Preference Elicitation with Subjective Features
Craig Boutilier, Kevin Regan & Paolo Viappiani
Department of Computer Science - University of Toronto

Introduction

- Make decisions on behalf of the user/help users making decisions
- Product configuration, recommender systems, personal assistants
- Preference elicitation
- What is the objective function?
- Is the elicitation effort worth the improvement it offers w.r.t. decision quality?
- Our vision: open-ended preference elicitation
- Let us express their preferences in a way that is natural to them

Subjective Features

- Preference elicitation usually focuses on "catalog" attributes (or product specifications)
- engine size, color, fuel economy, number of bedrooms,...
- We consider "user-defined" subjective features
- Constructed on the fly

Different Definitions of "Safe"

Model

Learn just enough about a concept in order to provide good recommendations

Abstract Model for Feature Elicitation

- Product space X \subseteq \mathbb{D} \times \{X_1, X_2, \ldots , X_D\}
- Reward r(x) reflects utility for catalog features
- Concept \text{c}(x) drawn from some hypothesis space \mathcal{H}
- Bonus p: additional utility for an x satisfying \text{c}(x)
- Utility \text{u}(x) = r(x) + p \text{c}(x)
- Goal: recommend products with highest utility
- Focus on combined elicitation of subjective features and reward weights
- \text{c}(x) and p are all unknown
- Version space \mathcal{V}
- Subset of \mathcal{H} that is consistent with the current knowledge about the concept
- \mathcal{W} is the set of feasible utility functions

Minimax Regret over Concepts and Utility Space

- Let V \subseteq \mathcal{V} be current version space
- \text{c}(x) reflects prior knowledge, responses, etc. Similarly W is updated
- The adversary chooses concept and witness x^\star
- \text{MAX}(V, W) = \max_{x \in V} \min_{y \in W} \text{u}(x)
- \text{MIN}(V, W) = \min_{x \in V} \max_{y \in W} \text{u}(x)
- If \text{MMR}(W) = r, x^\star is \text{c}-optimal.

Simultaneous Feature and Utility Elicitation

Query Type:
- Membership query
- Does a satisfying concept exist? Example: Do you consider this car safe?
- Comparison query
- Is x(u) the same? Example: Do you prefer this car to that car?

Responses to concept query refine version space

Responses to comparison queries impose conditional constraints w.r.t. W

Minimax Regret

\begin{align}
\text{MMR}(W) &= \max_{x \in V} \min_{y \in W} \text{u}(x)
\end{align}

\begin{align}
\text{MMR}(V, W) &= \max_{x \in V} \min_{y \in W} \text{u}(x)
\end{align}

\begin{align}
\text{MMR}(W) &= \min_{x \in V} \max_{y \in W} \text{u}(x)
\end{align}

\begin{align}
\text{MMR}(V, W) &= \min_{x \in V} \max_{y \in W} \text{u}(x)
\end{align}

Computing Minimax Regret: Conjunctions

- Difficulties computing minimax regret:
  - Mismatch (query) program (not straight min or max)
  - Generally, quadratic objective
  - General approach:
    - Benders' decomposition and constraint generation to break minimax program
    - Various encoding tricks to linearize-quadratic terms
  - The Minimax Regret Computation is encoded as a Mixed Integer Program

Experimental Results - Effectiveness

- MMR vs # Queries

Constraint Generation

- Constraint generation avoids the enumeration of W and V
  - \text{REPEAT}
  - Solve minimization problem with a subset GEN of V: W
  - The adversary's hands are tied to choose a concept from this subset
  - \text{UB} of minimax regret
  - \text{LB} of minimax regret
  - Find max violated constraint computing \text{MR}(u)
  - \text{UB} of minimax regret
  - \text{LB} of minimax regret
  - \text{UB} of minimax regret
  - \text{LB} of minimax regret
  - \text{UB} of minimax regret

Experimental Results - Sensitivity

- Number of comparison queries vs the number of membership queries used
- Greater bonus value: refining the concept becomes more critical

Summary & Future Directions

- Contributions:
  - Minimax regret formulation over concept
  - Query strategies to reduce regret
- Future Directions:
  - Further development of query strategies
  - Non-additive utility models such as SAI
  - Richer hypothesis spaces

Combined Comparison Strategy Methodology (CMMN)

- Asks both combination and membership queries about x^\star and x^w
- In general, counts as 3 queries