the well-known criteria, rule confidence and rule support, rule evidence is a relevant feature.

- A rule's support is measured by the number of patterns that accomplish the rule's premise. In other words, it is the number of times the rule 'fires'. For this we emulate the signal propagation within NEFCLASS. Rules with compound antecedents have to be split up into 'pure' rules combined by a disjunction.
- The proportion of patterns, in which classification by a rule is done correctly, determines its confidence.
- Evidence is a composite measurement, depending on rule confidence and rule support. The following equation is based on suggestions of Gebhardt [3] and the 'average error', proposed by Jim/ Wuthrich [5]:

$$\text{evidence}_i = \begin{cases} 1 - \left(\frac{a+b}{n}\right) & \text{if support } > 0 \\ 0 & \text{else} \end{cases} . \quad (2)$$

a = recognized patterns of the wrong class,
b = not recognized patterns of the right class and
n = number of patterns.

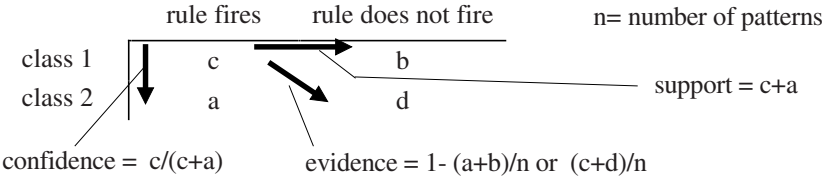

Fig. 2. Interrelation between the concepts of confidence, support and evidence.

Figure 2 visualizes the interrelation between confidence, support and evidence for a simple example with two classes. The rule's conclusion is class 1. While confidence measures how valid, and support measures how often a rule 'fires', evidence makes a more complex proposition. It penalizes rules that misfire (rule fires, although it should not) as well as rules that fire too infrequently (rule does not fire, although it should).

Rulebrowser sorts the rules based on the chosen performance measure (evidence is the default option, but confidence and support can be chosen). In each iteration these measures are calculated only for the patterns not already covered by rules that have entered the rulebase in previous steps. Hence, just the incremental value of the rule to the rulebase is taken into account. This is based on the idea of the 'rule cover'-algorithm by Toivonen [13]. The user defines a mimimum support of the rules as a stop-criterion (if this support is not reached, the algorithm stops after 100 iteration): This leads to a selection of the most relevant rules. Defining a high support threshold will often shrink a rulebase drastically, but must be weighted against ensuing information loss.

```
program Rule Sort
repeat
  determine performance for each rule in search space
  for all classes do begin
    select rule with maximum evidence and support > 0
    delete all cases covered by this rule
    delete rule from search space
  end
until stop-criterion fulfilled
```

4.2 Devaluation of Similar Rules

To this point, we have only judged rules based on their performance considered in isolation. We may now manipulate the evidence formula to account for novelty of

entering candidate to the rulebase. We compute reduced evidences, which penalize rules that are very similar to rules with a higher evidence [3].

```
program Devaluate Similar Rules
for all rules do begin
  reduced evidence := evidence
  rule := not marked
end
repeat
  for all classes do begin
    determine not-marked rule with highest reduced
    evidence and mark it
    for all not-marked rules do begin
      compute new reduced evidence according to (3)
    end
  end
until all rules are marked
```

$$V^{\text{new red}}(R_1) = \min \left\{ V^{\text{red}}(R_1),\ V(R_1) * \left[\frac{V(R_1)}{V^{\text{red}}(R_2)} \right]^{\delta * \text{sim}_{R1,R2}{}^{\kappa}} \right\}. \qquad (3)$$

$V^{\text{new red}}(R_i)$ = new reduced evidence of rule i,
$V^{\text{red}}(R_i)$ = reduced evidence of rule i,
$V(R_i)$ = evidence of rule i,
δ = strength of devaluation,
$\text{sim}_{R1,R2}$ = similarity of rule 1 and 2 according to formula (1),
κ = relevance of similarity.

This reduced evidence is again used to sort the rules [3]. In the resulting, rather user-friendly overview of the rules, the user can determine the parameters of devaluation by a scroll-bar, re- and devaluate rules manually and cut off rules with low reduced evidence. In addition he/she can change the sort criterion. In doing so the user can find his/her optimal position in the trade-off between the different facets of interestingness, especially validity and simplicity of the rule base.

5 Empirical Evaluation of *Rulebrowser*

Rulebrowser has been successfully applied to two real-life classification problems: fault diagnosis for a large car manufacturer and the target group selection for a bank. The first database (*car data*) contains 18.000 records on cars recently sold, their respective characteristics (21 standard equipment characteristics such as number of cylinders plus 221 on optional equipment as ABS) and data on faults detected (e.g. engine breakdown). The analysis aim was, How do car characteristics influence certain fault frequences? The second database (*bank data*) is based on a mailing campaign to

convince customers of a bank to buy a credit card. It consists of about 180.000 cases, 21 attributes and 2 classes (respondents/ non-respondents). For the car data, figure 3 shows the considerable reduction in the number of rules and the average number of attributes in one rule after rule aggregation. Figure 4 shows the further reduction in the number of rules for a minimum support of 95%, 90% and 80%.

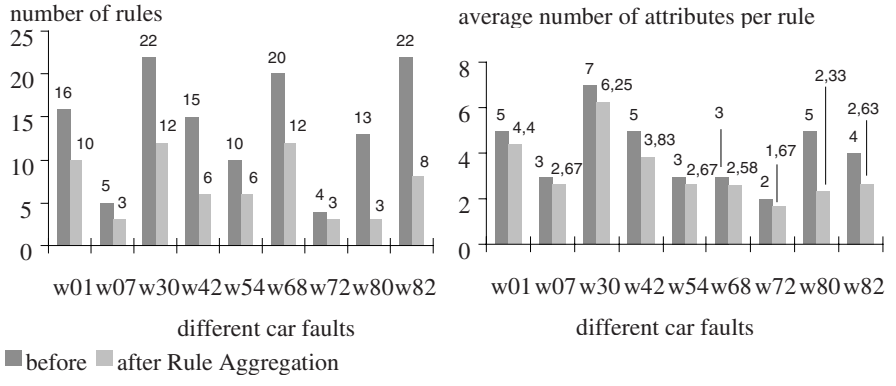

Fig. 3. Reduction of the number of rules and average number of attributes in one rule.

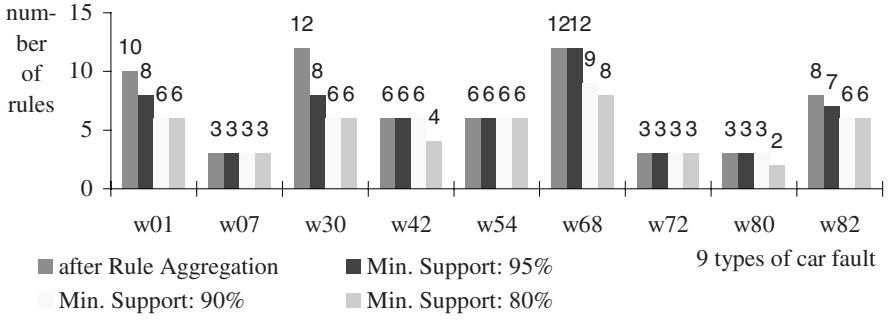

Fig. 4. Further reduction in the number of rules.

Looking at the bank data, which, in contrast to the car data, were only moderately preprocessed with respect to the number of attributes, we can see even better results. Rule aggregation reduced the average number of attributes per rule from 21 to 17.5, with the smallest rule containing only 11 attributes. The number of rules sank from 242 to 48 after rule aggregation. With a minimum support of 90% we even managed to decrease the number of rules to 16.

In both cases, rule aggregation leads to no loss of validity, as no information is deleted. Looking at the minimum support rule, we are confronted with a validity versus simplicity trade-off. But the loss of validity is small due to the elimination of the least powerfull rules, that cover only few cases and describe no new structures.

6 Related work

Pattern postprocessing, in contrast to the various preprocessing tasks, like attribute selection or treating missing values, is a stepchild of research. Only few approaches have been developed to solve the problem of enhancing rule interestingness. Most of them are found in the field of association analysis, where the phenomenon of 'exploding' rule bases is fundamental. Due to space restrictions we can only mention a selection of the approaches

Most methods rely on objective measures of rule interestingness, like the 'Neighborhood-Based Unexpectedness' of Dong/ Li [2], the use of rule-covers by Toivonen [13] or the rule aggregation approach of Major/ Mangano [8]. Only few approaches integrate the user, usually by requiring predefined rule templates (Klemettinen et al. [6]) or belief systems (Silberschatz/ Tuzhilin [12]). But this is often unfeasible in practice, due to the strong involvement of the user. Few authors propose combined approaches for the evaluation of interestingness. Gebhardt suggests evidence and affinity of rules for a composed measure of interestingness. He proposes several ways to quantify them and discusses pros and cons [3]. Hausdorf/ Müller have developed a complex system for evaluating interestingness based on different facets [4]. The developers of NEFCLASS themselves have proposed a rule aggregation method, based on four different steps of attribute, rule and fuzzy set elimination [10]. But these steps mainly aim at the validity of the rules and contain serious problems. Another algorithm by Klose et al. aggregates the rule base in three steps, 'input pruning on data set level' (attribute selection), 'input pruning on rule level' and 'simple rule merging' [7]. The last two steps correspond to the rule aggregation proposed in this paper.

7 Conclusions

Rule post-processing is an essential step in the Knowledge Discovery in Databases process. This holds true for Neuro-Fuzzy Systems in particular, too. In this paper we have proposed a prototype for a tool that aggregates and sorts rules according to different facets of interestingness. The results are promising, but further research is needed. For example, more sophisticated search strategies to identify candidate rules for aggregation might improve the resulting rule base. Visualization techniques, which have not been mentioned in this paper, could enhance the understandability of the rules. However, the quality of a post-processing method is very hard to quantify, as it strongly depends on the system's user. Hence, only further practical applications of our methods can evaluate their effectiveness.

Research in this subject was funded in part by the Thuringian Ministry for Science, Research and Culture. The authors are responsible for the content of this publication.

References

1. Brachman, R. J., Anand, T., The Process of Knowledge Discovery in Databases, In: Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. (Eds.), Advances in Knowledge Discovery and Data Mining, Menlo Park, CA 1996, pp. 37-58
2. Dong, G., Li, J., Interestingness of Discovered Association Rules in terms of Neighborhood-Based Unexpectedness, In: Wu, X., Kotagiri, R., Korb, E. (Eds.), Research and Development in Knowledge Discovery and Data Mining (Proceedings of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining), Heidelberg 1998, pp. 72-86
3. Gebhardt, F., Discovering interesting statements from a database, In: Applied Stochastic Models And Data Analysis 1/1994, pp. 1-14
4. Hausdorf, C. Müller, M., A Theory of Interestingness for Knowledge Discovery in Databases Exemplified in Medicine, In: Lavrac, C. Keravnou, E., Zupan, B. (Eds.), First International Workshop on Intelligent Data Analysis in Medicine and Pharmacology
5. Jim, K., Wuthrich, B., Rule Discovery: Error measures and Conditional Rule Probabilities, In: KDD: Techniques and Applications. Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining, Singapur u.a. 1997, pp. 82-89
6. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H.; Verkamo, A. I., Finding Interesting Rules from Large Sets of Discovered Association Rules, In: Adam, N. R., Bhargava, B. K., Yesha, Y. (Eds.), Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), Gaithersburg, Maryland, November 29 - December 2, 1994, (ACM Press) 1994, pp. 401-407
7. Klose, A., Nürnberger, A., Nauck, D., Some approaches to improve the interpretability of Neuro-Fuzzy Classifiers, In: Zimmermann, H.-J. (Eds.), 6th European Congress on Intelligent Techniques and Soft Computing. Aachen, Germany, September 7-10, 1998, Aachen 1998, pp. 629-633
8. Major, J. A., Mangano, J. J., Selecting among rules induced from a hurricane Database, In: Piatetsky-Shapiro, G. (Eds.), Knowledge Discovery in Databases, Papers from the 1993 AAAI Workshop, Menlo Park, CA 1993, pp. 28-44
9. Nauck, D., Kruse, R., NEFCLASS - A Neuro-Fuzzy Approach for the Classification of Data. Paper of Symposium on Applied Computing 1995 (SAC'95) in Nashville
10. Nauck, D., Kruse. R., New Learning Strategies for NEFCLASS, In: Proc. Seventh International Fuzzy Systems Association World Congress IFSA'97, Vol. IV, Academia Prague, 1997, pp. 50-55
11. Ruhland, J., Wittmann, T., Neurofuzzy Systems In Large Databases - A comparison of Alternative Algorithms for a real-life Classification Problem, in: Proceedings 5th European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, September 8-11 1997 (EUFIT' 97), pp. 1517-1521
12. Siberschatz, A., Tuzhilin, A., On subjective measures of interestingness in knowledge discovery, In: Proc. of the 1st International Conference on Knowledge Discovery and Data Mining, Montreal, August 1995, pp. 275-281
13. Toivonen, H., Klemettinen, M., Ronkainen, P., Hätönen, K., Mannila, H., Pruning and Grouping Discovered Associations Rules, In: Kodratoff, Y., Nakhaeizadeh, G., Taylor, C. (Eds.), Workshop Notes Statistics, Machine Learning, and Knowledge Discovery in Databases. MLNet Familiarization Workshop, Heraklion, Crete, April 1995, 1995, pp. 47-52