Conservative and Creative Strategies for the Refinement of Scoring Rules

Joachim Baumeister, Martin Atzmueller, Peter Kluegl and Frank Puppe

Department of Computer Science, University of Wuerzburg, Germany
email: {baumeister,atzmueller,pkluegl,puppe}@informatik.uni-wuerzburg.de

Abstract

In knowledge engineering research the refinement of manually developed intelligent systems is still one of the key issues. Since scoring rules are an intuitive and easy to implement knowledge representation they are suitable for the manual development of knowledge bases. In this paper, we present a conservative and a creative strategy for the refinement of scoring rule bases adapting existing rule weights and inducing new rules, respectively. The usefulness of the approach is demonstrated by a case study with a reasonably sized rule base from the biological domain.

Introduction

The success of intelligent systems has proved over the last decades. Although, machine learning methods have influenced knowledge acquisition research, many systems are still manually built, e.g., the knowledge is acquired and formalized by a domain specialist because no sufficient data is available for using learning techniques. Often, several varieties of rules are used for the formalization, as they provide the representation of the input–output behavior of the intelligent system in an intuitive manner. However, if the systems are deployed into real-world application, then it often becomes necessary to refine the weights of particular rules, since some rule weights are usually incorrectly estimated and entered. Erroneous weighting of rules can appear because of several reasons:

- Knowledge was not formalized completely, since the developer of the rule base did not consider all relationships between input and output values of the domain.
- Knowledge about the relations has changed during the time of development and system use.
- There exists bias in the weighting of the rules implemented by the developer, i.e., to some rules a falsely higher/smaller weight was attached than to other rules.

Then, the refinement of the implemented rule base comes into place in order to correct parts of the rule base that showed erroneous behavior.

In the context of this paper we concentrate on a specialized representation of rules, i.e. *scoring rules*. Score-based approaches using rules go back to the INTERNIST/QMR project (Miller, Pople, & Myers 1982). Many researchers have adopted the ideas of the score-based representation and build (partially) successful systems. The LEXMED system (Schramm & Ertel 1999) is an example of a successful medical application using scores within the PIT system. Other recent medical applications using scores are described in (Buscher *et al.* 2002; Ohmann & et al. 1999; Eich & Ohmann 2000). The semantics of score-based rules are quite simple: A possible output (solution) of the system is not categorically derived by the observation of some input values (e.g., specified by categorical rules), but a particular solution receives a specified score point for each observation. In a further step, the output is derived as a possible solution, if the aggregated score points exceed a given threshold. However, when using the score-based rule representation classical refinement approaches, e.g. (Ginsberg 1988; Knauf *et al.* 2002), cannot be directly applied. Such approaches mainly focus on the generalization/specialization of rule conditions or the introduction of new categorical rules. In this paper, we discuss the refinement of scoring rules by the modification of the scoring points and by the induction of new scoring rules. We will demonstrate the applicability of the approach by a case study using a real-world sized knowledge base.

The rest of the paper is organized as follows: In the next section we briefly introduce the basic notions of scoring rules and their inference. Then, two basic approaches are presented for the refinement of score-based knowledge: The conservative refinement strategy only focuses on the appropriate modification of scoring points, whereas the creative refinement strategy also considers the induction of new scoring rules. Thereafter, we exemplify the utility of the approach in a case study using a reasonably sized knowledge base from the biological domain. We conclude the paper with a discussion and an outlook to future work.

Rule Bases with Scoring Rules

We present a scoring formalism, which uses a rule-based representation augmented with scoring points to describe a quantitative weighting of the stated implication.

Scores are a well-known concept for diagnostic reasoning in medical decision making. For each diagnosis (solution, output) an account (score) is used for inferring the state of

Proc. of the 19th International Florida Artificial Intelligence Research Society Conference (FLAIRS-2006), pages 408-413, AAAI Press , 2006

this diagnosis. In its simplest form, any observed finding (input value) can contribute to the score of a specified diagnosis. Then, the state of the diagnosis is determined by a given threshold value. In its general form, not only isolated observations of findings can contribute to a diagnosis score, but also conditioned observations among findings. Rule-based approaches for implementing structural knowledge with uncertainty were mainly influenced by the work of the MYCIN project (Buchanan & Shortliffe 1984), and have been undergoing fruitful research for the last decades.

Knowledge Representation

In the following we define the necessary notions for the refinement task. First, we want to consider the objects that are used as an input and output of the rule system.

Definition 1 (Input and Output) Let Ω_{sol} be the universe of all *solutions* (outputs) and Ω_a the set of all *attributes* (inputs). A range $dom(a)$ of values is assigned to each attribute $a \in \Omega_a$. Further, we assume Ω_{obs} to be the (universal) set of observable *input values* $a\!:\!v$, where $a \in \Omega_a$ is an attribute and $v \in dom(a)$ is an assignable value. An observable input $a\!:\!v$ is often called a *finding*. The universe set of all possible values is defined by Ω_V, i.e., $\Omega_V = \cup_{a \in \Omega_a} dom(a)$.

A problem-solving session is described by a case.

Definition 2 (Case) A *case* c is defined as a tuple

$$c = (\, OBS_c, SOL_c \,)\,,$$

where $OBS_c \subseteq \Omega_{obs}$ is the *problem description* of the case, i.e., the observed finding set of the case c. The set $SOL_c \subseteq \Omega_{sol}$ is the set of solutions of the case.

A simple and intuitive way for representing inferential knowledge is the utilization of the *diagnostic scores* pattern (Puppe 2000; Puppe *et al.* 2001). Then, simple scoring rules are used for describing the domain knowledge.

Definition 3 (Simple Scoring Rules) A scoring rule is denoted as follows: $r = cond_f \xrightarrow{s} o$, where $f \in \Omega_{obs}$ is an attribute value, $s \in \mathbb{N}$ denotes the scoring point $sp(r)$ of the rule, and $o \in \Omega_{sol}$ is the targeted solution/output. Further, let $cond_f$ be a condition for the finding f, e.g., for $f = a\!:\!v$ we can define an equals-condition $a = v$.

If the condition of a scoring rule consists of more than one atomic condition, i.e., connected by conjunctions/disjunctions, then we call these rules *general scoring rules*. We will call the collection of all scoring rules defined for a specific domain a *rule base* $\mathcal{R}$.

Inference

Deriving a solution using scoring rules is very simple: For each solution o a scoring account $\mathcal{A}_{o,c}$ is created, which is initially set to 0 for each problem solving session, i.e. the creation of a case c. In a problem solving session the user enters findings and the particular rules are evaluated according to their rule conditions. A rule fires when the condition evaluates true. If a rule $cond_f \xrightarrow{s} o$ fires, then the scoring point s is added to the account $\mathcal{A}_{o,c}$. Whenever the sum of the added numbers exceeds a given threshold, the corresponding solution is considered to be established, i.e. derived as a possible explanation for the entered case c.

Knowledge Acquisition

When building a knowledge base using scoring rules the domain specialist typically tries to rate all correlations between the findings and the solutions. Each finding–solution correlation is rated by a point score estimated by the domain specialist's experience. By rating only single finding–solution relations by scores we avoid the creation of ambivalent or subsuming rules.

The scoring points are aggregated by a simple (linear) sum function. Thus, the weightings of relations are not normalized, but the final state of a score is determined using a fixed threshold value. If the strength of a combination of attribute values is disproportionate when compared to the single observation of the attribute values, then the presented knowledge representation is not appropriate, since the particular attribute values can only contribute to a diagnostic score in a linear way. This problem is commonly tackled by introducing an abstracted attribute, for which its values are derived w.r.t. the values of the "combined" attributes. Subsequently, the abstracted attribute is used instead of the combined attributes. For a detailed discussion of the acquisition, inference and evaluation of score-based rules we refer to (Baumeister 2004, p. 91).

The Refinement of Scoring Rule Bases

Recently, interactive methods for the refinement of scoring rule bases have been presented, cf. (Atzmueller *et al.* 2005). The approaches were motivated by the fact that available methods only provide limited control over the refinement process and make assumptions both on the correctness of the knowledge base and the available test case base. In real-life scenarios those assumptions do not necessary hold and therefore subgroup mining methods were proposed to support the domain specialist during the refinement task. Then, the results of a subgroup mining session were used to identify potential faults in the knowledge base but also in the (often manually) collected case base. In the context of this work we assume a case base that only contains correctly solved cases.

In this paper, we present an automatic approach for the refinement of scoring rule bases. In order to motivate the basic parts of the proposed refinement approach we briefly will show the relations of scoring rule bases with perceptron networks. Thus, the repair strategy of the refinement method was motivated by the training strategy of perceptrons.

Relation with Perceptron Networks

Since scoring rules are used to directly connect input values (findings) with possible outputs of the system, there exist strong similarities with perceptron networks. Thus, the possible input value f of the network is also directly connected with an output o of the network. The scoring point s of a rule $cond_f \xrightarrow{s} o$ is related with the perceptron weight of a particular input value. Given a collection of input values $f_1, \ldots, f_n$ connected with a solution o by scoring rules $cond_{f_i} \xrightarrow{s_i} o$ the state of the solution is calculated according to the following equation:

$$s_o(f_1, \dots, f_n) = \begin{cases} derived(o) & \text{if } s_1 + \dots + s_n > \mathcal{T} \\ \neg\, derived(o) & \text{otherwise} \end{cases}, \tag{1}$$

where $\mathcal{T} > 0$ is the globally defined threshold for deriving the particular solutions. The sum of score points s_i corresponds to the scoring account $\mathcal{A}_{o,c}$ for a given output o and input values f_i in a given case c.

In general, we can define the derivation state s_o of a given solution o w.r.t. a given case $c = (\, OBS_c, SOL_c\,)$

$$s_o(c) = \begin{cases} derived(o) & \text{if } \sum\limits_{r \in \mathcal{R}_{o,c}} sp(r) > \mathcal{T} \\ \neg\, derived(o) & \text{otherwise} \end{cases}, \tag{2}$$

where $\mathcal{R}_{o,c} = \{\, r \in \mathcal{R} \,|\, r = cond_f \xrightarrow{s} o \wedge f \in OBS_c \,\}$ are all scoring rules in the rule base $\mathcal{R}$ with the solution o in the consequent, and for which the rule condition was satisfied w.r.t. the case c; again, let $\mathcal{T} > 0$ be a globally defined threshold for the derivation of the particular solutions.

The function $s_o(c)$ is equivalent to the perception function $o(\overrightarrow{s})$, where $\overrightarrow{s}$ is the vector of all input weights. In Figure 1 the inference schema of a scoring rule base for one output o is depicted. Thus, deriving a solution for a specified problem is done in the same way for a perceptron as done for a scoring rule base. In consequence, we will present a conservative approach for the refinement of scoring rules that was inspired by the gradient decent search of perceptron networks.

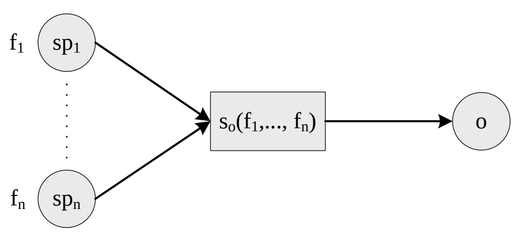

Figure 1: The inference in scoring rule bases

The Refinement Strategy

For the proposed refinement approach we assume that the case base is already in a correct state, i.e., all cases c contain a correct solution SOL_c for the specified problem description OBS_c. For manually collected cases this assumption is not always true and therefore interactive methods for findings inconsistencies in cases may be applied, e.g. in (Atzmueller *et al.* 2005) a method based on a statistical analysis was described.

The presented approach distinguishes two refinement strategies: The *conservative refinement strategy* maintains the structure of the original rules but only adapts the corresponding scoring points w.r.t. the given test case base. In contrast, the *creative refinement strategy* adds new rules, i.e. input–output relations, to the rule base that were not included in the rule base before, but show a significant relation w.r.t. the given test case base. We will discuss the application of the two strategies in the following in more detail.

Conservative Refinement Strategy Here, only scoring points of existing rules are adapted but no new rules are induced. The strategy is applicable, if the developer of the rule base feels certain that all necessary connections between input and output values are modeled, and that only the weights of the particular rules need to be refined.

We distinguish two conservative refinement strategies: The *general refinement strategy* selects the incorrectly derived output and uniformly adapts all scoring points of rules for this output. The *analytical refinement strategy* only adapts the relevant scoring points of rules that have been actually used (i.e. fired) in the incorrectly solved cases. For both approaches we can give reasonable motivations: If the developer of the rule base feels certain that the distribution of the particular scoring points for a given output has been modeled very carefully, then the general refinement strategy should be applied. Here the scoring points of all relations are uniformly adapted, and we therefore preserve the overall proportions of the weights. In contrast, if the developer of the rule base feels not confident w.r.t. the proportions of the scoring points, then the analytical refinement strategy should be more reasonable. Since only a selection of rules, i.e. the used rules for a case, are considered for refinement, only the corresponding scoring points are disproportionately increased/decreased w.r.t. the overall proportion.

The general algorithmic framework is given as follows: The available training set of cases is successively given to the rule system in order to derive a solution for every given case c. If the derived solution $SOL_{c'}$ differs from the (correct) solution SOL_c stored in the training case, then the false positive and false negative outputs are determined. For each false output the deviation $\Delta(o, SOL_{c'})$ of the error is computed and propagated to the scoring points of the relevant rules. This basic procedure terminates, if all cases were solved correctly or the number of iterations has exceeded a maximum iterations threshold. Algorithm 1 describes the method in more detail.

For the algorithm we introduce the following notions: The functions $fn(SOL_c,\, SOL_{c'})$ and $fp(SOL_c,\, SOL_{c'})$ determine the false negative and false positive outputs w.r.t. the cases c and c'. The Δ-error of a given output $o \in \Omega_{sol}$ w.r.t. a tested case c' is defined as the deviation

$$\Delta(o, SOL_{c'}) = \mathcal{T} - \mathcal{A}_{o,c'}\,, \tag{3}$$

where $\mathcal{T}$ is the fixed threshold for deriving a solution, and $\mathcal{A}_{o,c'}$ is the scoring account of the output o in the case c', i.e. the actual sum of the aggregated scoring points w.r.t. case c'. The set $\mathcal{R}_o$ is defined according to the actual conservative refinement strategy: For the general refinement strategy $\mathcal{R}_o = \{\, r \in \mathcal{R} \,|\, r = f \xrightarrow{s} o \}$, i.e., the set of all scoring rules deriving the specified output o; for the analytical refinement strategy we define $\mathcal{R}_o = \mathcal{R}_{o,c'}$, i.e., all rules actually deriving the specified output o in the given case c'. In summary, we see that the conservative refinement approach shows strong similarities with the simple gradient decent search used in perceptron learning.

Algorithm 1 Conservative Refinement Strategy

repeat
  for all $c \in cases$ **do**
    $c' \leftarrow$ runCase(c)
    if $SOL_c \neq SOL_{c'}$ **then**
      $F = fp(SOL_c, SOL_{c'}) \cup fn(SOL_c, SOL_{c'})$
      for all $o \in F$ **do**
        $\delta_s = \Delta(o, SOL_{c'}) \,/\, |\mathcal{R}_o|$
        for all $r \in \mathcal{R}_o$ **do**
          for $r = cond_f \xrightarrow{s} o$ **do**
            $s \leftarrow s + \delta_s$
          end for
        end for
      end for
    end if
  end for
until all cases have been solved correctly **or** the number of iterations exceeds a given threshold

Creative Refinement Strategy The creative refinement approach applies a subgroup mining method in order to add new rules, i.e., input–output relations. Subgroup mining or subgroup discovery (Wrobel 1997) is a method to identify relations between a dependent variable (target variable) and usually many explaining, independent variables. For example, consider the subgroup described by "*smoker=true* AND *family history=positive*" for the target variable *coronary heart disease=true*.

Subgroup discovery does not necessarily focus on finding complete relations; instead partial relations, i.e., (small) subgroups with "interesting" characteristics can be sufficient. Therefore, subgroup mining is especially suitable in our setting, since such local relations can then be represented in the score pattern.

In our setting the proposed subgroup mining task mainly relies on the following properties: the target variable, the subgroup description language, and the applied quality function: We consider binary target variables $t \in \Omega_{sol}$ denoting solutions, i.e., the outputs. The subgroup description is defined using a subgroup description language, e.g., using conjunctions of input values $a : v \in \Omega_V$. More formally, a subgroup description $sd = \{e_l\}$ is defined by the conjunction of a set of selection expressions. These selectors $e_i = (a_i, V_i)$ are selections on domains of attributes, $a_i \in \Omega_a, V_i \subseteq dom(a_i)$. For the refinement task we only consider conjunctions of single input values, i.e., $|V_i| = 1$. Furthermore, we restrict the subgroup mining task to subgroup descriptions containing a single selector only, i.e., the subgroup descriptions are of the form $sd = \{e_l\}$ with $l = 1$. Then, a subgroup for the target variable t and the subgroup description $sd = \{(a_i, v_i)\}$ can be formalized as the rule $r = cond_f \xrightarrow{s} t$, where $f = a_i : v_i$ and the score s is dependent on the association strength between t and f.

A quality function measures the interestingness of the subgroup and is used by the search method to rank the discovered subgroups during search. Typically, quality functions are based on a statistical evaluation function, e.g., the $\chi 2$-test for independence. For the $\chi 2$-test we can construct a 2×2 contingency table containing the target and non target counts w.r.t. the subgroup and its complementary group, respectively. Then, the quality of the subgroup is given by the statistical significance of the test considering to the null hypothesis that the distribution of the target variable does not differ comparing the subgroup and the complementary group. We use a certain significance level α to ensure statistically significant relations.

Algorithm 2 Creative Refinement Strategy

1: Consider global blacklist $B \subseteq \Omega_{sol}$
2: **for all** $c \in cases$ **do**
3:   runCase(c), and update the frequency of incorrectly solved solutions
4: **end for**
5: Retrieve the set SOL_k, i.e., the set of size k containing the most frequent solutions that were solved incorrectly
6: $SOL_k = SOL_k - B$
7: **if** $SOL_k \neq \emptyset$ **then**
8:   **for all** $t \in SOL_k$ **do**
9:     $V = \{v \in \Omega_V \mid \nexists rule(a\!:\!v \rightarrow t\}$
10:     $S =$ mineSubgroups(t, a, V)
11:     $S' = \{s \in S \mid q(s) > \alpha\}$
12:     reduce S' to the m best subgroups
13:     **for all** $s \in S'$ **do**
14:       compute the subgroup score rating $r_s = \phi(s) \cdot \mathcal{T}$
15:       **if** $|r_s| > 0$ **then**
16:         create a rule r with the assigned score r_s, i.e., $r = cond_{f_s} \xrightarrow{r_s} t$, where f_s is the finding contained in the subgroup description of s
17:       **end if**
18:       **if** no rule for t was created **then**
19:         $B = B \cup \{t\}$
20:       **end if**
21:     **end for**
22:   **end for**
23: **end if**
24: Apply a conservative refinement strategy (see Alg. 1)

For the creative refinement approach we discover subgroups for the k most frequent solutions SOL_k that were solved incorrectly. In each iteration, statistically significant subgroups for each target $t \in SOL_k$ are retrieved. We restrict the subgroup mining method to the input values V that are not already contained in rules for the specific solution, since the rule already exists (see lines 9/10). Then, based on the association strength between the single factor contained in the subgroup and the target variable, we compute the scoring point and create a rule, if the confirmation strength is not zero. To compute the association strength we use the correlation coefficient between subgroup and target variable, which simplifies to the ϕ-coefficient for binary variables, given by the subgroup description and the target variable. The ϕ-coefficient is determined by utilizing the same contingency table that is used for the $\chi 2$-test. If no significantly strong rules could be created for a given solution, then it is added to a blacklist, i.e., a list of solutions excluded from the refinement step. This provides a suitable termination criterion, since the algorithm can terminate if only solutions contained in the blacklist are falsely solved, then these solutions have been already considered for rule creation. In each iteration only the m best subgroups of a solution are considered. By this step we imply a hill-climbing strategy by creating only the statistically most significant (and important) rules: In the most conservative case only the best rule is considered, i.e., by setting $m = 1$. In such a conservative setting the entire Algorithm 2 should be iterated. A necessary step in the creative refinement method is the application of a conservative approach approach after the new rules have been created: While the assigned scoring points express the general strength of the relations they may need to be fine-tuned in order to increase the accuracy of the rule base. The algorithm also considers the existence of a blacklist B of solutions that should be excluded from the refinement process, i.e., no new relation for this solutions should be discovered. Such a blacklist can be helpful if the algorithm should be iterated. However, in our case study we only used one iteration of the creative strategy with a subsequent application of a conservative refinement strategy, since after the first iteration no improvement of the knowledge quality was measured. The algorithm for the creative refinement method is shown in Algorithm 2.

Extensions to the proposed method include to add complex rules to the rule base, i.e., rules created from subgroups that contain more than one selector. However, this is not implemented in the approach since it does not conform to the proposed score knowledge representation.

Case Study

The applicability of the presented approach was evaluated using a rule base from the biological domain: The plant system (Ernst 1996) is a consultation system for the identification of the most common flowering plants vegetating in Franconia, Germany. For a classification of a given plant the user enters findings concerning the flowers, leaves and trunks of the particular plant. Since the system was planned to be used by non-expert users the scoring rule pattern was applied in order to increase the robustness of the system w.r.t. possibly erroneous data acquisition.

Experimental Setting

The rule base contains 6763 scoring rules stating relations between input and solution values. The applied case base consists of 546 cases containing in total 94 different output values (solutions) and 133 inputs (with different values for each input). Due to the restructuring of some rules the rule base was not in a valid state, i.e., 220 cases were solved incorrectly (84 false negatives, 136 false positives). For the experiments we applied a stratified 5-fold cross-validation; the creative refinement strategy used the significance level $\alpha = 0.05$, i.e., with confidence level of 95%, for finding statistical significant relations.

Results

In the first experiment we implemented the **conservative refinement strategy**; the incorrect cases were reduced to 9.4 cases with false negative outputs and 10.6 cases with false positive outputs (avg. per fold). The second experiment considered the analytical refinement strategy: Here we were able to reduce the errors to 6.2 cases with false negative outputs and 8.4 cases with false positive outputs (avg. per fold).

Conservative refinement strategy

Original rule base		Refined (general)	Refined (analytical)
FP	136	10.6	8.4
FN	84	9.4	6.2

Table 1: Number of false positive (FP) and false negative (FN) cases for the original rule base and the conservative refinement strategies, respectively.

In Table 1 the results for the conservative refinement strategy are shown: The numbers of cases with false positive (FP) and false negative (FN) outputs are displayed w.r.t. the original rule base, the general refinement strategy and the analytical refinement strategy, respectively.

In order to evaluate the usefulness of the **creative refinement strategy** we synthetically worsened the quality of the rule base. Three samples of the already refined rule base were generated by randomly removing 300 rules for each sample. Then, we applied the creative refinement strategy in order to see if the manually deleted relations can be rediscovered and appropriately adjusted in the samples. Subsequently, an analytical refinement strategy was performed to adjust the scoring points. The initially 66.1 cases with false negative outputs and 40.8 cases with false positive outputs (avg. per fold and sample) were reduced to 6.1 cases with false negative outputs and 7.1 cases with false positive outputs. In Table 2 a more detailed analysis of the refinement results is shown for each sample.

Creative refinement strategy

	Sample 1		Sample 2		Sample 3		Avg. samples	
	FP	FN	FP	FN	FP	FN	FP	FN
Original	31.2	37.4	45.6	60.0	45.6	100.8	40.8	66.1
Refined	3.6	4.6	6.8	6.6	10.8	7.0	7.1	6.1

Table 2: Results of the creative refinement strategy with number of false positives (FP) and false negatives (FN) for each sample and averaged for all samples.

As expected the results show that the analytical strategy performs better than the general strategy. Since the analytical refinement strategy implements a more goal-oriented modification of scoring points, i.e. by adapting only points that were actually used in the particular cases, an improved refinement was expected at the beginning of the case study. The analysis of the creative refinement strategy showed that the removed relations were naturally re-inserted into the rule base with some additional rules, that were either missed by the domain specialist or were discovered due to statistical significance; afterwards, these rules were adapted by the analytical strategy. Therefore, we can observe slightly better results than for the standard analytical strategy.

Conclusion

In this paper we introduced a novel refinement technique applicable to the scoring rule representation. Due to the simple and intuitive manner this representation is used for the manual construction of intelligent systems, and therefore the refinement of such systems is often necessary. The presented method distinguishes a conservative refinement strategy, which basically adapts the scoring points of responsible rules, and a creative refinement strategy, which additionally tries to induce new scoring rules for hidden input–output relations.

In general, the conservative refinement approach should be applied if the developer of the rule base feels confident that the knowledge is almost completely covered and no new relations should be inserted. The general refinement strategy is reasonable to preserve the original distribution of the scoring weights. In contrast, the analytical refinement strategy promises even better improvements of the knowledge quality but changes the overall proportions of the scoring weights. The creative refinement strategy is considered when new relations should be discovered in order to refine the knowledge quality. The induction of new relations can be necessary if the domain has changed during development (e.g. new input values were introduced in the system but are not covered by the rule base) or if the domain is not considered to be completely covered by the rule base.

We demonstrated the usefulness of the particular approaches with a case study utilizing a reasonable sized knowledge system taken from the biological domain. Here, a 5-fold cross-validation was implemented for all experiments; for the creative refinement strategy a 5-fold cross-validation was implemented with three randomly generated samples in order to reduce unintentionally included bias.

The presented method only induces simple scoring rules, i.e., rules with only one constrained finding in the rule condition. In the future, we are planning to consider the induction of complex scoring rules containing conjunctions of simple conditions. Although such rules do not conform to the diagnostic scores pattern, they are useful to represent the disproportional confirmation or disconfirmation of a solution w.r.t. a combined observation of a set of findings. Due to the generality of the presented subgroup mining approach this extension can be simply included. Furthermore, the application of domain knowledge may be very useful to support the refinement task: Here, the developer of the rule base is able to specify a collection of input–output relations that should not be refined or that are considered for the refinement with a higher priority. In such a semi-automatic scenario the developer would insert domain knowledge in order to target on a particular area in the knowledge base to be refined. Another promising extension of the presented work is the inclusion of negated findings during the subgroup discovery process. Creating rules with negated findings in their rule condition does not conform with the compared perceptron representation but may increase the expressiveness of the knowledge significantly. Therefore, we also expect an improvement of the refinement results.

References

Atzmueller, M.; Baumeister, J.; Hemsing, A.; Richter, E.-J.; and Puppe, F. 2005. Subgroup Mining for Interactive Knowledge Refinement. In *Proceedings of the 10th Conference on Artificial Intelligence in Medicine (AIME 05), LNAI 3581*, 453–462. Springer.

Baumeister, J. 2004. *Agile Development of Diagnostic Knowledge Systems*. Number 284 in DISKI. infix.

Buchanan, B., and Shortliffe, E. 1984. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.

Buscher, H.-P.; Engler, C.; Führer, A.; Kirschke, S.; and Puppe, F. 2002. HepatoConsult: A Knowledge-Based Second Opinion and Documentation System. *Artificial Intelligence in Medicine* 24(3):205–216.

Eich, H.-P., and Ohmann, C. 2000. Internet-Based Decision-Support Server for Acute Abdominal Pain. *Artificial Intelligence in Medicine* 20(1):23–36.

Ernst, R. 1996. Untersuchung verschiedener Problemlösungsmethoden in einem Experten- und Tutorsystem zur makroskopischen Bestimmung krautiger Blütenpflanzen [Analysis of various problem solving methods with an expert and tutoring system for the macroscopic classification of flowers]. Master's thesis, University Würzburg, Biology department.

Ginsberg, A. 1988. *Automatic Refinement of Expert System Knowledge Bases*. Morgan Kaufmann Publisher.

Knauf, R.; Philippow, I.; Gonzalez, A. J.; Jantke, K. P.; and Salecker, D. 2002. System Refinement in Practice – Using a Formal Method to Modify Real-Life Knowledge. In *Proceedings of 15th International Florida Artificial Intelligence Research Society Conference 2002 Society (FLAIRS-2002)*, 216–220. Pensacola, FL, USA: AAAI Press.

Miller, R. A.; Pople, H. E.; and Myers, J. 1982. INTERNIST-1, an Experimental Computer-Based Diagnostic Consultant for General Internal Medicine. *New England Journal of Medicine* 307:468–476.

Ohmann, C., and et al. 1999. Clinical Benefit of a Diagnostic Score for Appendicitis: Results of a Prospective Interventional Study. *Archives of Surgery* 134:993–996.

Puppe, F.; Ziegler, S.; Martin, U.; and Hupp, J. 2001. *Wissensbasierte Diagnosesysteme im Service-Support [Knowledge-Based Systems for Service-Support]*. Berlin: Springer.

Puppe, F. 2000. Knowledge Formalization Patterns. In *Proceedings of PKAW 2000*.

Schramm, M., and Ertel, W. 1999. Reasoning with Probabilities and Maximum Entropy: The System PIT and its Application in LEXMED. In et al., K. I., ed., *Operations Research Proceedings*, 274–280. Springer.

Wrobel, S. 1997. An Algorithm for Multi-Relational Discovery of Subgroups. In Komorowski, J., and Zytkow, J., eds., *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-97)*, 78–87. Berlin: Springer.