Application of Constraint-based Technologies in Financial Services Recommendation

Alexander Felfernig
Graz University of Technology, Austria
alexander.felfernig@ist.tugraz.at
Overview

• Motivation
• Constraint-based Recommendation
• Knowledge Acquisition
• Recommendation Knowledge for e-Learning
• Future Work: New Product Development
• Conclusions
Motivation

Financial services can be complex

Dependencies between services and additional restrictions (e.g., limited resources and age limits)

Recommendation algorithms have to take constraints into account

Knowledge acquisition issues exist (efforts in terms of knowledge base development & maintenance)

Efficient testing and debugging is needed
Motivation (contd.)

Recommendation process is conversational

Users have to be proactively supported (e.g., if no solution can be identified)

Sales representatives should not only rely on the proposed recommendations & explanations

Sales knowledge has to be internalized

Sales knowledge from recommenders should be reused when training sales representatives
Recommender Systems

- **Collaborative**: recommendations based on preferences of „nearest neighbors“

- **Content-based**: recommendations based on an analysis of consumed items

- **Knowledge-based**: usage of constraints (constraint-based recommendation) or similarities (critiquing)

- **Recommendations for groups**: i.e., not single users

- **Focus of this talk**: constraints

Constraint-based Recommendation as CSP

Variables $V = \{av, wr, rr\}$

Domains $D = \{\text{dom}(av) = \{\text{low}, \text{medium}, \text{high}\},$ 
$\quad \text{dom}(wr) = \{\text{low}, \text{medium}, \text{high}\},$ 
$\quad \text{dom}(rr) = \{\text{low}, \text{medium}, \text{high}\}\}$

Constraints $C = \{c1: \neg (av = \text{high} \land wr = \text{high}),$ 
$\quad c2: \neg (wr = \text{low} \land rr = \text{high}),$ 
$\quad c3: \neg (rr = \text{high} \land av = \text{high})\}$

Customer Requirements $CREQ = \{r1: av = \text{high},$ 
$\quad r2: rr = \text{low}\}$
Solutions & Ranking

- Solutions can be determined with **CSP solving**
- Often there exist **many solutions**
- Solutions **have to be ranked**
- **Utility-based**: ranking of solutions with regard to their contributions to a set of interest dimensions
- **Similarity-based**: distance of solution with regard to optimal values (e.g., less is better, more is better, etc.)
Resolving Inconsistencies in CREQ

Customer Requirements (CREQ) Diagnosis with Customer Requirements ($r_i \in \text{CREQ}$)

- Should be: consistent
- But: inconsistent

Diagnosis $\Delta \subseteq \text{CREQ}$:
$C \cup (\text{CREQ}-\Delta)$ consistent

Minimal($\Delta$): not $\exists \Delta'$: $\Delta' \subset \Delta$

Conflict Set $\text{CS} \subseteq \text{CREQ}$:
$\text{CS} \cup C$ inconsistent

Minimal($\text{CS}$): $\neg \exists \text{CS}'$: $\text{CS}' \subset \text{CS}$


### Prediction Quality (Precision)

#### Dataset:
Computer Configuration Interaction Log (n= 415 sessions).

#### Table:

<table>
<thead>
<tr>
<th>METHOD / N</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>breadth first</td>
<td>0.55</td>
<td>0.76</td>
<td>0.82</td>
<td>0.88</td>
<td>0.96</td>
</tr>
<tr>
<td>utility-based</td>
<td>0.66</td>
<td>0.83</td>
<td>0.94</td>
<td>0.95</td>
<td>0.98</td>
</tr>
<tr>
<td>similarity-based</td>
<td>0.65</td>
<td>0.81</td>
<td>0.90</td>
<td>0.93</td>
<td>0.98</td>
</tr>
<tr>
<td>probability-based</td>
<td>0.64</td>
<td>0.85</td>
<td>0.93</td>
<td>0.95</td>
<td>0.99</td>
</tr>
<tr>
<td>ensemble-based</td>
<td>0.68</td>
<td>0.86</td>
<td>0.94</td>
<td>0.96</td>
<td>0.99</td>
</tr>
</tbody>
</table>

#### Best First

- # allowed guesses

#### Formula:

\[
\text{precision} = \frac{\#(\text{correct predictions})}{\#(\text{predictions})}
\]

#### Statistics:
- avg. diagnoses: 5.32
- std.dev.: 1.67

---

MyLife

• **Alternative approach** to financial services advisory (integrated view on family situation)

• Recommenders determine **individual services**

• Further constraints used **for resource balancing**, e.g., investments per month should be below amount of disposable capital

• **MyLife features**: complete view on contracts, planning for whole family, visualization of goals
Experiences from Recommender Projects

- **Example FS provider**
  + Pre-informed customers
  + No faulty offers
  + Reduced efforts for *documenting advisory sessions*
  + Time savings of ~9mins per customer advisor session

- **Reason**: automated consistency checks & explanations

- **10% of products** sold on the basis of constraint-based recommenders (loans & funds), ~12,500 products p.a.
Knowledge Acquisition

• **Constraints** $C = \{c_1, c_2, \ldots, c_n\}$ have to be maintained

• **Knowledge acquisition bottleneck**: communication overheads, too much time needed for maintenance

• Improvements in terms of **regression testing** and **automated debugging** approaches

• If some of the **test cases** $T = \{t_1, \ldots, t_m\}$ are not fulfilled, corresponding diagnoses can be determined

---

Inconsistencies and Redundancies in C

Knowledge Base (C) Diagnosis

\[ \Delta \subseteq C : C - \Delta \text{ consistent} \]

Minimal(\(\Delta\)): \(\neg \exists \Delta' : \Delta' \subseteq \Delta\)

Redundancy Elimination (COREDIAG)

\[ \exists \alpha : (C - \{c_\alpha\}) \cup C \text{ incons.} \]

Conflict Set \(CS \subseteq C : CS \cup C \text{ incons.} \)

Minimal(CS): \(\neg \exists CS' : CS' \subseteq CS\)


Knowledge Base (C) Diagnosis with Test Cases \((t_i \in T)\)

- **Should be:** consistent
- **But:** inconsistent

\[ C \]
\[ c_1 \]
\[ c_2 \]
\[ c_3 \]
\[ c_n \]

\[ T \]
\[ t_1 \]
\[ t_2 \]
\[ t_k \]

**Diagnosis** \(\Delta \subseteq C:\)
\((C-\Delta) \cup \{t_i\}\) consistent \(\forall t_i \in T\)

**Minimal(\Delta):** not \(\exists \Delta': \Delta' \subset \Delta\)

**Conflict Set** \(CS \subseteq C:\)
\(\exists t_i \in T: \{t_i\} \cup CS\) inconsistent

**Minimal(CS):** \(\neg \exists CS': CS' \subset CS\)

WeeVis: modeling constraint-based recommenders, wiki-style definition and recommendation (weevis.org)
Knowledge Acquisition: Experiences

- Most domain experts (no computer science expertise) are **not able to directly develop knowledge bases**

- However, there are also **exceptions of the rule**

- **Reduced development efforts** compared to conventional (e.g. Java-based) software development

- **More efficient knowledge acquisition** due to automated testing and debugging support

- Needed: **test case generation for knowledge bases, human computation**
Recommendation Knowledge for e-Learning

- Sales representatives should not only rely on recommendations & explanations
- Expert knowledge about product domain needed
- **Means**: trainings, e-learning environments
- **Issue**: e-learning contents needed
- **Approach**: automated content generation (in our case: question generation)
StudyBattles User Interface

StudyBattles start screen

Learning applications, e.g., related to financial services
Basic Approach to Question Generation

• Recommendation task represented as CSP
• CSP can be exploited for generating questions …

(1) Given CREQ, which services to recommend?
(2) Inconsistent (CREQ), how to resolve inconsistency?
(3) Redundant (C), how to resolve redundancy?
(4) Inconsistent (C), how to resolve the inconsistency?

• Applicable for further types of well-formedness rules.

Future Work: New Product Development

- **New products and services** are developed

- **Feedback** from customers & sales needed (e.g., new sums)

- **Open Innovation** integrates communities as soon as possible into new product development

- **Group decision support** needed, e.g., in the idea screening phase
New Product Development: Decision Tasks

Example decisions: alternative sub-features (XOR), optional features, and optional sub-features (OR)
When to disclose preferences of other users?

Group discussions increase the decision quality, but the earlier preferences are disclosed, the lower the #comments.

Fig. 8. Number of comments in the CHOICLA discussion forum depending on preference disclosure time (after 1..3, or all users articulated preferences).

Ongoing Projects

• **WeWant**: Enabling Technologies for Group-based Configuration (Austria Research Promotion Agency)

• **PeopleViews**: Recommender Systems based on Human Computation (Austria Research Promotion Agency)

• **AGILE**: Adoptive Gateways for diverse MuLtiple Environments in the Internet of Things (H2020 Project)
Conclusions

• **Constraint-based technologies** help to improve the overall quality of FS advisory processes

• **Diagnosis** improves efficiency of conversational recommendation and knowledge engineering

• **Improvements**: reduced efforts related to customer advisory and KB development & maintenance, more efficient content generation

• **Open issues**: end-user programming support, automated test case generation, counteracting decision biases
Thank You!