Mathware & Soft Computing 11 (2004) 179-195

A Cost-Sensitive Learning Algorithm for Fuzzy Rule-Based Classifiers

S. Beck, R. Mikut, and J. Jäkel
Institute for Applied Computer Science
Forschungszentrum Karlsruhe, Germany
e-mail:{sebastian.beck;jens.jaekel;ralf.mikut}@iai.fzk.de

Abstract

Designing classifiers may follow different goals. Which goal to prefer among others depends on the given cost situation and the class distribution. For example, a classifier designed for best accuracy in terms of misclassifications may fail when the cost of misclassification of one class is much higher than that of the other. This paper presents a decision-theoretic extension to make fuzzy rule generation cost-sensitive. Furthermore, it will be shown how interpretability aspects and the costs of feature acquisition can be accounted for during classifier design. Natural language text is used to explain the generated fuzzy rules and their design process.

1 Motivation

There are many approaches for solving classification problems using fuzzy techniques, e. g. clustering, neuro-fuzzy or tree-oriented approaches [1, 2, 3, 4]. The usually applied criterion is to minimize the average classification error. This approach may not be suitable for problems with both overlapping classes and asymmetric costs of wrong decisions ("decision cost") [5, 6, 7]. If there is no error-free solution for such a problem, trying to reduce misclassifications does not necessarily lead to the solution with the lowest decision cost. This consideration of costs is well established in crisp and fuzzy decision-making [7, 8, 9, 10], but only few approaches include costs in classifier design [11, 12].

Apart from the decision costs, other aspects may influence the design process of a classifier. The cost of the acquisition of data used for classification ("classifier cost") and the interpretability of the classifier may be relevant during classifier design and application [13].

In this paper, an approach for generating fuzzy classifiers considering decision cost, classifier cost, and interpretability will be proposed. To evaluate the designed classifier, decision-theoretic measures are used in all design steps, i. e. a tree-oriented rule generation with subsequent pruning to generate generalized rules

and a selection of cooperating rules for a final rule base. The average cost per decision is the only performance measure.

The paper is structured as follows: Section 2 presents the design process, decision-theoretic measures, and interpretability aspects. In Section 3, a simple example and the resulting rule bases are discussed for different cost situations.

2 Rule generation and evaluation

2.1 Fuzzy system

A data set for supervised learning with N examples, features $x_l[k]$ $(k = 1, \ldots, N, l = 1, \ldots, s)$ and one observed output variable $y[k]$, labelled B_i $(i = 1, \ldots, m_y)$ is assumed. The fuzzy system to be generated contains rules of the following general structure:

$$P_r\text{: IF } \underbrace{\underbrace{x_1 = A_{1,Rr}}_{\text{partial premise } P_{r1}} \text{ AND } \ldots \text{ AND } \underbrace{x_s = A_{s,Rr}}_{\text{partial premise } P_{rs}}}_{\text{premise } P_r} \text{ THEN } y = C_r, \quad r = 1, \ldots, r_{max}$$

and a default rule $R_{r_{max}+1}$: ELSE $y = C_{r_{max}+1}$.

The premise P_r consists of a conjunctive (AND) combination of partial premises $P_{r1}, \ldots, P_{rs}$. The linguistic term $A_{l,Rr}$ can be a (primary) linguistic term $A_{l,i}$ $(i = 1, \ldots, m_l)$ of the feature x_l or a disjunctive (OR) combination of some neighboring or all linguistic terms of x_l, called derived linguistic term [14, 15]. In the case of all terms, this partial premise has no influence on the rule activation and is omitted in the presentation of the rule. Each rule conclusion C_r consists of one linguistic term $\hat{B}_j$. A maximum defuzzification chooses the best decision $\hat{B}_j$ that results from feature values and the generated rule base.

2.2 Criterion and probability estimation

2.2.1 Decision cost

The criterion used here originates from decision theory [16]. The expectation of the cost per decision $\hat{L}_T$ composed of decision cost $\hat{L}_D$ and classifier cost per decision L_C is estimated using a cost matrix $\mathbf{L}$ with elements $L(\hat{B}_j|B_i)$ and the probabilities of the decision-class combinations:

$$\hat{L}_T = L_C + \underbrace{\sum_{i=1}^{m_y} \sum_{j=1}^{m_y} L(\hat{B}_j|B_i) \cdot \hat{p}(\hat{B}_j \wedge B_i)}_{\hat{L}_D}. \tag{1}$$

$L(\hat{B}_j|B_i)$ denotes the cost of decision $\hat{B}_j$ given the actual class B_i and $\hat{p}(\hat{B}_j \wedge B_i)$ is the estimated joint probability of this decision-class combination. To differentiate between the evaluation of the tree, a single rule, or the whole rule base, the joint

probability can be estimated for the whole data set or part of it only (see Sections 2.3-2.5). Here, this measure is used to rank features during tree induction, single rule evaluation, or rule base selection.

The cost-optimal conclusion C_r for a generated rule with the given premise P_r is:

$$C_r = \underset{\hat{B}_j}{\operatorname{argmin}} \sum_{i=1}^{m_y} L(\hat{B}_j|B_i) \cdot \hat{p}(B_i|P_r), \tag{2}$$

where $\hat{p}(B_i|P_r)$ denotes the estimated conditional probability of the class $y = B_i$ within the examples that are covered by the premise P_r.

2.2.2 Classifier cost

The classifier cost for feature x_l per data set $L_{C,l}$ includes both fixed $L_{C,l,fix}$ and variable costs $L_{C,l,var}$. The fixed costs consist of the investment (engineering, asset cost e. g. for sensors and microcontrollers, and commissioning) prorated to the number of years the equipment is in use and the estimated operational fixed costs per year (e. g. staff, maintenance, energy, reduced availability of the device due to sensor failures). The fixed costs arise whether the equipment is in use or not. In contrast, the variable costs are directly related to the generation of a single example (e. g. consumable material). Thus, the classifier cost per data set $L_{C,l}$ is the sum of the total fixed costs divided by the number of examples per year N_{Test} and the variable costs:

$$L_{C,l} = \frac{L_{C,l,fix}}{N_{Test}} + L_{C,l,var}. \tag{3}$$

If $L_{C,l}$ is not precisely known, a rough estimation is reasonable to rank different features in a qualitative way. In addition, virtual costs like interpretability aspects and user preferences may be included.

The overall classifier cost includes the costs of all features used in at least one rule premise. With X_P representing the set of indices l of the employed features, the total feature cost L_C summarizes to:

$$L_C = \sum_{l \in X_P} L_{C,l} - L_{CD,l}(X_P). \tag{4}$$

The costs may be reduced by $L_{CD,l}$, if other features are used simultaneously [17]. For example, the cost of a feature is smaller, if another feature based on the same sensor information has already been chosen and there is no need for another sensor.

Considering the classifier cost in the design process leads to classifiers using mostly the features with a reasonable cost-information ratio (see example in Section 3). Additional information from the design process may indicate that certain rules are not selected due to high feature costs. This is illustrated by the explanation text presented in Section 2.6.

2.2.3 Probability estimation

All probabilities of fuzzy events are estimated by counting membership values in learning data and solving constrained optimization problems [14, 18]. As an exam-

ple, the probabilities $\hat{p}(P_r), \hat{p}(B_i), \hat{p}(B_i|P_r), \hat{p}(B_i \wedge P_r)$ are estimated by:

$$\hat{p}(P_r) = \frac{1}{N}\sum_{k=1}^{N} \mu_{P_r}(\mathbf{x}[k]),\ \hat{p}(B_i) = \frac{1}{N}\sum_{k=1}^{N} \mu_{B_i}(y[k]) \tag{5}$$

$$E = \min_{\mathbf{R}_{B|P}} \| \underbrace{\mathbf{R}_{B|P} \cdot \boldsymbol{\mu}_P}_{\hat{\boldsymbol{\mu}}_B} - \boldsymbol{\mu}_B \|_F^2 \tag{6}$$

$$\text{s.t. } \mathbf{R}_{B|P} \geq \mathbf{0}_{m_y \times (r_{max}+1)},\ \mathbf{1}_{m_y}^T \mathbf{R}_{B|P} = \mathbf{1}_{r_{max}+1}^T$$

$$\text{with } \mathbf{R}_{B|P} = ((\ \hat{p}(B_i|P_r)\ )) \in [0,1]^{m_y \times (r_{max}+1)},$$

$$\boldsymbol{\mu}_P = ((\ \mu_{P_r}(\mathbf{x}[k])\ )) \in [0,1]^{(r_{max}+1) \times N},$$

$$\boldsymbol{\mu}_B = ((\ \mu_{B_i}(y[k])\ )) \in [0,1]^{m_y \times N},$$

$$\hat{p}(B_i \wedge P_r) = \hat{p}(B_i|P_r) \cdot \hat{p}(P_r).$$

The matrix $\boldsymbol{\mu}_P$ refers to the rule activations $\mu_{P_r}(\mathbf{x}[k])$ of the r-th rule for the k-th example and the matrix $\boldsymbol{\mu}_B$ to the class assignment $\mu_{B_i}(y[k])$. In case of $\boldsymbol{\mu}_y \in \{0,1\}^{m_y}$, the problem is "crisp". In case of $\boldsymbol{\mu}_y \in [0,1]^{m_y}$, it is a fuzzy classification problem. The matrix $\mathbf{R}_{B|P}$ is used to estimate the conditional probabilities $\hat{p}(B_i|P_r)$ by minimizing the quadratic error E.

The joint probability $\hat{p}(\hat{B}_j \wedge B_i)$ of the combination of decision $\hat{B}_j$ and class B_i is estimated from the learning data set by

$$\hat{p}(\hat{B}_j \wedge B_i) = \sum_{r \in (C_r = \hat{B}_j)} \hat{p}(B_i|P_r) \cdot \hat{p}(P_r). \tag{7}$$

The constraints

$$\sum_{i=1}^{m_y}\sum_{j=1}^{m_y} \hat{p}(\hat{B}_j \wedge B_i) = 1,\ \sum_{r=1}^{r_{max}+1} \hat{p}(P_r) = 1, \tag{8}$$

are met by the optimization of (6) and the inference scheme proposed in [15]. This inference scheme computes $\mu_{P_r}(\mathbf{x}[k])$ by using the product as conjunctive operator for membership values $\mu_{A_{l,R_r}}(x_l[k])$. The bounded sum is used as disjunctive operator for the derived terms. A correction is made for overlapping premises.

The membership functions for each linguistic term $A_{l,i}$ are designed (e. g. by using fast heuristic methods with a fixed number of terms and triangular membership functions like clustering or similar sample frequencies for each term) independent of the cost matrix. An optimization of the membership functions upon the completion of the design process may further reduce the expected cost per decision.

2.3 Generation of decision trees

The proposed rule generation process of a fuzzy system consists of three steps. After the generation of decision trees, the extracted rule hypotheses are pruned and, finally, a rule base is selected. This design scheme was already proposed in [18] to minimize classifier errors. In this paper, cost-sensitive measures will be integrated in this scheme.

In the first step, one or more decision trees are induced. In each node of these trees, (2) is used to determine the best decision for the examples in the node. Afterwards, the feature x_l leading to minimal expected costs estimated with (1) is chosen to split the data set. To estimate this cost, an auxiliary "rule base", consisting of $r = 1, \ldots, m_l$ rules with premises $P_r = A_{l,r}$ and conclusions found by (2), is used. (1) is evaluated only for the examples in the node. Consequently, (1) is evaluated for all examples only in the root node of the decision tree. In the case of all conclusions C_r being equal, there is no further cost reduction splitting the tree, and the node is set to a terminal node with the conclusion C_r.

Otherwise, the algorithm creates m_l new nodes. The algorithm terminates when all nodes are terminal nodes. The probabilities $\hat{p}(B_i|P_r)$, $\hat{p}(P_r)$, $\hat{p}(B_i)$ in (2) and (7) are estimated only for the N_{node} examples in the actual node.

In order to obtain a comprehensive rule set, additional decision trees with different features in the root node are induced by step-wise discarding of the best features of previous trees.

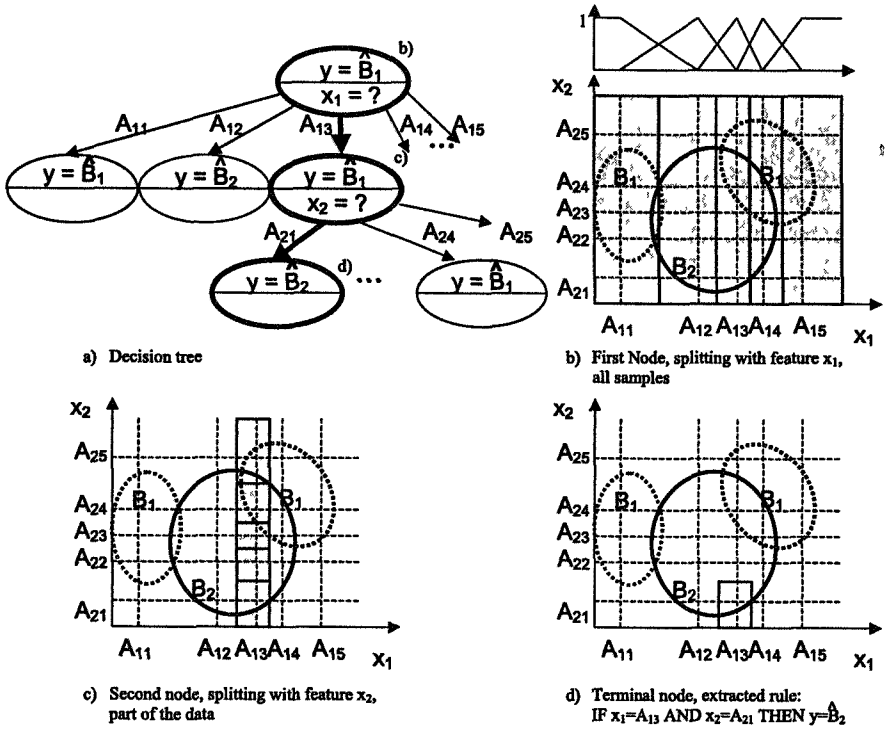

Figure 1: Example of tree generation, misclassifying B_1 is more expensive than B_2, parameters of membership functions (dotted lines), grey: 0.5 α-cut as an approximate visualization of the region for probability estimation

The induction algorithm employed here is similar to the popular ID3 algo-

rithm [19] and several methods for fuzzy decision tree induction [11, 20, 21, 22]. In contrast to these methods, the feature relevance (1) is used to choose features for decision nodes taking different decision costs into account.

2.4 Rule pruning

The rule hypotheses for the following pruning process are extracted from the terminal nodes of the decision trees. Fig. 1 shows the generation of a decision tree for the example explained in detail in Section 3. There are two overlapping classes with asymmetric costs of misclassification. The cost of misclassifying B_1 is set to ten times the cost of misclassifying B_2. Thus, the default decision in the root node is B_1.

Some work has already been done on cost-sensitive pruning of decision trees [23]. Pruning the whole tree by taking back several splits can not remedy the suboptimal selection of features in the first stages of tree generation. For this reason, we decided not to prune the whole tree, but the rule candidates extracted from the terminal nodes of the trees. The extracted rules are generalized one after the other. As long as there is an improvement of the rule evaluation, candidates for a generalization are generated by adding disjunctions with neighboring terms (e. g. $P_5 : x_1 = A_{14}$ OR $\underline{A_{15}}$ AND $x_2 = A_{25}$) or deleting partial premises (e. g. $P_2 : x_2 = A_{25}$) (see Fig. 2 for an example). The pruning options are indicated by arrows. In each step, the best alternative of the rule and pruning candidates is accepted. If there is no further improvement, the rule is saved and the next candidate is pruned.

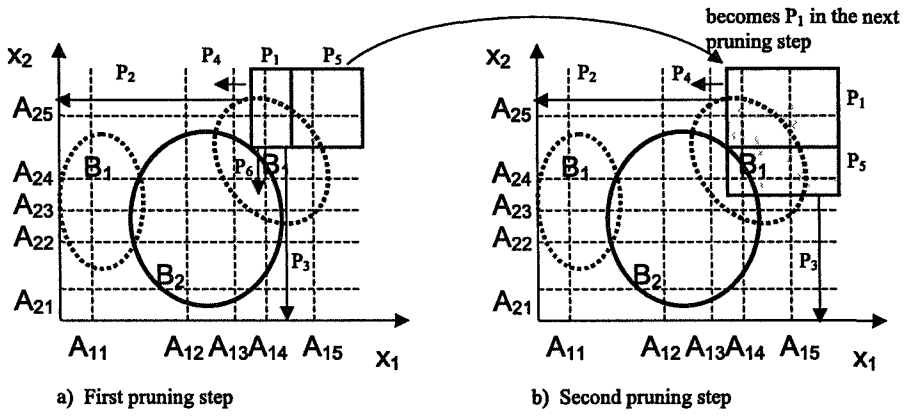

Figure 2: Pruning of the rule hypothesis R_1: IF $x_1 = A_{14}$ AND $x_2 = A_{25}$ THEN $\hat{y}$=1. First step: P_5 is accepted, second step: P_5 is accepted (renumbering of rules for each pruning step).

The evaluation criterion for single rules is also based on (1). Likewise during tree

generation, criterion (1) can be evaluated for the whole data set or part of it. An optimal single rule covers all examples of one class (rate of detection: $\hat{p}(P_r|C_r) \rightarrow 1$), but none of the other classes (rate of misclassification: $(1-\hat{p}(C_r|P_r)) \rightarrow 0$). In addition, the classifier cost for the information used in the rule premise P_r should be less than the cost reduction of the rule.

In many classification tasks, more than one rule is necessary to cover all examples of at least one class in a reasonable sense. Trying to cover all examples of a non-compact class with one premise may lead to a high number of misclassifications. In many applications, a compromise between a low rate of misclassification and a high rate of detection has to be found. This compromise depends on the cost of misclassifications. However, during rule generalization it is even more important to avoid misclassifications, because other rules with the same conclusion may exist.

To determine the set of examples for the evaluation of (1), two simple approaches exist. Either all examples or only those examples covered by the premise are considered. Both approaches are not suitable for the compromise discussed above. Suppose misclassifying B_1 is much more expensive than misclassifying B_2. Then, the premises P_1 and P_2 shown in Fig. 3 and the negated premises $\bar{P}_1$ and $\bar{P}_2$ have the same conclusion $C_1, C_{\bar{1}}, C_2, C_{\bar{2}} = \hat{B}_1$. For the compromise discussed, the premise P_2 is better than P_1, because P_2 covers more examples of B_1 and both premises have no misclassifications. However, the result of (1) will be equal for both premises, but depend on the examples used for evaluation $\hat{L}_D = L(\hat{B}_1|B_2) \cdot \hat{p}(B_2)$ (for all examples) or $\hat{L}_D = 0$ (for all examples in the premise). Thus, both approaches can not distinguish between the two premises P_1 and P_2.

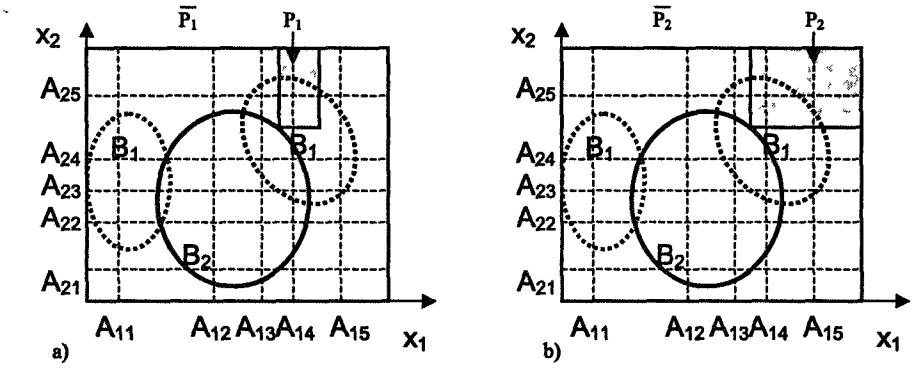

Figure 3: Pruning with criterion (1) for the rules R_1 (premise P_1) and R_2 (premise P_2): All examples or only premise; Misclassifying B_1 is more expensive than B_2

Therefore, (1) is calculated using all examples covered by the premise (cost of the premise) or belonging to the class of the conclusion and to the negated premise. For the latter examples, the decision of the negated premise is considered, because they might be misclassified by another premise with a different conclusion (potential cost). If the premise and the negated premise have the same conclusion, the second

best decision $C_{\bar{r}(2nd)}$ is taken. This is a pessimistic estimation of the potential cost. To evaluate a pruning candidate, $L_{D,r}$ is computed as follows:

$$\hat{L}_{D,r} = \underbrace{\sum_{i=1}^{m_y} [L(C_r|B_i) \cdot \hat{p}(B_i|P_r)]}_{\text{cost of the premise}} + \underbrace{\hat{p}(C_r|\bar{P}_r) \cdot \begin{cases} L(C_{\bar{r}}|C_r) & \text{for} \quad C_r \neq C_{\bar{r}} \\ L(C_{\bar{r}(2nd)}|C_r) & \text{for} \quad C_r = C_{\bar{r}} \end{cases}}_{\text{potential cost}} \tag{9}$$

This approach prefers the rule R_2 to R_1, because the examples not covered by P_1 increase the potential cost of R_1.

As the rules extracted from the tree are processed successively, the features already used in other rules are not known. Therefore, the classifier cost is not considered in the pruning process, as this would lead to oversimplified rules with too few features.

2.5 Rule base selection

Starting with the default rule and its resulting cost per decision, a rule base is chosen from the pool of pruned rules. Rules are added step by step, such that (1) computed for all examples is minimized. The conclusion of the default rule for all examples not yet covered by a premise can be fixed either manually to a decision $\hat{B}_j$ or automatically set by (2) (with $\hat{p}(P_{r_{max}+1}) = 1 - \hat{p}(\bigcup_{r=1}^{r_{max}} P_r)$). Overlapping premises are handled by the algorithm in [15]. Using one of these options, the algorithm tends to use the "ELSE" rule for one class. Suppose a problem with two compact classes and one premise covering one of the classes is already selected for the rule base. No cost reduction will result from the selection of another premise to cover the other class.

To force the algorithm to select premises for all classes, a third possibility is to introduce a rejection class $\hat{B}_{m_y+1}$ as conclusion of the "ELSE" rule. The examples in the rejection class are not assigned to a specific class and the cost of the decision $\hat{B}_{m_y+1}$ is the mean value of the lowest and second lowest cost of each class:

$$L(\hat{B}_{m_y+1}|B_i) = \frac{1}{2} \cdot \Big(\min_{j \text{ with } \hat{B}_j \neq \arg\min_j L(\hat{B}_j|B_i)} L(\hat{B}_j|B_i) + \min_j L(\hat{B}_j|B_i) \Big). \tag{10}$$

With this cost of the rejection class, the choice of a rule that assigns examples of the rejection class to the cheapest decision leads to a cost reduction. The cost of the rejection class is higher than the cheapest decision. On the other hand, it is cheaper to leave examples in the rejection class than to misclassify them. The cost of the rejection class is lower than any misclassification. Thus, the algorithm tends to find at least one rule for each class B_i. At the end of the design process, the examples that are still assigned to the rejection class are turned back into the ELSE rule with an automatically fixed conclusion. The algorithm stops when no further rule reducing (1) is found. To avoid large rule bases, a threshold for improvement can be defined.

It is important to state that due to the suboptimal results of all three design steps (tree induction, pruning, rule base selection), the final rule base will be sub-

optimal in general. The suboptimality is due to the step-by-step processes in all three design phases. However, a concurrent complete search over all possible rules and rule bases is not practicable for real-world problems with many features due to the combinatorial explosion of the search space.

Apart from the decision cost $\hat{L}_D$, the classifier cost per decision L_C is integrated in the criterion (1). The question arises in which design steps classifier cost should be considered? Only those features should be used for classification, whose costs are lower than their reduction of the decision cost. During tree induction and rule pruning, (1) is evaluated only for a part of the examples and, thus, it is not known whether a feature will be used for one or more rules in the final rule base. Splitting the tree or generalizing a pruning candidate, however, may depend on the cost of an additional feature. Hence, the classifier cost is considered only when (1) is evaluated for the whole data set, i. e. during rule base selection. Consideration of classifier cost in tree induction may lead to oversimplified rules when the development of rules terminates before the possible reduction of the decision cost during specialization (tree induction) and pruning (deleting subpremises instead of adding terms) is reached.

Alternative approaches which partially integrate classifier cost in tree induction and pruning will be investigated in future research.

Table 1 summarizes the design process.

Table 1: Cost criterion during design phase.

Design step	Rule premises	Examples	Cost
1. Tree Root node	m_l: one feature, all terms	N	(1) with $L_C = 0$
1. Tree Other nodes	m_l: additional feature, all terms	N_{node}	(1) with $L_C = 0$
2. Pruning	2: rule premise, negated premise	$N(\mathbf{x} = P_r \vee y = C_r)$	(9) with $L_C = 0$
3. Rule base selection	$r_{max}+1$: rule premises, negated premise	N	(1) with rejection class (10)

2.6 Interpretability and explanation of fuzzy rules

One definition of interpretability may be the following: "Interpretability is the degree to which one can assign qualitative meaning to an instrument's quantitative scores" [24].

Regarding our fuzzy system, interpretability means that human beings are able to understand the behavior of the fuzzy system when inspecting the rule base [18,

25]. The fuzzy rules generated usually are displayed in a technical manner (see Section 2). For interpretability it is more useful to express them in natural language text, as presented in [26]. In this text, the feature and class names, linguistic terms and frequency information gained during the design process (e. g. rate of detection) are combined with predefined text blocks.

In addition to the explanation of the rule base itself, an automatically created explanation text on the rule base's design process is presented. This explanation is helpful to interpret the cost-sensitive fuzzy system generated, as classification accuracy is not necessarily the consequence of the cost-sensitive design process (see example in Section 3).

For the explanation of the design process, the following information is considered to be interesting for interpretation:

- General information:
  - The best decision and the expected cost when no classifier is designed and a default decision is set,
  - the decision-theoretic design parameters.
- Rule-specific information:
  - The cost reduction and the rate of detection for each selected rule,
  - the feature cost corresponding to the rules,
  - overlapping rules with the same conclusion.
- Further information:
  - The best decision for examples not covered by the premises,
  - the final expected cost per decision for classification,
  - the cost reduction of the classifier compared to a default decision,
  - the reason for the rejection of certain rules.

Estimation of the cost of a default decision is based on the class distribution in the learning data set. It is the best decision in (2) for all examples (1). The expected cost of a designed classifier can be compared with the default cost. Especially in the case of high feature cost (e. g. very expensive sensors), a classification might not be useful as far as money is concerned. Likewise, the reason for the rejection of promising rules is often related to high feature cost.

To create the explanation text, a protocol of relevant information and prepared text blocks are used. For a detailed example, the reader is referred to the next section.

3 Example

The method will be explained by a simple illustrative example with $m_y = 2$ classes and $s = 4$ features. The class B_1 (abnormal) with $N_1 = 60$ examples is non-compact and consists of two subclasses B_{1a}, B_{1b}. This subdivision is not labelled in the learning data set. The class B_2 (normal) contains $N_2 = 300$ examples. The examples of both classes are produced by a constant mean value $\bar{x}_i(B_{1a}) = [2.5,\ 3,\ 1,\ -2.5]$, $\bar{x}_i(B_{1b}) = [-2,\ 2,\ 1,\ 2]$, and $\bar{x}_i(B_2) = [1,\ 2,\ 1,\ -1]$ with an additional non-correlated normal-distributed noise. The third feature x_3 is not

useful for classification, as the mean values for both classes are identical. The fourth feature x_4 is highly correlated with x_1 and gives almost redundant information only.

The feature costs of $x_1 - x_4$ (classifier cost) are $L_{C,l} = (0.15 \quad 0.07 \quad 0.05 \quad 0.03)$. There is no discount for simultaneous use of features: $L_{CD,l}(X_P) = 0$. The decision cost matrix is

$$\mathbf{L} = \begin{pmatrix} 0 & L(\hat{B}_1|B_2) \\ L(\hat{B}_2|B_1) & 0 \end{pmatrix}, \min(L(\hat{B}_1|B_2), L(\hat{B}_2|B_1)) = 1 \quad (11)$$

$$L_{Ratio} = L(\hat{B}_2|B_1)/L(\hat{B}_1|B_2) \quad (12)$$

where the decisions $\hat{B}_j$ are given in the rows and the actual classes B_i are listed in the columns.

The results of the proposed method for different ratios L_{Ratio} (12) are shown in Fig. 4. Here, estimated probabilities of misclassifications $\hat{p}(\hat{B}_2 \wedge B_1)$, $\hat{p}(\hat{B}_1 \wedge B_2)$, the rate of misclassification $\hat{p}(error)$, decision cost $\hat{L}_D$, classifier cost L_C, total cost $\hat{L}_T$, and the number of rules in the rule base are compared for three different approaches. The rule bases of all three classifier types are evaluated by (1) with (11) and the given feature costs (Table 2, Type 3). In the design phase, however, only Types 2 and 3 use the cost matrix in (11) and only Type 3 includes the feature cost (see Table 2). For Type 1 $L_{Ratio} = 1$ is used in the design process. For comparison of the three cost approaches both decision cost and classifier cost are taken into account (see Figure 4). Some resulting rule bases are shown in Table 3.

Table 2: Cost parameters during the design phase.

Classifier	L_{Ratio} (12)	$L_{C,l}$
Type 1	fixed to 1	$(0 \quad 0 \quad 0 \quad 0)$
Type 2	0.05 - 20	$(0 \quad 0 \quad 0 \quad 0)$
Type 3	0.05 - 20	$(0.15 \quad 0.07 \quad 0.05 \quad 0.03)$

Type 1 always generates the same solution, because it does not use the different misclassification costs. This leads to high decision costs at very small and very high values of the cost ratio in comparison to the other types. In addition, the classifier uses both redundant features x_1 and x_4, resulting in high classifier cost.

The main difference between the classifier designs in Type 2 and Type 3 compared to Type 1 is the acceptance of different pruning candidates, resulting in misclassifications as shown in Fig. 5 and Table 3. The selected rules are differentially generalized. Type 3 selects more generalized rules with less features to minimize classifier cost. Type 2 and Type 3 avoid more expensive misclassifications in uncertain situations and set these estimated probabilities to zero or close to it. As a consequence, both types accept higher probabilities of cheaper misclassifications to reduce the decision cost.

In addition, Type 3 is able to reduce the classifier cost by preferring the cheaper feature x_4 compared to the more expensive feature x_1 which contains almost the

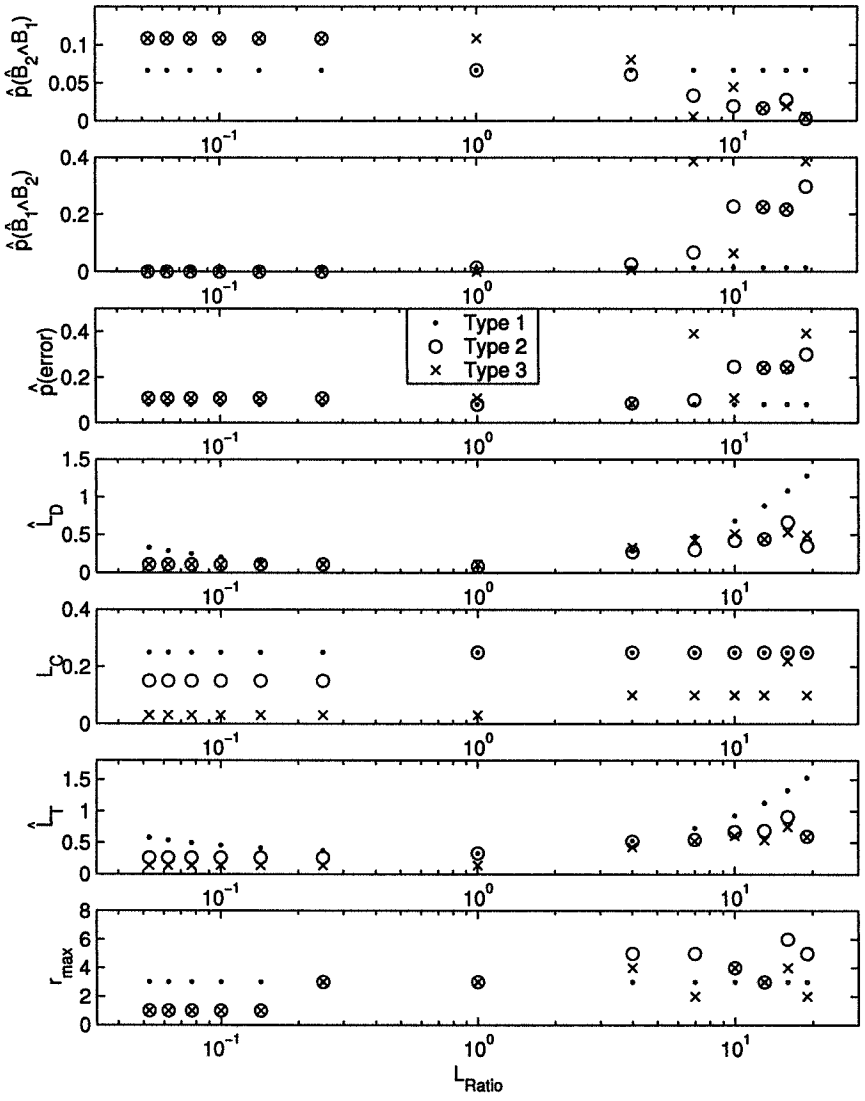

Figure 4: Estimated probabilities of misclassifications $\hat{p}(\hat{B}_2 \wedge B_1)$, $\hat{p}(\hat{B}_1 \wedge B_2)$ and rate of misclassification $\hat{p}(error)$, decision cost $\hat{L}_D$, classifier cost L_C, total cost $\hat{L}_T$, and number of rules in the rule base r_{max} (top down) as functions of the decision cost ratio L_{Ratio} for different cost approaches. For comparison purposes, classifier cost L_C are calculated for all three cost approaches using the values of Table 2 Type 3.

same information and partly by skipping x_2 with some loss of information. It reduces classifier cost without a significant increase in decision cost. Some small differences for $L_{Ratio} \approx 10$ in $\hat{L}_D$ between Type 2 and Type 3 (Type 3 causes less misclassifications using less features) are caused by the suboptimality of our approach. In combination with the reduced number of features used, the rule bases of Type 3 tend to consist of less rules. As expected, none of the classifiers uses feature x_3. In Table 3 the rule bases created for Type 1 ($L_{Ratio} = 1$) and Type 2/3 ($L_{Ratio} = 19$) are shown.

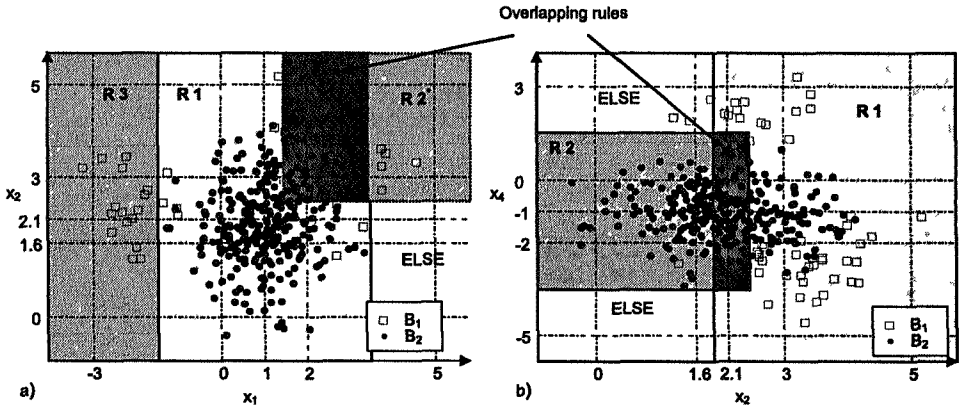

Figure 5: Selected rule base, a) Type 1 $L_{Ratio} = 1$ (only Rules 1 and 3), b) Type 3, $L_{Ratio} = 19$, R_2^* (left): Displays approximately the examples of R_2, as x_4 is negatively correlated with x_1

The results can also be displayed as Receiver Operating Characteristics (ROC) [6]. The commonly used threshold parameter is replaced by the cost ratio L_{Ratio}. The ROC-Graph for the detection of the samples of class B_1 shows depending on L_{Ratio} the relation between the incorrectly classified examples of class B_2 ($\hat{p}(\hat{B}_1|B_2)$) and the correctly classified examples of class B_1 ($\hat{p}(\hat{B}_1|B_1)$). Figure 6 shows this ROC-Graph. As the resulting rule bases are identically equal for $0.05 \leq L_{Ratio} \leq 0.25$ there is only one point in the ROC-Graph for these classifiers.

An example of an automatically created explanation text is given in the following box. It refers to the example for Type 3 with a cost ratio $L_{Ratio} = 19$ and included classifier cost.

Table 3: Created rule bases for the different cost approaches. Type 1: design phase with $L_{Ratio} = 1$, evaluation with $L_{Ratio} = 19$. Type 2 and 3: $L_{Ratio} = 19$ for design phase and evaluation. Misc.: Misclassifications, Ex.: Examples, rules are sorted by the selection order of the rule base.

R Nr.	Misc./Ex.	IF	THEN
Type 1, $L_{Ratio} = 19$, $L_D = 1.28$, $L_C = 0.25$: ($L_{Ratio} = 1$, $L_D = 0.08$, $L_C = 0.25$)			
R_1	39 /339	x_1=A_{12} OR A_{13} OR A_{14}	$y = B_2$
R_2	22 / 52	x_2=A_{24} OR A_{25} AND x_4=A_{41} OR A_{42}	$y = B_1$
R_3	0 / 16	x_1=A_{11}	$y = B_1$
R_4		ELSE	$y = B_1$
Type 2, $L_{Ratio} = 19$, $L_D = 0.35$, $L_C = 0.25$:			
R_1	179 /233	x_2=A_{23} OR A_{24} OR A_{25}	$y = B_1$
R_2	9 /229	x_1=A_{12} OR A_{13}	$y = B_2$
R_3	0 / 16	x_1=A_{11}	$y = B_1$
R_4	6 /225	x_2=A_{21} OR A_{22} OR A_{23} ... AND x_4=A_{42} OR A_{43} OR A_{44}	$y = B_2$
R_5	80 /115	x_1=A_{14} OR A_{15}	$y = B_1$
R_6		ELSE	$y = B_1$
Type 3, $L_{Ratio} = 19$, $L_D = 0.49$, $L_C = 0.1$:			
R_1	179/233	x_2=A_{23} OR A_{24} OR A_{25}	$y = B_1$
R_2	6 /225	x_2=A_{21} OR A_{22} OR A_{23} ... AND x_4=A_{42} OR A_{43} OR A_{44}	$y = B_2$
R_3		ELSE	$y = B_1$

The best default-decision for this problem without rule selection is B_1. This decision causes an expected average cost per decision of 0.83. To determine the cost of examples that are not covered by a premise a rejection class with the cost (9.5 0.5) was used. Thus, the rule base selection starts with expected cost per decision of 1.99. Feature cost have been considered during rule base selection.

Rule R_1 was selected, because it reduces the expected cost of misclassification from 1.99 to 0.82 per decision. The cost of the feature used for the selected rule is 0.07 (x_2). Thus, the total cost per decision is 0.89. Rule R_1 covers approx. 90 % of class B_1.

Rule R_2 was selected, because it reduces the expected cost for misclassification from 0.82 to 0.64 per decision. The costs for the features used for the selected rules are 0.1 (x_2 and x_4). Thus, the total cost per decision is 0.74. Rule R_2 covers approx. 73 % of class B_2.

For the examples that are not covered by a premise the decision with the lowest cost is B_1. As expected cost per decision incl. feature cost follows 0.59. The difference to the estimated cost in the last search step is due to the cost of the rejection class. The designed classifier reduces the expected cost per decision by 29 %.

Rule R_4 was not selected, because the additional feature x_1 causes a cost of 0.15 per decision. It is more expensive then the reduction of the expected cost for misclassifications of 0.07.

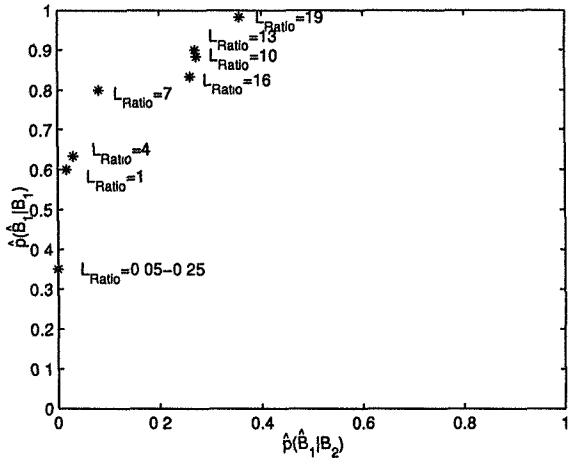

Figure 6: Results for Type 2 displayed as ROC-Graph. $\hat{p}(\hat{B}_1|B_1)$ corresponds to the true positive rate and $\hat{p}(\hat{B}_1|B_2)$ corresponds to the false positive rate.

4 Conclusions

The proposed method fully integrates decision-theoretic measures in the data-based design of fuzzy rule-based classifiers. Different cost types (decision cost, classifier cost, or other virtual cost e. g. related to interpretability) have been integrated in the classifier design process. Most alternative methods ignore these costs or only change rule conclusions for given premises, depending on costs. By means of an illustrative example, it was shown that in applications with asymmetric costs of misclassifications and classifier costs, this approach can reduce the total cost per decision. In such applications, the method proposed often reduces costs in comparison to other methods.

The explanation text proposed for the cost-sensitive design process improves interpretability of the created rule bases.

References

[1] J. J. Buckley, Y. Hayashi, Fuzzy neural networks: A survey, Fuzzy Sets and Systems 66 (1) (1994) pp. 1–13.

[2] F. Höppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy cluster analysis, Wiley, Chichester, 1999.

[3] J. Rives, FID3: Fuzzy induction decision tree, in: Proc. 1st Int. Symp. Uncertainty, Modelling and Analysis, IEEE Computer Soc. Press, Los Alamitos, Calif., 1990, pp. 457–462.

[4] A. Klose, An Evolutionary Algorithm for Semi-Supervised Fuzzy Classification, in: GMA/GI-Workshop Fuzzy-Systeme (2002) FZK-Bericht, FZKA 6767, pp. 100–106.

[5] S. Merler, C. Furanello, B. Larcher, A. Sboner, Automatic model selection in cost-sensitive boosting, Information Fusion 4 (2003), pp. 3–10.

[6] F. Provost, T. Fawcett, Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions, in: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), American Association for Artificial Intelligence (www.aaai.org), 1997.

[7] R. Keeney, H. Raiffa, Decisions with Multiple Objective, John Wiley & Sons, Inc., 1976.

[8] R. Bellman, L. Zadeh, Decision making in a fuzzy environment, Management Science 17 (4) (1970), pp. 141–163.

[9] C. Elkan, The foundations of cost-sensitive learning, in: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI01), 2001.

[10] Y. Elovici, D. Braha, A decision-theoretic approach to data mining, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 33/1 (2003), pp. 42–50.

[11] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, Wadsworth, Belmont, Ca., 1984.

[12] S. Beck, R. Mikut, J. Jäkel, G. Bretthauer, Decision-theoretic approaches in fuzzy rule generation for diagnosis and fault detection problems, in: Proc. 3rc International Conference in Fuzzy Logic and Technology (EUSFLAT 2003), 2003, pp. 558–563.

[13] P. Turney, Types of cost in inductive concept learning, in: Proc. Workshop On Cost-Sensitive Learning at the 17th International Conference on Machine Learning (WCSL at ICML-2000), Stanford University, California, 2000, pp. 15–21.

[14] J. Jäkel, L. Gröll, R. Mikut, Tree-oriented hypothesis generation for interpretable fuzzy rules, in: Proc. 7th Europ. Congr. on Intelligent Techniques and Soft Computing EUFIT'99, Sep. 13-16, Aachen, 1999, pp. 279–280.

[15] R. Mikut, J. Jäkel, L. Gröll, Inference methods for partially redundant rule bases, in: R. Hampel, M. Wagenknecht, N. Chaker (Eds.), Fuzzy Control: Theory and Practice, Advances in Soft Computing, Physica, Heidelberg, 2000, pp. 177–185.

[16] J. v. Neumann, O. Morgenstern, Theory of Games and Economic Behavior, Princeton University Press, 1953.

[17] P. Paclík, On feature selection with measurement cost and grouped features, in: Proceedings of SPR2002 Workshop, Windsor, Canada, http://www.ph.tn.tudelft.nl/ pavel/files/paclik02:SPR.pdf, 2002.

[18] R. Mikut, J. Jäkel, L. Gröll, Interpretability issues in data-based learning of fuzzy systems, Accepted paper Fuzzy Sets and Systems, 2004.

[19] J. R. Quinlan, Induction on Decision Trees, Machine Learning 11, 1986, pp. 81–106.

[20] P. E. Maher and D. St. Clair, Uncertain reasoning in an ID3 machine learning framework, IEEE International Conference on Fuzzy Systems, 1993, pp. 7–12.

[21] C. Z. Janikow, Fuzzy decision trees: Issues and methods, IEEE Transactions on Systems, Man, and Cybernetics 28(1), 1998, pp. 1–14.

[22] T. A. Runkler and S. Roychowdhury, Generating Decision Trees and Membership Functions by Fuzzy Clustering, in: Proc. 7th Europ. Congr. on Intelligent Techniques and Soft Computing EUFIT'99, Sep. 13-16, Aachen, 1999.

[23] J. Bradford, C. Kunz, R. Kohavi, C. Brunk, C. Brodley, Pruning decision trees with misclassification costs, in: Proceedings ECML-98, 1998, pp. 131–136.

[24] Instrument review criteria, Tech. rep., Scientific Advisory Committee (SAC), http://www.qolid.org/public/34sacrev.htm (1995).

[25] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), Trade-off between Accuracy and Interpretability in Fuzzy Rule-Based Modelling, Studies in Fuzziness and Soft Computing, Physica, Heidelberg, 2002.

[26] R. Mikut, T. Loose, J. Jäkel, Rule-oriented information acquisition from biological time series in clinical decision making, in: Proceedings: 10th Fuzzy Colloquium, Zittau, 2002, pp. 300–307.