Introspective Subgroup Analysis for Interactive Knowledge Refinement

Martin Atzmueller and **Joachim Baumeister** and **Frank Puppe**
Department of Computer Science,
University of Würzburg, 97074 Würzburg, Germany
{atzmueller, baumeister, puppe}@informatik.uni-wuerzburg.de

Abstract

When knowledge systems are deployed into a real-world application, then the maintenance and the refinement of the knowledge are essential tasks. Many existing automatic knowledge refinement methods only provide limited control and clarification capabilities during the refinement process. Furthermore, often assumptions about the correctness of the knowledge base and the cases are made. However, such assumptions do not necessarily hold for real-world applications. In this paper, we present a novel interactive approach for the refinement of knowledge bases: Subgroup mining is used to discover local patterns that describe factors potentially causing incorrect behavior of the knowledge system. The approach is supplemented by introspective subgroup analysis techniques in order to help the user with the interpretation of the refinement recommendations proposed by the system.

Introduction

The refinement of knowledge systems is a crucial success factor for the implementation and maintenance of systems deployed into real-world applications. When the knowledge base is built manually, then typically refinements are necessary throughout the initial deployment phase. Sometimes, the developed knowledge base is still incomplete. In consequence, extensions and not only modifications of the knowledge have to be applied in order to improve the reliability of the system.

In the past, many approaches for the automatic refinement of knowledge bases have been proposed, e.g., (Ginsberg 1988; Boswell & Craw 1999; Carbonara & Sleeman 1999; Knauf *et al.* 2002). In this paper, we propose a less automatic but user-guided approach for carrying out refinements of a knowledge base. For finding *hot-spots* in the knowledge, i.e., possibly faulty areas, we use an subgroup mining method that is well-known from machine learning research. The user is pointed to hot spots, i.e., recommendations for refinement given by a set of subgroup factors; these can then be considered for the refinement selecting from four basic refinement operators: Adapt/modify rules, extend knowledge, fix case, and exclude case. As a major point we propose introspective methods to support the interpretation of the discovered subgroups: The characteristic factors can be intuitively presented and ranked, and a subgroup can be exemplified in terms of its typical or extreme cases. Our refinement approach also includes the modification or elimination of used test cases, which we found reasonable if the test cases are taken from a real world application. Then, the assumption, that all test cases are correct, cannot always be made. Furthermore, we also emphasize the possibility of adding new (previously missing) knowledge to the system, which is important in the initial phase of development if the modeled knowledge is incomplete.

The rest of the paper is organized as follows: First we introduce subgroup mining and describe the subgroup-driven interactive refinement process. Then, we introduce two novel methods for introspective subgroup analysis. After that, we present a case study of the presented approach, discuss related work, and conclude with a summary.

Subgroup Mining

In this section, we introduce the used knowledge representation and describe the general subgroup mining approach.

General Definitions

Let Ω_A the set of all attributes with an associated domain of values $dom(a)$ for $a \in \Omega_A$. $\Omega_D \subseteq \Omega_A$ denotes the set of all diagnoses. $\mathcal{V}_A$ is defined as the (universal) set of attribute values (inputs) of the form $(a = v)$, $a \in \Omega_A, v \in dom(a)$. For each diagnosis $d \in \Omega_D$ we define a range $dom(d)$: $\forall d \in \Omega_D : dom(d) = \{established, not\ established\}$.

A diagnosis (output, solution) $d \in \Omega_D$ is derived by (heuristic) rules. A rule r for the diagnosis d can be considered as a triple $\big(cond(r), conf(r), d\big)$, where $cond(r)$ is the rule condition, $conf(r)$ is the confirmation strength. Such a rule $r = cond(r) \rightarrow d, conf(r)$ is used to derive the diagnosis d, where the rule condition $cond(r)$ contains conjunctions and/or disjunctions of (negated) attribute values (*findings*) $f_i \in \mathcal{V}_A$. The state of a diagnosis is gradually inferred by adding the confirmation strengths (points) of all the rules that have fired; if the sum is greater than a specific threshold value, then the diagnosis is assumed to be established.

Let CB denote the case base containing all available cases. A case $c \in CB$ is defined as a tuple $c = (\mathcal{V}_c, \mathcal{D}_c)$, where $\mathcal{V}_c \subseteq \mathcal{V}_A$ is the set of attribute values observed in the case c. The set $\mathcal{D}_c \subseteq \Omega_D$ is the collection of diagnoses describing the *solution* of this case.

Basic Subgroup Mining

Subgroup mining (Klösgen 2002) is a method to discover "interesting" subgroups of cases, e.g., "smokers with a positive family history are at a significantly higher risk for coronary heart disease". A subgroup mining task mainly relies on the following four properties: the target variable, the subgroup description language, the quality function, and the search strategy. We will focus on binary target variables.

Subgroups are described by relations between independent (explaining) variables and a dependent (target) variable. A subgroup description $sd = \{e_1, e_2, \ldots, e_n\}$ 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, $a_i \in \Omega_A, V_i \subseteq dom(a_i)$. Ω_{sd} denotes the set of all possible subgroup descriptions.

A quality function estimates the interestingness of the subgroup mainly based on a statistical test. It is used by the search method to rank the discovered subgroups during search. Formally, a quality function $q : \Omega_{sd} \times \mathcal{V}_A \rightarrow R$ evaluates a subgroup description $sd \in \Omega_{sd}$ given a target variable $t \in \mathcal{V}_A$. Several quality functions are proposed, for example in (Klösgen 2002). A classic quality function is the binomial test that is applicable for binary target variables,

$$q_{BT} = \frac{p - p_0}{\sqrt{p_0 \cdot (1 - p_0)}} \cdot \sqrt{n} \cdot \sqrt{\frac{N}{N - n}}\,,$$

where p is the relative frequency of the target variable in the subgroup, p_0 is the relative frequency of the target variable in the total population, N is the size of the total population, and n denotes the size of the subgroup. The quality function takes into account both the size of the subgroup and its the deviation from the total population.

An efficient search strategy is necessary for subgroup mining, since the search space is exponential concerning all possible selection expressions. We apply a modified beam search method, for which a subgroup description can be selected as an initial value for the beam.

Statistical Characterization of Subgroups Subgroups can always be characterized by the factors used to describe them, i.e., by the selectors contained in the subgroup description. However, beside these *principal factors* there are certain *supporting factors*, c.f., (Gamberger *et al.* 2005): These are attribute values $supp \subseteq \mathcal{V}_A$ contained in the subgroup that are identified using basic statistical analysis. The value distributions of their corresponding (supporting) attributes differ significantly comparing the true positive (target class) cases contained in the subgroup and non-target class cases contained in the total population.

We say, that an attribute value $(a = v)$ of a supporting attribute is characteristic for the subgroup, i.e., it is a supporting factor, if it is positively associated with the true positive cases contained in the subgroup compared to all the negative cases. For testing the statistical significance of an attribute and an attribute value we apply the standard χ^2-test for independence with a 0.05 significance level (i.e., with a confidence level of 95%), and the correlation- or ϕ-coefficient for binary variables, respectively.

Subgroup-Driven Interactive Refinement

In this section, we describe the process for interactive knowledge refinement, and present the subgroup method that provides potential faulty factors, i.e., recommendations for refinement.

The Process of Interactive Knowledge Refinement

For subgroup mining we consider a binary target variable corresponding to a diagnosis d, that is true (established) for incorrectly solved cases. We try to identify subgroups with a high share of this "error" target variable. We distinguish different *error analysis states* relating to the measures *false positives* FP (a diagnosis is falsely predicted), *false negatives* FN (a diagnosis is falsely *not* predicted) and the total error ERR combining both false positives and false negatives. Then, the potential faulty factors consist of the *principal factors* contained in the subgroup description and the *supporting factors*.

The subgroup-driven interactive refinement process mainly consists of seven steps: (1) We consider a diagnosis $d \in \Omega_D$, and select an analysis state $e \in \{FP, FN, ERR\}$. (2) A set of subgroups $\mathcal{SGS}_e$ is mined, either interactively by the domain specialist, or automatically by the system. Then, for each subgroup $\mathcal{SG}_i \in \mathcal{SGS}_e$ a set of *potential faulty factors* PFF_i is retrieved. (3) This set PFF_i is interpreted by the domain specialist. (4) If needed, subgroup introspection methods are applied in order to support the interpretation of PFF_i. (5) Based on the interpretation and analysis of PFF_i *guilty* (faulty) elements in the knowledge base or the case base are identified, and appropriate modification steps are applied. Then, the solutions of each case in the case base are recomputed. (6) The (changed) state of the system is assessed: The analysis measure e is checked for improvements. (7) If necessary, the process is iterated.

Refinement operators can either modify the knowledge base or the used case base. The knowledge base is usually adapted in order to fit the available correct cases. The case base is adapted, if particular cases are either wrong or they denote an extraordinary, exceptional state, which should not be modeled by the knowledge base. If the expert decides that the subgroup descriptions are reasonable (valid), then the knowledge base needs to be corrected. Otherwise, if they are not meaningful, then this can imply that the contained cases need corrections. We propose the following refinements:

- **Adapt/modify rules:** generalize or specialize conditions and/or rule actions. This operator is often appropriate if only one selector is contained in valid subgroups.
- **Extend knowledge:** add missing relations to the knowledge base. This operator is often applicable for at least two factors with a meaningful dependency relation.
- **Fix case:** correct the solution of a single case, or correct the findings of a case, if the domain specialist determines that the case has been labeled with the wrong solution.
- **Exclude case:** exclude a case from the analysis. If the setting of the case cannot be explained by factors accounted for by the knowledge base, e.g., by external decisions, then the case should be removed.

Examples of the application of the refinement operators are given in the case study below.

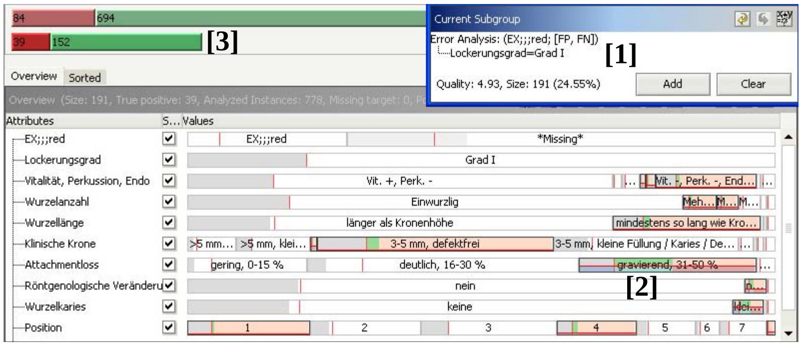

Figure 1: Visualizing Subgroups and Interesting Factors

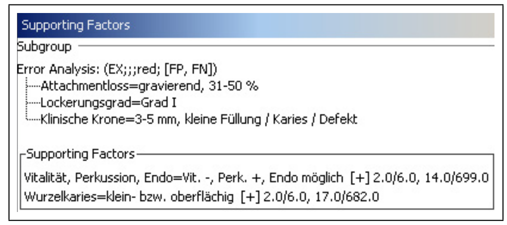

Figure 2: Supporting Factors

Visualizing Subgroups and Interesting Factors

An interactive refinement approach typically is not reasonable, if the user is not supported by visualization techniques, since the refinement space is usually too large. Therefore, we provide visualization methods for interactively browsing and testing subgroup hypotheses.

An exemplary visualization from the medical domain is shown in Figure 1, where the distributions of several factors are given. The subgroup *toothlax = minor (Lockerungsgrad = Grad I)* (Annotation 1) is shown with 39 incorrectly solved cases and 152 correctly solved cases; the general population contains 84 incorrectly and 694 correctly solved cases (Annotation 3). The rows in the table below the subgroup show the value distributions of the other attributes. Labels with a large 'dark-gray' sub-label, or a vertical bar that is close to the top, indicate "interesting" attribute values. The size of the 'dark-gray' sublabel relates to the share of the target variable in the subgroup. In the example visualization the cell *attachmentloss = strong (Attachmentloss = gravierend, 31-50%)* is the best one considering its size, and also the target share (Annotation 2). In this visualization the user is able to inspect different subgroups directly by one click on the corresponding cells. All elements, i.e., subgroups, rules, and cases, can be browsed directly by one click, and changes can be traced immediately. The changes are also intuitively reflected by the size of the bars (Annotation 3). Therefore, the user-guided integrated method provides direct interaction and instant feedback to the user.

Figure 2 shows an exemplary list of the supporting factors for the subgroup *toothlax = minor (Lockerungsgrad = Grad I) AND attachmentloss = strong (Attachmentloss = gravierend, 31-50%) AND clinical crown = 3-5mm/Caries/Defect (Klinische Krone=3-5mm, kleine Füllung/Karies/Defekt)*. The supporting factors include the descriptor *endodontic state = possible (Vitalität, Perkussion, Endo=Vit. -, Perk +, Endo möglich)* with the weak score (+) and the descriptor *root caries=minor or on surface (Wurzelkaries = klein bzw. oberflächig)* also with a weak score (+). These two factors both occur in 2 of the 6 positive subgroup cases. The scores represent the strength of the supporting factors that provide an intuitive overview of other significant factors in the subgroup. A method to derive these confirmation strengths will be introduced below.

Subgroup Introspection

As outlined above, the results of the subgroup mining step are a collection of subgroups which are used to derive a set of potential faulty factors PFF (principal and supporting factors). These are then proposed for refinement. For example, consider the subgroup "smokers with a positive family history are at a significantly higher risk for coronary heart disease": the principal factors consist of *smoker=true* and *family history=positive*, and the potential supporting factors could be *hypertension=true* and *overweight=true*. The interpretation of PFF depends on the judgment of the user, especially on his/her existing background knowledge.

The principal factors can be regarded as *strong* factors that occur in all cases of the subgroup, while the supporting factors can be regarded as *weak* factors that occur only in some cases. As discussed by (Gamberger *et al.* 2005) presenting the supporting factors can be very important since they can provide additional evidence similar to naive Bayes.

In the following, we describe two methods for subgroup introspection. The first approach obtains the factors that statistically characterize a subgroup and assesses the individual strength of the characteristic factors in the sub-population defined by the incorrectly solved, i.e., the target class cases of the subgroup. The second method aims to exemplify a subgroup using the subgroup extension, i.e., the set of cases covered by the subgroup. Then, typical or extreme cases of the subgroup can be presented to the user in order to provide distinctive examples of the subgroup objects.

Introspecting Scored Subgroup Factors

In the following we will discuss how to characterize the subgroup by its principal factors and the supporting factors; we propose a technique for presenting the supporting factors in an intuitive way using symbolic diagnostic scores. The characteristic subgroup factors are presented with an assigned strength corresponding to the evidence they provide for the target concept.

Scoring Subgroup Factors A score is simple to interpret and one of the standard knowledge formalization formats, e.g., in the medical domain diagnostic relations are often modeled as diagnostic scores (Puppe 1998). In general, scores consist of a set of factors with assigned symbolic categories. This representation is very suitable to be used for characterizing subgroups by the set of supporting factors: the symbolic categories of the factors contained in the score denote the relative importance, or the strength of the individual scoring relations, i.e., the relation between the factor and the target variable. For each factor (selector) e we construct a scoring selector $e' = (e, sc)$ assigning a confirmation category contained in the set $sc \in \{sc_1, sc_2, sc_3\}$ that specifies confirming symbolic categories in ascending order. So, the symbolic category sc expresses the strength or the relative importance of the observation of a given selector e.

For rating the subgroup factors contained in the set PFF concerning their confirmation strengths, we compare two populations: The true positive contained in the subgroup and the false positives of the total population. In this way we identify how significantly a selector can discriminate between the cases containing the target concept in the subgroup, and all remaining non-target class cases. It is easy to see that the principal factors will always obtain the strongest confirmation category, while the weaker categories will be assigned to the supporting factors.

To compute scores we can utilize a method presented in (Atzmueller, Baumeister, & Puppe 2006). Adapting it to our refinement task, we construct a 2×2 contingency table comparing the distribution of the supporting factor of the true positives in the subgroup, i.e., the target class cases, vs. all negative cases. If the association is significant, then we compute a quasi-probabilistic score according to the strength of the association utilizing the ϕ-coefficient. This score is then mapped to a symbolic confirmation category $sc \in \{sc_1, sc_2, sc_3\}$ using a suitable conversion table.

Discussion By characterizing a given subgroup by its principal and supporting factors we obtain more evidence for the target variable within the subgroup. If the supporting factors are ranked and are assigned a score, then the user can get a comprehensive and intuitive overview of the statistically significant factors: The principal factors are the most important factors describing the subgroup while the supporting factors are used to statistically characterize the sub-population defined by the positive cases of the subgroup.

For the refinement task, the ranked factors can provide direct feedback for applying a refinement operator. If the factor has a high confirmation category, and the false-positives of a diagnosis are considered, then the factor is a candidate for the "adapt/modify rule" operator, i.e., reducing its strength. Furthermore, principal factors combined with strong supporting factors can be good candidates for very specific complex rules, by applying the "extend knowledge" operator.

Subgroup Introspection by Exemplification

To support the user in the interpretation of the potential faulty factors PFF, we propose to utilize the implicit experiences contained in the cases of the case base as explaining examples. Then, typical and extreme cases with a high coverage of the set of describing factors PFF can be retrieved for presentation to the user.

Retrieving Exemplary Cases A naive solution retrieves all cases contained in the subgroup that are also containing the target concept. However, this approach suffers from two shortcomings: First, the set of cases can be quite large for a comprehensive overview, and second a subset of PFF is not taken into account very precisely, i.e., the supporting factors. Therefore, we aim to retrieve a set of cases that have a high coverage with the set PFF. Then, we have two options to characterize the elements of PFF: First we can retrieve *typical* cases that are highly similar to PFF while the individual cases can also be very similar to each other. These cases can be used to exemplify the most common factors contained in PFF. Second, we can retrieve *extreme* cases, i.e., cases that are very similar to PFF but not to each other. This set of diverse cases is discriminative but still similar to PFF and can be used to get a comprehensive description of extreme factor combinations concerning PFF.

For the retrieval step we use techniques known from *case-based reasoning* (Aamodt & Plaza 1994). Here, given a query case q the general goal is to retrieve a set of most similar cases $\{c_i\}$. The attribute values contained in the query case are commonly called the *problem description*. We construct a *virtual* query case q and define its problem description as the set of potential faulty factors PFF_i obtained from a given subgroup $\mathcal{S}G_i$. Optionally, the user can select a subset of factors contained in PFF_i, e.g., focusing on the most *interesting* factors, or can include the target variable such that specific queries can be formulated. Thus, the factors of the query case can be interactively adapted to fit the analysis requirements of the user.

For assessing the similarity of the query case q and a retrieved case c, we can use e.g., the well-known *matching features* similarity function. For case comparison the set of attributes is restricted to the attributes contained in the query (w.r.t. PFF_i), i.e., to the attributes $\Omega'_A = \{a \mid \exists v \in PFF_i, v \in dom(a)\}$; $\pi_a(c)$ returns the value of attribute a:

$$sim(q, c) = \frac{|\{a \in \Omega'_A : \pi_a(q) = \pi_a(c)\}|}{\Omega'_A} \qquad (1)$$

The diversity of a set of retrieved cases $\mathcal{R}C = \{c_i\}_k$ of size k is measured according to the measure $diversity(\mathcal{R}C)$, defined as follows:

$$diversity(\mathcal{R}C) = \frac{\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \big(1 - sim(c_i, c_j)\big)}{k \cdot (k-1)/2}\ , \qquad (2)$$

where the similarity of two cases is computed with respect to the attributes in the constructed query case q.

To retrieve the set of most extreme cases we apply techniques that obtain a set of most similar but diverse cases R w.r.t the query case. There are several methods to retrieve a set of diverse cases, c.f., (McSherry 2002). We apply the *Bounded Greedy (BG)* algorithm: BG starts with a retrieval set initially containing the most similar case to the query case. In each iteration of the algorithm the case of the set of $2k$ most similar cases is selected that maximizes the product of its similarity to the query case and its relative diversity w.r.t. the cases that have been selected for the retrieval set so far. The relative diversity $relDiversity(c, RC)$ of a case c w.r.t. the retrieval set $\mathcal{R}C = \{c_i\}_m$ of size m is defined as

$$relDiversity(c, RC) = \frac{\sum_{i=1}^{m} 1 - sim(c, c_i)}{m} \tag{3}$$

BG stops if the retrieval set reaches its pre-specified size of k. Then the set of diverse cases can be presented to the user. To obtain a smaller number of diverse (extreme) cases, we can optionally select the smallest subset $R' \subseteq R$ where the coverage between the problem description of a query case q and the union of the problem descriptions contained in R' is maximized. The retrieved set of typical (or extreme) cases is then presented to the user as a set of explaining examples for the given set of potential faulty factors characterizing a specific subgroup.

Discussion The exemplification approach for subgroup introspection described above provides the option for further exploratory analysis of a specific subgroup. By presenting typical or extreme cases, the user obtains an intuitive impression about the objects (cases) contained in the subgroup. Then, the cases can be analyzed in-depth, e.g., with regard to incorrect findings or an incorrectly assigned solution if a subgroup is not meaningful to the domain specialist.

Besides inspecting discovered subgroups the exemplification technique can also be used for summarizing certain cases, i.e., if a subgroup is constructed with the special goal of obtaining an overview of the contained cases. The presented approach is an alternative to the primary description of a subgroup by its principal factors. For example, by inspecting the set of diverse cases the domain specialist can obtain a comprehensive overview of the general *problem setting* that is manifested within a certain subgroup containing a significant share of incorrectly solved cases.

Case Study

We performed a case study of the proposed interactive refinement method, and already presented first initial results in (Atzmueller *et al.* 2005), without applying the introspection methods. Then, based upon the obtained experiences and by utilizing the methods for subgroup introspection we were able to further improve the knowledge system.

The case study was implemented in the medical domain with a consultation and documentation system for dental findings regarding any kind of prosthetic appliance, which is currently being extended. The systems aims to decide about a diagnostic plan using the clinical findings: For decision support the system derives two distinct diagnosis *EX* and *IN* that either indicate the teeth that could be conserved (IN) or should be extracted (EX). The cases always contain the standard anamnestic findings and additional findings from x-ray examinations, e.g., abnormal x-ray findings (apical, periradicular), grade of tooth lax, endodontic state (root filling, pulp vitality), root quantity, root length, crown length, level of attachment loss, root caries, tooth angulation and elongation/extrusion.

The used case base contained 778 cases corresponding to 778 examined teeth. We investigated the diagnosis referring to tooth extraction/non extraction. Initially, the case base contained 108 falsely solved cases (as evaluated by a domain specialist). In the first phase of the case study described in (Atzmueller *et al.* 2005) we managed to reduce the number of incorrectly solved cases from 108 to 54 by 50%.

The domain specialist assessed several subgroups mined by the system as significant, which were then used for knowledge base refinement. We modified and added several rules, some examples are given in Table 1. Subgroup description #1 is an example for a simple modification. For *abnormal x-ray = only apical* we modified the score, such that the rule only contributes 5 points. For the following two subgroup descriptions the corresponding rules exemplify two general mechanisms: In rule #2 the condition *root length = longer than crown length* counts as negative for extraction, and relativizes the factor *tooth lax = medium* which is positive for extraction. The relativization can also work the other way round, i.e., when a positive factor relativizes a negative one. Then, for extraction, we would have to add points, e.g., for *tooth lax = medium* and *attachmentloss = minor*. For subgroup description #3 the selectors *tooth lax = minor* and *attachmentloss = strong* are both positive for extraction, but since they are assessed independently in the rule base they should not be over-emphasized by being counted twice. Therefore, the score points of the corresponding rules were decreased.

During the case study especially the interactive part of the method was very well accepted by the domain specialist, who was supported by the presented visualization methods. Furthermore, the domain specialist considered it very helpful and important to stay in full control of the refinements during the steps of the refinement process.

The experiences obtained throughout the first part of the case study motivated the development of further methods for subgroup introspection, since the remaining 54 incorrectly solved cases contained very small subgroups of erroneous cases. Thus, the "hot spots" needed to be analyzed in detail, either statistically or by viewing the detailed cases. Both the presentation of characterizing subgroup factors and exemplifying cases were key features for the domain specialist, who performed the analysis. Furthermore, the exemplification method allowed for a comprehensive overview on the sub-population defined by a small set of exemplary cases which was very helpful throughout the analysis.

Examples for further rules are given by the subgroup descriptions #4 and #5. These were observed in small sub-populations, and therefore the introspection techniques proved highly useful in determining and validating such relations. Rule #4 is similar to rule #1 while it was observed in a significantly smaller number of cases. The highly specific rule #5 is an exception rule similar to rule #2: Factor *tooth lax = none* observed in that situation is a strong factor negative for extraction. So far, we were able to improve the knowledge base by reducing the number of incorrectly solved cases by more than 50% from a total of 108 to 42. Thus, we were able to improve the precision of the knowledge base from 86% to 95%.

No.	*Subgroup Description*	*Diagnosis*	*Points*
1	abnormal x-ray = only apical	EX	10 $\rightarrow$ 5
2	tooth lax = medium $\wedge$ root length = longer than crown length	EX	-20
3	tooth lax = minor $\wedge$ attachmentloss = strong	EX	-20
4	root caries = minor or on surface	EX	10 $\rightarrow$ 5
5	tooth lax = none $\wedge$ attachmentloss = strong $\wedge$ endodontic state = possible	EX	-10

Table 1: Examples of discovered subgroups and according refinements

Related Work

In the past, various approaches for (automatic) knowledge refinement were proposed, e.g. (Ginsberg 1988; Knauf *et al.* 2002; Carbonara & Sleeman 1999). However, all automatic methods depend on the *tweak assumption* (Carbonara & Sleeman 1999), which implies that the knowledge base is almost valid and only small improvements need to be performed. In the case study described above the validity of the knowledge base was quite poor (about 86% accuracy) and therefore no tweak assumption could be made. In contrast, we expected that important rules were missing and that we have to acquire additional knowledge during the process. For this reason, we decided to choose a mixed refinement/elicitation process, which emphasizes the interactive analysis and modification of the implemented rules based on found subgroup patterns. Similarly, (Carbonara & Sleeman 1999) use an inductive approach for generating new rules using the available cases. (Diamantidis & Giakoumakis 1999) describe a framework for refinement by inductively creating a new knowledge base using incorrectly solved cases annotated with justifying explicit explanations by experts. (Kelbassa & Knauf 2005) also describe an approach supplementing formal methods with domain knowledge. However, in our application we cannot expect that all cases contain the correct solution, while automatic approaches mainly do assume a correct case base. Therefore a thorough analysis of the cases within the process was also necessary. Then, the user is supported by the interactive approach and the introspection strategies in order to obtain a comprehensive and intuitive view of the subgroups and their descriptive factors.

Summary and Future Work

In this paper we presented an interactive approach for the refinement of rule-based knowledge. In contrast to classical (automatic) approaches the user has to decide about the actual refinement operators to be carried out, and is strongly supported by the identification of hot spots that can be analyzed in detail using introspective subgroup analysis methods. In the future, we plan to investigate further (automatic) refinement techniques to support the user, e.g., a semi-automatic refinement method that is adapted to the used knowledge representation (rules with point scores).

Acknowledgements

We want to thank Achim Hemsing and Prof. Ernst-Jürgen Richter from the department of prosthodontics at the Würzburg University Hospital for their medical expertise and analysis while performing the case studies of this research project.

References

Aamodt, A., and Plaza, E. 1994. Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7(1):39 – 59.

Atzmueller, M.; Baumeister, J.; and Puppe, F. 2006. Semi-Automatic Learning of Simple Diagnostic Scores utilizing Complexity Measures. *AI in Medicine* in press.

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

Boswell, R., and Craw, S. 1999. *Validation and Verification of Knowledge Based Systems*. Oslo, Norway: Kluwer. chapter Organizing Knowledge Refinement Operators, 149–161.

Carbonara, L., and Sleeman, D. 1999. Effective and Efficient Knowledge Base Refinement. *Machine Learning* 37:143–181.

Diamantidis, N. A., and Giakoumakis, E. A. 1999. An Interactive Tool for Knowledge Base Refinement. *Expert Systems* 16(1):2 – 10.

Gamberger, D.; Krstacic, A.; Krstacic, G.; Lavrac, N.; and Sebag, M. 2005. Data Analysis Based on Subgroup Discovery: Experiments in Brain Ischaemia Domain. In *Proc. 10th Intl. Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2005)*, 52–56.

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

Kelbassa, H.-W., and Knauf, R. 2005. A Process Approach to Rule Base Validation and Refinement. In *Proc. Workshop on Knowledge Engineering and Software Engineering at 28th German Conf. on AI*, 25–36. Koblenz, Germany.

Klösgen, W. 2002. *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press, New York. chapter 16.3: Subgroup Discovery.

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

McSherry, D. 2002. Diversity-Conscious Retrieval. In *Proc. 6th European Conference on Advances in Case-Based Reasoning*, 219–233. London, UK: Springer.

Puppe, F. 1998. Knowledge Reuse among Diagnostic Problem-Solving Methods in the Shell-Kit D3. *Intl. Journal of Human-Computer Studies* 49:627–649.