Diagnosis and Redundancy Detection in PeopleViews

Alexander Felfernig
Applied Software Engineering
Inffeldgasse 16b/II
A-8010, Graz, Austria
alexander.felfernig@ist.tugraz.at

Martin Stettinger
Applied Software Engineering
Universitätsstraße 65-67
A-9020, Klagenfurt, Austria
martin.stettinger@ist.tugraz.at

Stefan Reiterer
Applied Software Engineering
Universitätsstraße 65-67
A-9020, Klagenfurt, Austria
stefan.reiterer@ist.tugraz.at

Müslüm Atas
Applied Software Engineering
Inffeldgasse 16b/II
A-8010, Graz, Austria
muatas@ist.tugraz.at

Seda Polat Erdeniz
Applied Software Engineering
Inffeldgasse 16b/II
A-8010, Graz, Austria
spolater@ist.tugraz.at@ist.tugraz.at

Trang Tran
Applied Software Engineering
Inffeldgasse 16b/II
A-8010, Graz, Austria
ttrang@ist.tugraz.at

Amal Shehadeh
Applied Software Engineering
Inffeldgasse 16b/II
A-8010, Graz, Austria
ashehade@ist.tugraz.at

ABSTRACT

The PEOPLEVIEWS project focuses on the provision of intelligent techniques that support the creation of recommender knowledge bases on the basis of mechanisms from the area of Human Computation. In this context, the project focuses on the development of constraint-based recommender systems where recommendations are determined on the basis of an underlying knowledge base. The approach to engage domain experts more deeply in knowledge engineering processes helps to relieve knowledge engineers from routine tasks and to tackle the knowledge acquisition bottleneck which is a major reason for the non-scalability of knowledge-based (recommendation) technologies. In this paper we focus on a specific aspect which is the support of PEOPLEVIEWS users (a) interacting with PEOPLEVIEWS (if no solution can be identified) and (b) when building knowledge bases (to identify redundancies and to identify faulty constraints).

Author Keywords

Recommender Systems; Constraint-based Recommenders; Human Computation; Model-based Diagnosis

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation

INTRODUCTION

The overall goal of PEOPLEVIEWS [4][1] is to support the development and maintenance of constraint-based recommender applications on the basis of Human Computation technologies [8]. The outcome of a PEOPLEVIEWS development process is a knowledge base that can be used within the scope of recommendation processes. Example knowledge bases resulting from a PEOPLEVIEWS development process are financial services, digital cameras or tourist destinations.

Constraint-based recommenders are primarily used in scenarios where constraints exist between different different properties of an item. Such items are often of type "high-involvement" where sub-optimal decisions are associated with a higher degree of risk (e.g. house, apartment, expensive technical equipment, group holidays, and software solutions).

When dealing with constraints, inconsistent situations can occur. For example, if a user enters a set of requirements it cannot be guaranteed that a solution can be identified for these requirements. For a given set of test cases, it cannot be guaranteed that these are accepted by the current version of the knowledge base. Furthermore, knowledge bases can be become redundant, i.e., there are constraints in the knowledge base which do not further restrict the overall solution space. A constraint $c_\rho \in C$ is redundant if $C - \{c_\rho\} \models c_\rho$.

For an overview of the different recommendation approaches integrated in the PEOPLEVIEWS environment and also an overview on related work in the area of Human Computation

[1]The work presented in this paper has been conducted within the scope of the PEOPLEVIEWS project funded by the Austrian Research Promotion Agency (843492).

and recommender systems we refer to [5, 7]. The remainder of this paper is organized as follows. In Section *Diagnosis of Inconsistent Requirements* we show which process was chosen to support users in situations where no solution can be identified by a PEOPLEVIEWS recommender. In Section *Intra-Constraint Diagnosis* we sketch an approach to make diagnosis more granular, i.e., allowing diagnosis approaches to return parts (individual expressions) of a constraint. In Section *Intra-Constraint Redundancy Detection in Filter Constraints* we show how redundant expressions on the left hand side of a PEOPLEVIEWS filter constraint can be identified. In Section *Redundancy Detection in Conjunctive Normal Form* we shortly explain the basic approach to transform general constraint structures to a representation "understandable" by the diagnosis algorithm.

DIAGNOSIS OF INCONSISTENT REQUIREMENTS

The diagnosis of inconsistent requirements is handled in PEOPLEVIEWS similar to existing constraint-based recommender approaches [1].

Recommendation Task. A recommendation task is defined in terms of a Constraint Satisfaction Problem (CSP) $(V, D, C \cup R)$ where $V = \{v_1, v_2, .., v_n\}$ represents a set of variables, $D = \{dom(v_1), dom(v_2), .., dom(v_n)\}$ represents a set of domain definitions, and $C = \{c_1, c_2, .., c_m\}$ represents a set of constraints. A set of customer requirements $R = \{r_1, r_2, .., r_k\}$ completes the definition of a recommendation task.

Diagnosis Task. A diagnosis task is defined by a recommendation task $(V, D, C \cup R)$ where $R \cup C$ is inconsistent.

Conflict Set. A conflict set $CS \subseteq R$ is a subset of the requirements where $CS \cup C$ is inconsistent.

Diagnosis. A diagnosis $\Delta \subseteq R$ for a given diagnosis task is a set of requirements where $R - \Delta \cup C$ is consistent. A diagnosis Δ is minimal if there does not exist a diagnosis Δ' such that $\Delta' \subset \Delta$.

Diagnosis determination of inconsistent user requirements in PEOPLEVIEWS is implemented on the basis of direct diagnosis techniques [2, 3] which are efficient enough to be used in interactive recommendation settings.

INTRA-CONSTRAINT DIAGNOSIS

Within the scope of PEOPLEVIEWS we have developed intra-constraint diagnosis approaches which help to identify sources of inconsistencies within individual constraints. In this context, we made the assumption of implicative forms, i.e., each constraint has the structure $a_1 \wedge a_2 \rightarrow b_1 \wedge b_2 \wedge b_3$. In order to sketch our approach, we introduce a simple knowledge base C and one corresponding test case t that helps to induce a conflict in the knowledge base.

- $V = \{u, v, x, y, z\}, dom(u) = dom(v) = dom(x) = dom(y) = dom(z) = \{1, 2\}$
- $C = \{c_1 : x = 1 \rightarrow y = 2, c_2 : x = 2 \wedge y = 2 \rightarrow z = 2, c_3 : x \leq 1 \rightarrow y \neq 2 \wedge z = 8\}$
- $t : x = 1$

Obviously, test case t induces a conflict in the knowledge base since $\{t\} \cup \{c_1\} \cup \{c_2\} \cup \{c_3\}$ is inconsistent. Standard diagnosis approaches such as [3] would determine two minimal diagnoses $\Delta_1 = \{c_1\}$ and $\Delta_2 = \{c_3\}$, i.e., c_1 or c_3 have to be deleted from C (or adapted) such that the remaining constraints are consistent with the (positive) test case t. Approaches based on conflict detection and hitting set calculation [6] would predetermine the induced conflict sets and at the same time construct a corresponding hitting set directed acyclic graph (HSDAG).

If we want to be more precise, we have to take a more detailed look at the constraints in C. For this purpose, we rewrite our example recommendation knowledge base as follows.

- $V = \{u, v, x, y, z\}, dom(u) = dom(v) = dom(x) = dom(y) = dom(z) = \{1, 2\}$
- $C = \{c_1 : x = 1 \rightarrow y = 2, c_2 : x = 2 \wedge y = 2 \rightarrow z = 2, c_{3a} : x \leq 1 \rightarrow y \neq 2, c_{3b} : x \leq 1 \rightarrow z = 8\}$
- $t : x = 1$

In this scenario, the original constraint c_3 is split into two different constraints c_{3a} and c_{3b}. The reason for doing this split is that the diagnosis approach can now focus on individual parts of the right hand side of a constraint. After this rewriting step, diagnosis determination can stay the same, i.e., we can exploit direct diagnosis approaches such as [3] or conflict-directed diagnosis search approaches [6]. This PEOPLEVIEWS rewriting approach can be applied to implicative structures a mentioned above, i.e., cannot be applied to arbitrary constraint structures. In our working example, $\Delta = \{c_{3a}\}$ would be a minimal diagnosis.

INTRA-CONSTRAINT REDUNDANCY DETECTION IN FILTER CONSTRAINTS

Similar to the identification of diagnoses in inconsistent constraint sets, redundancy detection in PEOPLEVIEWS relies on a basic assumption about constraint structures. The redundancy detection approach will as well be explained on the basis of an example. In the following, c_1 is a constraint directly and automatically generated from user feedback on PEOPLEVIEWS micro-tasks. In this example, the first conjunctive expression (1) and the last one (2) have an overlap in terms of the referrals to variables x and y. Furthermore, (1) can be seen as a specialization of (2) since every time (2) is active, (1) is active as well but not vice-versa. Consequently, (2) is a redundant subexpression that can be omitted since it's deletion does not change the semantics (the solution space) defined by the knowledge base. Again, this approach relies on the constraint structures used in PEOPLEVIEWS, i.e., it cannot be applied to more general constraint structures.

- $V = \{u, v, x, y, z\}, dom(u) = dom(v) = dom(x) = dom(y) = dom(z) = \{1, 2\}$
- $C = \{c_1 : (x = 1 \wedge y = 1) \vee (x = 2 \wedge y = 2) \vee (x = 1 \wedge y = 1 \wedge z = 1) \rightarrow u = 5.\}$

REDUNDANCY DETECTION IN CONJUNCTIVE NORMAL FORM

Every constraint can be transformed into a corresponding conjunctive normal form where the individual disjunctions can be regarded as basic components taken into account by the diagnosis process. This way, those individual elements of constraints can be identified that are responsible for a given inconsistency. The disadvantage of this approach is the need for rewriting constraints and the issue that the understandability of generated constraint structures is rather low.

COMMUNITY-BASED DIAGNOSIS

Human Computation approaches can also be exploited to receive community feedback on the quality of individual constraints. Other examples of community feedback are in which context which items should be recommended, in which way existing conflicts should be resolved, and in which way recommendation candidate items should be ranked. In the following we show a simple example of how constraint evaluations can be exploited to determine preferred diagnoses, i.e., those constraints which have a high probability of being responsible for an inconsistency. In order to explain the basic idea of community-based diagnosis, we introduce the following example.

- $V = \{u, v, x, y, z\}, dom(u) = dom(v) = dom(x) = dom(y) = dom(z) = \{1, 2\}$
- $C = \{c_1 : x > y, c_2 : x < y, c_3 : u > v, c_4 : u < v\}$

In this knowledge base, two conflict sets are responsible for the given inconsistency: $CS_1 = \{c_1, c_2\}$ and $CS_2 = \{c_3, c_4\}$. Up to now, we have four different diagnosis candidates which are $\Delta_1 = \{c_1, c_3\}$, $\Delta_2 = \{c_1, c_4\}$, $\Delta_3 = \{c_2, c_3\}$, and $\Delta_4 = \{c_2, c_4\}$. In order to better decide which of the candidates should be chosen, we can collect feedback from a community, for example, on the correctness of the different constraints.

Let us assume that N=10 users provided feedback on the overall correctness of the individual constraints on a rating scale 1...5 where *1=I am not sure if this is correct* and *5=absolutely correct*. Furthermore let us assume that the individual users provided the following feedback (average values): c_1 : 3.2, c_2 : 4.1, c_3 : 5.0, and c_4 : 2.2. The basic assumption now is that the diagnosis with the overall lowest evaluation in terms of correctness should be shown first. In the case of our working example, we have the following evaluations for our four different diagnoses: $\{(\Delta_1, 4.1), (\Delta_2, 2.7), (\Delta_3, 4.55), (\Delta_4, 3.15)\}$. Consequently, the diagnosis with the lowest evaluation is Δ_2, i.e., knowledge engineers should first take a look at the constraints mentioned in Δ_2.

CONCLUSIONS AND FUTURE WORK

In this short paper we provided an overview of basic approaches to solve diagnosis and redundancy detection tasks in the PEOPLEVIEWS environment. In this context we also sketched in which way diagnoses and redundant constraints can be determined on a more granular level, i.e., not on the constraint level, but on the level of individual factors. The approaches presented here help PEOPLEVIEWS users to find out ways from the no solution could be found dilemma and also to handle faulty constraints in an efficient fashion. Further work will include the extension of the diagnosis and redundancy detection approaches presented in this paper to settings where the basic assumptions on constraint structures mentioned in this paper are not needed.

REFERENCES

1. Felfernig, A., and Burke, R. Constraint-based recommender systems: Technologies and research issues. In *IEEE ICEC08* (Innsbruck, Austria, 2008), 17–26.
2. Felfernig, A., Hotz, L., Bagley, C., and Tiihonen, J. *Knowledge-based Configuration: From Research to Business Cases*, 1st ed. Elsevier/Morgan Kaufmann Publishers, 2014.
3. Felfernig, A., Schubert, M., and Zehentner, C. An Efficient Diagnosis Algorithm for Inconsistent Constraint Sets. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM) 25*, 2 (2012), 175–184.
4. Felfernig, A., Ulz, T., Haas, S., Schwarz, M., Reiterer, S., and Stettinger, M. PeopleViews: Human Computation for Constraint-based Recommendation. In *ACM RecSys'15 CrowdRec Workshop* (Vienna, Austria, 2015), 1–68.
5. Larson, M., Said, A., Shi, Y., Cremonesi, P., Tikk, D., and Karatzoglou, A. Activating the Crowd: Exploiting User-Item Reciprocity for Recommendation. In *CrowdRec13* (Hong Kong, China, 2013), 1–2.
6. Reiter, R. A theory of diagnosis from first principles. *AI Journal 23*, 1 (1987), 57–95.
7. Ulz, T., Schwarz, M., Felfernig, A., Haas, S., Reiterer, S., and Stettinger, M. Human Computation for Constraint-based Recommenders. *Journal of Intelligent Information Systems (JIIS)* (2016).
8. VonAhn, L. Human Computation. In *Technical Report CM-CS-05-193* (2005).