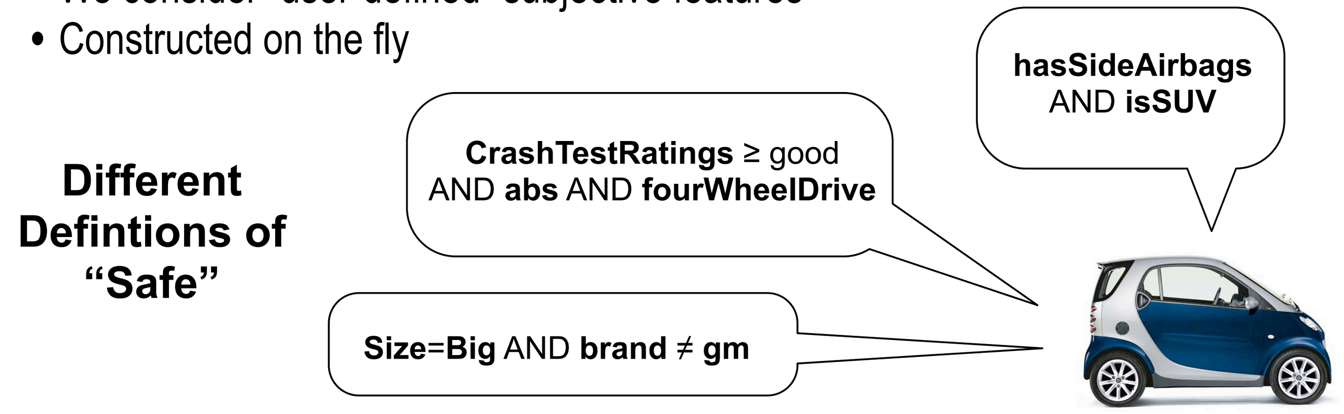
Preference Elicitation with Subjective Features

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 users 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



Model

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

Abstract Model for Feature Elicitation

- Product space X ⊆ Dom{X1 … Xn}
  - Reward r(X) reflects utility for catalog features
  - Concept c(X) drawn from some hypothesis space H
  - Bonus p: additional utility for an x satisfying c(x)
  - Utility u(x) = r(x) + p c(x)
  - Goal: recommend products with highest utility
- Focus on combined elicitation of subjective features and reward weights
  - r,c and p are all unknown
- Version space V
  - Subset of H that is consistent with the current knowledge about the concept
- W is the set of feasible utility functions

Minimax Regret over Concepts and Utility Space

- Let V⊆ H be current version space
- c∈V iff c respects prior knowledge, responses, etc. Similarly W is updated
- The adversary chooses concept and witness x^w

$$MR(\mathbf{x}; W, V) = \max_{w \in W} \max_{c \in V} \max_{\mathbf{x}' \in \mathbf{X}} u(\mathbf{x}'; w, c) - u(\mathbf{x}; w, c)$$
$$MMR(W, V) = \min_{\mathbf{x} \in \mathbf{X}} MR(\mathbf{x}; W, V)$$
$$\mathbf{x}^*_{W,V} = \arg\min_{\mathbf{x} \in \mathbf{X}} MR(\mathbf{x}; W, V)$$

- If MMR(W,V) = ε, x* is ε–optimal.

Simultaneous Feature and Utility Elicitation

Query Type:

- Membership query
  - Does x satisfy concept c. Example: Do you consider this car safe?
- Comparison Query
  - Is u(x) > u(y). 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

$$\begin{aligned} w\mathbf{x} - w\mathbf{y} &> 0 \text{ if } c(\mathbf{x}), c(\mathbf{y}) \\ w\mathbf{x} + b - w\mathbf{y} &> 0 \text{ if } c(\mathbf{x}), \neg c(\mathbf{y}) \quad (2) \\ w\mathbf{x} - w\mathbf{y} - b &> 0 \text{ if } \neg c(\mathbf{x}), c(\mathbf{y}) \\ w\mathbf{x} - w\mathbf{y} &> 0 \text{ if } \neg c(\mathbf{x}), \neg c(\mathbf{y}) \quad (4) \end{aligned}$$

These can be encoded with a set of IP constraints, for example (2):

$$w\mathbf{x} + b - w\mathbf{y} > [\sum_{j \le n} I(\neg \mathbf{x}[j]) + (1 - I(\neg \mathbf{y}[k]))]\, \Delta\!\downarrow \quad \forall k \le n$$

The number of constraints is quadratic in the number of attributes due to (4)

Query Strategies

Aim: reduce regret quickly

Strategies for Membership Queries

- **Halving**: aims to learn concept directly
  - "random" query x until positive response; then refine (unique) most specific concept in V (negate one literal at a time)
- **Current Solution** (MCSS): tackle regret directly
  - If x*, xw both in cw : query xw (unless certain)
  - If xw in cw but not x* : query x* (unless certain)
  - If x*, xw both not in cw : query xw if xw (unless certain)

Strategies for Comparison Queries

- **Current Solution** (CCSS):
  - compare x* and xw

Strategies for deciding query type

- **Phased:** always ask membership queries if there is some concept uncertainty
- **Interleaved:** ask comparison queries when the reward component of regret is higher: r(xw;w)-r(x*;w) > w_b(c(xw) - c(x*))

Combined Comparison Membership Strategy (CMM)

- Asks both comparison and membership queries about x* and x^w
- In general, counts as 3 queries

Computing Minimax Regret: Conjunctions

- Difficulties computing minimax regret:
  - Minimax (integer) 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

$$\begin{aligned} &\min \quad \delta \\ &\text{s.t. } \delta \ge r(\mathbf{x}^*_{w,c}) - r(X_1, \cdots X_n) \\ &\qquad + b(\mathbf{x}^*_{w,c}, c) - w_b I^c \quad \forall c \in V, \forall w \in W \\ &\quad I^c \le X_j \qquad \forall c \in V, \forall x_j \in c \\ &\quad I^c \le 1 - X_j \qquad \forall c \in V, \forall \overline{x}_j \in c \end{aligned}$$

Constraint Generation

Constraint generation avoids the enumeration of W and V

- REPEAT
- Solve minimization problem with a subset GEN of W,V
  - The adversary's hands are tied to choose a concept from this subset
  - LB of minimax regret
- Find max violated constraint computing MR(x)
  - UB of minimax regret
- Add the concept to GEN
- Terminate when UB = LB

In practice MMR computation require less than 1s

Max Regret: Conjunctions

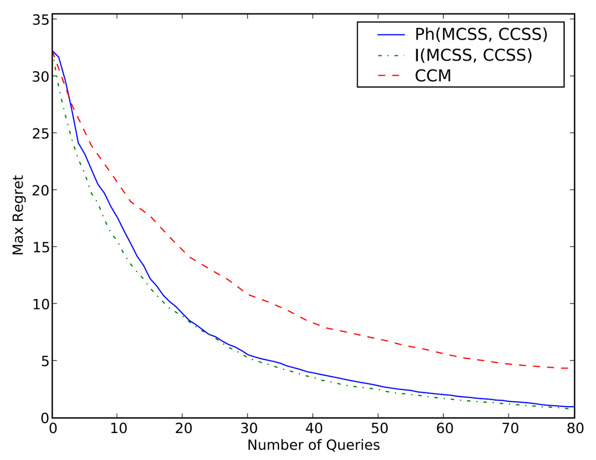
Find maximally violated constraint: combination of weight vector and concept that maximizes regret MR(x*,W,V)

- Let E^+, E^- be positive, negative instances

$$\begin{aligned} &\max \quad \sum_{j \le n} Y_j + Z^a - \sum_{j \le n} w_j \mathbf{x}[j] - Z^x \\ &\text{s.t. } B^a + I(x_j) \le X_j + 1.5 \quad \forall j \le n \\ &\quad B^a + I(\overline{x}_j) \le (1 - X_j) + 1.5 \quad \forall j \le n \\ &\quad B^x \ge 1 - \sum_{j:\mathbf{x}[j] \text{ positive}} I(\overline{x}_j) - \sum_{j:\mathbf{x}[j] \text{ negative}} I(x_j) \\ &\quad \sum_j I(\neg \mathbf{y}[j]) = 0 \quad \forall \mathbf{y} \in E^+ \\ &\quad \sum_j I(\neg \mathbf{y}[j]) \ge 1 \quad \forall \mathbf{y} \in E^- \\ &\quad Y_j \le X_j w_j\!\uparrow; \quad Y_j \le w_j \quad \forall j \le n \\ &\quad Z^a \le B^a w_b\!\uparrow; \quad Z^a \le w_b \\ &\quad B^x w_b\!\downarrow \le Z^x; \quad B^x w_b\!\uparrow \le Z^x + w_b\!\uparrow - w_b \\ &\quad (w_1, \cdots, w_n, w_b) \in W; \quad (X_1, \cdots, X_n) \in \mathbf{X} \end{aligned}$$

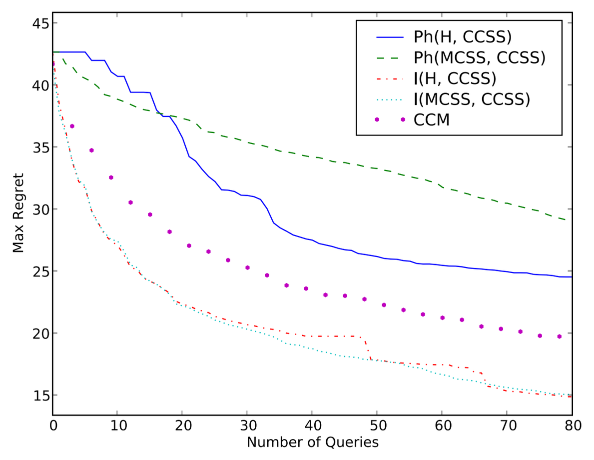
Experimental Results - Effectiveness

MMR vs # Queries



20 variables, 60 constraints

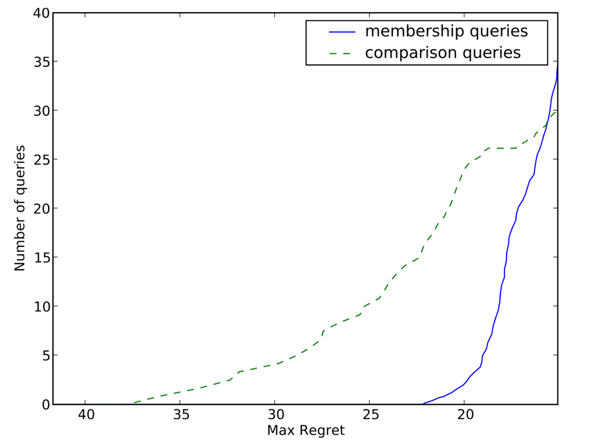
MMR vs # Queries



30 variables, 90 constraints

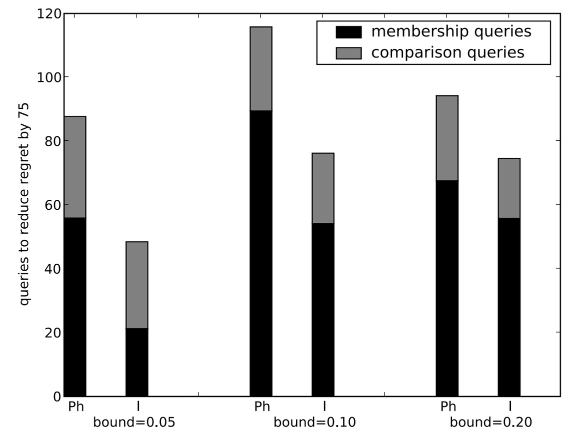
Experimental Results - Sensitivity

Interleaved MCSS & CCSS



Number of comparison queries vs the number of membership queries used

Bonus Sensitivity



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 GAI
- Richer hypothesis spaces