Exemplifying Subgroup Mining Results for Interactive Knowledge Refinement

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

Abstract: When intelligent 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. In this paper, we present a novel user-guided approach for the refinement of knowledge bases using subgroup mining methods. Additionally, we describe a technique that helps the user to interpret the refinement recommendations proposed by the system.

1 Introduction

The refinement of intelligent systems is an essential task for the implementation and maintenance of systems deployed into a real-world applications. When the knowledge base is built manually, then typically refinements are necessary throughout the initial deployment phase. Often, the domain specialists face the problem that the developed knowledge base is still incomplete. In consequence, important extensions and not only modifications of the knowledge have to be conducted 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., [Gin88, BC99, CS99, KPJ+02]. 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. Within this interactive approach the user is pointed to hot spots and has to decide about four basic refinement operators: Adapt/modify knowledge, extend knowledge, fix case, and exclude case. We see that 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 the development phase if the modeled knowledge is incomplete. Moreover, we propose a method to exemplify the refinement recommendations provided by the system using exemplary cases.

The rest of the paper is organized as follows: In Section 2 we introduce subgroup mining and describe the subgroup-driven interactive refinement process in Section 3. Finally, we conclude the paper in Section 4 with a summary and some future work.

2 Subgroup Mining

In this section, we first introduce the used knowledge representation; then, we briefly describe the subgroup mining approach.

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

A diagnosis $d \in \Omega_D$ 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. Thus 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) findings $f_i \in \mathcal{V}_A$. The state of a diagnosis is gradually inferred by summing all the confirmation strengths (points) of 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 set of diagnoses describing the *solution* of this case.

Basic Subgroup Mining Subgroup mining [Klö02] 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 $sd = \{e_i\}$ 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 measures the interestingness of the subgroup mainly based on a statistical evaluation functions. 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 [Klö02]. The exemplary quality function $q_{BT} = \frac{p-p_0}{\sqrt{p_0 \cdot (1-p_0)}} \sqrt{n} \sqrt{\frac{N}{N-n}}$, is applicable for binary target variables, 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.

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, where a subgroup description can be selected as an initial value for the beam.

3 The Subgroup-Driven Interactive Refinement Process

In this section, we describe the process for interactive knowledge refinement We present the method that describes potential faulty factors, i.e., recommendations for refinement using cases contained in the case base with their context. Finally, we discuss related work.

Subgroup Mining for the Refinement Task 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. However, we need to 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, for all cases contained in the case base. Then, the potential faulty factors consist of the *principal factors* contained in the subgroup description and the *supporting factors*. These are findings $supp \subseteq \mathcal{V}_A$ contained in the subgroup, which are characteristic for the subgroup, i.e., the value distributions of their corresponding attributes (supporting attributes) differ significantly comparing two populations: the target class cases contained in the subgroup and non-target class cases contained in the total population.

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, (typical) exemplary cases for PFF_i are retrieved. (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 restarted.

Refinement operators can either modify the knowledge base or the applied 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. In summary, the following refinements can be performed:

- **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.

Exemplifying Subgroup Mining Results As outlined above, the results of the subgroup mining step are a set of subgroups which are used to derive a set of potential faulty factors PFF (principal and supporting factors). These are then presented and 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*, *overweight=true*, *age>50*. As outlined above, the interpretation of PFF depends on the judgment of the user, especially on his/her existing background knowledge. To support the user, we propose to utilize the implicit experiences contained in the cases of the case base as explaining examples for PFF. Then, typical and extreme cases with a high coverage of the set of PFF can be retrieved for presentation to the user. These cases contain "real-world" experience, and additional factors that are related to PFF. These factors can potentially help to further support refinement decisions.

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 accounted for 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 PFF elements: 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* [AP94]. 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 $\mathcal{SG}_i$. Optionally, the user can define a subset of factors contained in PFF_i, e.g., concentrating on the most *interesting* factors such that specific queries can be formulated. The factors of the constructed query case can be interactively adapted to fit the analysis task at hand.

For assessing the similarity of a query q and a retrieved case c, e.g., we can use the well-known *matching features* similarity function. Then, 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:

$$sim(q,c) = \frac{|\{a \in \Omega'_A : \pi_a(q) = \pi_a(c)\}|}{\Omega'_A} \quad diversity(\mathcal{RC}) = \frac{\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \Big(1 - sim(c_i, c_j)\Big)}{k \cdot (k-1)/2}$$

The diversity of a set of retrieved cases $\mathcal{RC} = \{c_i\}_k$ of size k is measured according to $diversity(\mathcal{RC})$ where the similarity of two cases is assessed with respect to the attributes in the constructed query case q, as outlined above. Then a set of most similar diverse cases

R to this query is retrieved as described, e.g., in [McS02]. 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 potential faulty factors characterizing a specific subgroup.

Case Study We already conducted a case study of the presented interactive refinement process, c.f., [ABH$^+$05] (without the exemplification method). The case study was implemented in the medical domain with a consultation and documentation system for dental findings regarding any kind of prosthetic appliance, which is currently being extended. The systems aims to decide about a diagnostic plan using the clinical findings: for decision support the system derives two distinct diagnosis *EX* and *IN* that either indicate the teeth that could be conserved (IN) or should be extracted (EX). The cases always contain the standard anamnestic findings and additional findings from x-ray examinations, e.g., abnormal x-ray findings (apical, periradicular), grade of tooth lax, endodontic state (root filling, pulp vitality), root quantity, root length, crown length, level of attachment loss, root caries, tooth angulation and elongation/extrusion.
The applied case base contained 778 cases corresponding to 778 examined teeth. We investigated the diagnosis referring to tooth extraction/non extraction. Initially, the case base contained 108 falsely solved cases (as evaluated by a domain specialist). Using the proposed refinement process we managed to reduce the number of incorrectly solved cases from 108 to 54 by 50%, increasing the precision of the knowledge base from 86% to 93%. Especially the interactive part of the method was very well accepted by the domain specialist, who was supported by visualization methods (c.f. [ABH$^+$05] for more details). This case study and the experiences we obtained motivated the method to exemplify subgroups and their describing factors as presented above. We are planning to evaluate the usefulness of these mechanisms in a case study in the near future.

Discussion In the past, various approaches for (automatic) knowledge refinement were proposed, e.g. [Gin88, KPJ$^+$02, CS99]. However, all automatic methods depend on the *tweak assumption* [CS99], which implies that the knowledge base is almost valid and only small improvements need to be performed. In the case study briefly 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 and Sleeman [CS99] use an inductive approach for generating new rules using the available cases. Diamantidis and Giakoumakis [DG99] describe a framework for refinement by inductively creating a new knowledge base using incorrectly solved cases annotated with justifying explicit explanations by experts. 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. Thus, the user is supported by the interactive approach and the exemplification strategy for subgroups and their descriptive factors.

4 Summary and Future Work

In this paper we introduced 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, but is strongly supported by the indication and exemplification of hot spots that are identified by a subgroup mining method. The interactive refinement approach has been already evaluated using a medical knowledge system that is currently extended and used in a real-world application. Due to the experiences made with the interactive refinement process we developed the exemplification capabilities proposed in this paper. This method can be potentially be extended using background knowledge, e.g., to split the problem descriptions into partially disjunctive partitions corresponding to certain problem areas. Then partial cases for these partitions can be retrieved and recombined, as described in [ABP03]. In the near future we are planning to evaluate the usefulness of the presented approach within an extended case study.

References

[ABH+05] Martin Atzmueller, Joachim Baumeister, Achim Hemsing, Ernst-Jürgen Richter, and Frank Puppe. Subgroup Mining for Interactive Knowledge Refinement. In *Proc. 10th Conference on Artificial Intelligence in Medicine (AIME 05)*, Aberdeen, Scotland, 2005.

[ABP03] Martin Atzmueller, Joachim Baumeister, and Frank Puppe. Evaluation of two Strategies for Case-Based Diagnosis handling Multiple Faults. In *Proc. 2nd Confernence of Professional Knowledge Management (WM2003)*, Luzern, Switzerland, 2003.

[AP94] Agnar Aamodt and Enric Plaza. Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1), 1994.

[BC99] Robin Boswell and Susan Craw. *Validation and Verification of Knowledge Based Systems*, chapter Organizing Knowledge Refinement Operators, pages 149–161. Kluwer, Oslo, Norway, 1999.

[CS99] Leonardo Carbonara and Derek Sleeman. Effective and Efficient Knowledge Base Refinement. *Machine Learning*, 37:143–181, 1999.

[DG99] Nikolaos A. Diamantidis and E. A. Giakoumakis. An Interactive Tool for Knowledge Base Refinement. *Expert Systems*, 16(1):2 – 10, 1999.

[Gin88] Allen Ginsberg. *Automatic Refinement of Expert System Knowledge Bases*. Morgan Kaufmann, 1988.

[Klö02] Willi Klösgen. *Handbook of Data Mining and Knowledge Discovery*, chapter 16.3: Subgroup Discovery. Oxford University Press, New York, 2002.

[KPJ+02] Rainer Knauf, Ilka Philippow, Klaus P. Jantke, Avelino Gonzalez, and Dirk Salecker. System Refinement in Practice – Using a Formal Method to Modify Real-Life Knowledge. In *Proc. 15th International Florida Artificial Intelligence Research Society Conference (FLAIRS-2002)*. AAAI Press, 2002.

[McS02] David McSherry. Diversity-Conscious Retrieval. In *Proc. 6th European Conference on Advances in Case-Based Reasoning*, pages 219–233, London, UK, 2002. Springer.