Conversational Recommender Systems: From Critiquing to Utility-based Elicitation

Paolo Viappiani
CNRS and LIP6
UPMC, France

June, 25, 2013
Summary

- Overview of preference-handling systems
- Conversational recommender systems and behavioral aspects
- Critiquing-based systems
- Utility-based recommendations
- Recommendations with subjective features
Preference-aware Systems

- Personalization technologies are becoming widespread (web applications and more)
  - Aiding the user in carrying out tasks
  - Recommender systems, conversational systems, personal agents, interface assistants,…

- Interactive artificial intelligence systems employ preferences in both reasoning and interaction
  - Personalization is a source of benefit and challenges
  - Designing successful preference-aware systems is challenging
Technical Challenges

1) **Degree of interactivity**: from fully manual through varying degrees of mixed initiative, to fully automated.

2) **Preference representation**: which way to effectively model preferences?

3) **Preference elicitation**: how to assess the user preference model

4) **Nature of the reasoning with preferences**: how to do inference, add new preferences, use the preference information

5) **Resulting output or behavior of the system**: information, decisions, actions

From Peintner, Viappiani, Yorke-Smith: *Preferences in Interactive Systems: Technical Challenges and Case Studie*

AI Magazine Fall 2008 special Issue on Preferences
Recommender Systems

- Conversational systems (Interactive)
  - Aim to replace a «decision analyst»
  - The user is (at least partially) active
  - Mixed-initiative interaction, dialog systems, query-answering

- Collaborative filtering (usually *one-shot*)
  - Aim to provide automatic marketing
  - (Mostly) Passive observations of user preferences
  - Statistical analysis of user data

- The distinction is more and more blurred
Preference-based Search

- **Product configuration**
- **Large collection of outcomes**

  - Users are not familiar with available items and features
  - Users do not know their preferences: theory of preference construction [Payne]
  - Biases in decision: framing, prominence, means-objectives [Gilovich, Kahneman]
SmartClient [Torrens, Pu & Faltings]

- Interactive critiquing based on constraint satisfaction/optimization
- Preferences as soft constraints
- Use of visualization techniques to highlight performance of products wrt different criteria
Teaching Salesman [Stolze]

- « Need-based » conversational recommender system
- TS helps user choose an item based on his or her needs
- Options characterized at the user layer, abstract features and user attributes
- Three different “phases”:
  - Preference discovery
  - Preference optimization
  - Preference debugging
Collaborative Filtering

- Help users making choices, advertise interesting products (personalized marketing), suggest new products based on purchases (ratings) of similar users
  
  - Basic idea: similar people like similar things
  - Identify similarity between users and/or items
Personal Agents

- Autonomous personalized agents, need to be proactive
- Examples: CAP (calendar apprentice) and PexA (project execution assistant)
- Cognitive assistants, Tutoring systems
- When to take action? When to prompt the user?
- How to expose the agent's assistance without degrading the user experience

[Myers et al., 2007]
Personalized Interfaces: Supple

[Gajos & Weld]

Adapt interfaces according to preferences and abilities

Can improve UI by either critiquing or by direct query-answering

http://www.eecs.harvard.edu/~kgajos/research/supple/
Ad Placement

Point of view of publisher: which bidding strategy?
Point of view of advertiser: which mechanism?
Social Networks & AI

- Rising interests in modeling social networks with mathematical models
- **Goal**: learn the structure of the networks
  - Who is friend with who?
  - *People you might know...*
- **The Contagion** effect
  - Product adoption can spread through the network
  - *Homophily*: people behave as their peers
  - Which nodes are most influential?
- **Applications**: targeted marketing campaign, recommender systems with social component
Preference-based Search

“If users cannot find a product, they cannot buy it”, J. Nielsen
Is form-filling a reasonable strategy for preference elicitation?

- Example: actual scenario with travel website (July 5th, 2006)
- User wants to travel from Geneva to Dublin
- Return flight
- Preferences
  - Outbound flight, arrive by 5pm
  - Inbound flight, arrive by 3pm
  - (Cheapest)
To be there at 5pm, I should leave around noon.

To arrive back at 3pm, I should leave in the morning.
Leave out preference about SWISS

Still expensive but cheaper; does not arrive at the preferred time
Outbound arrives by 5pm; Return arrives by 3pm, as desired.
Preference Construction & Decision Biases

- Users’ preferences are often **constructed** when considering specific examples

  [Payne et al. ’93; Slovic’95; Tversky ’96]

  **Incorrect means objectives:** formulate the real goal by a “substitute” goal believed to lead to desired outcome

  - Users often state more preferences than necessary when prompted
  - The preference model may be complete, but not accurate

- Decision biases studied by *behavioral decision theory*
Dynamic Critiquing

- Show a set of “compound critiques” mining data for patterns (Original work proposed the use of Apriori)
- Incremental critiquing: improve it by using weighted model
“Advanced” critiquing

- Exploit previous sessions, similar users
  - Experience-based critiquing: account for critiquing history [Smyth et al., 2010]
  - Nearest-neighbour compatibility critiquing [Felfering et al., 2012]
User-based vs System-based Critiquing

User-Initiated Critiquing

System-Suggested Critiquing

[Chen & Pu]
Since the user cannot observe accuracy directly, it is important that users feel confident about their choices!

- **Accuracy**: the system's ability to help users find their most preferred item
- **Confidence**: its ability to inspire trust in selecting the item displayed by the system
- **Effort**: amount of effort (cognitive, time) required
Guidelines for Trust inspiring interfaces [Chen & Pu]

- Any effort / any order / any preferences
- Allow preference revision and partial satisfaction
- Tradeoff assistance
  - Tweaking / critiquing
  - “I like this portable PC, can I find something lighter?” Clustering of similar products, explanation why they are recommended...

<table>
<thead>
<tr>
<th>Why?</th>
<th>Price</th>
<th>Processor speed</th>
<th>Battery life</th>
<th>Installed memory</th>
<th>Hard drive capacity</th>
<th>Display size</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$1,379.00</td>
<td>3.3 GHz</td>
<td>2 hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,179.00</td>
<td>3.2 GHz</td>
<td>2 hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,529.00</td>
<td>1.7 GHz</td>
<td>6.5 hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,599.00</td>
<td>1.7 GHz</td>
<td>6.5 hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,425.00</td>
<td>1.6 GHz</td>
<td>5.5 hour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$2,235.00</td>
<td>1.8 GHz</td>
<td>2.5 hour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,900.00</td>
<td>3.2 GHz</td>
<td>1 hour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,125.00</td>
<td>1.5 GHz</td>
<td>5 hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$2,319.00</td>
<td>1.67 GHz</td>
<td>4.5 hour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,499.00</td>
<td>1.5 GHz</td>
<td>5 hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,739.99</td>
<td>1.5 GHz</td>
<td>4.5 hour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,629.00</td>
<td>1.8 GHz</td>
<td>5.8 hour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,625.99</td>
<td>1.5 GHz</td>
<td>5 hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1,426.99</td>
<td>1.5 GHz</td>
<td>5 hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Price</th>
<th>Processor speed</th>
<th>Battery life</th>
<th>Installed memory</th>
<th>Hard drive capacity</th>
<th>Display size</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$1,185.00</td>
<td>1.67 GHz</td>
<td>4.5 hours(s)</td>
<td>512 MB</td>
<td>80 GB</td>
<td>36.5 cm</td>
<td>2.04 kg</td>
</tr>
<tr>
<td></td>
<td>$1,739.99</td>
<td>1.5 GHz</td>
<td>4.5 hours(s)</td>
<td>512 MB</td>
<td>80 GB</td>
<td>36.5 cm</td>
<td>2.04 kg</td>
</tr>
<tr>
<td></td>
<td>$1,625.00</td>
<td>1.6 GHz</td>
<td>5 hours(s)</td>
<td>512 MB</td>
<td>80 GB</td>
<td>36.5 cm</td>
<td>2.04 kg</td>
</tr>
<tr>
<td></td>
<td>$1,426.99</td>
<td>1.5 GHz</td>
<td>5 hours(s)</td>
<td>512 MB</td>
<td>80 GB</td>
<td>36.5 cm</td>
<td>2.04 kg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Price</th>
<th>Processor speed</th>
<th>Battery life</th>
<th>Installed memory</th>
<th>Hard drive capacity</th>
<th>Display size</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$1,215.00</td>
<td>1.6 GHz</td>
<td>5 hours(s)</td>
<td>1 GB</td>
<td>100 GB</td>
<td>36.1 cm</td>
<td>2.05 kg</td>
</tr>
<tr>
<td></td>
<td>$1,148.00</td>
<td>2 GHz</td>
<td>4 hours(s)</td>
<td>1 GB</td>
<td>100 GB</td>
<td>36.1 cm</td>
<td>2.05 kg</td>
</tr>
<tr>
<td></td>
<td>$1,219.00</td>
<td>1.6 GHz</td>
<td>5 hours(s)</td>
<td>1 GB</td>
<td>100 GB</td>
<td>36.1 cm</td>
<td>2.05 kg</td>
</tr>
<tr>
<td></td>
<td>$1,426.99</td>
<td>1.5 GHz</td>
<td>5 hours(s)</td>
<td>1 GB</td>
<td>100 GB</td>
<td>36.1 cm</td>
<td>2.05 kg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Price</th>
<th>Processor speed</th>
<th>Battery life</th>
<th>Installed memory</th>
<th>Hard drive capacity</th>
<th>Display size</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$1,215.00</td>
<td>1.6 GHz</td>
<td>5 hours(s)</td>
<td>1 GB</td>
<td>100 GB</td>
<td>36.1 cm</td>
<td>2.05 kg</td>
</tr>
<tr>
<td></td>
<td>$1,148.00</td>
<td>2 GHz</td>
<td>4 hours(s)</td>
<td>1 GB</td>
<td>100 GB</td>
<td>36.1 cm</td>
<td>2.05 kg</td>
</tr>
<tr>
<td></td>
<td>$1,219.00</td>
<td>1.6 GHz</td>
<td>5 hours(s)</td>
<td>1 GB</td>
<td>100 GB</td>
<td>36.1 cm</td>
<td>2.05 kg</td>
</tr>
<tr>
<td></td>
<td>$1,426.99</td>
<td>1.5 GHz</td>
<td>5 hours(s)</td>
<td>1 GB</td>
<td>100 GB</td>
<td>36.1 cm</td>
<td>2.05 kg</td>
</tr>
</tbody>
</table>
Anchoring effect

- Users are biased to what is shown to them (Tversky 1974)
- Example
  - Three laptops that all weigh around 3-4 kg
  - The user might never consider a lighter model
- Metaphor: local optimum
  - When all options look similar, motivation to state additional preference is low
- Need for diversity
  - Case-based Reasoning
    - $\text{Quality}(x, S) = a \times \text{Sim}(\text{target}) + (1-a) \times \text{Diversity}(x, S)$
    - $\text{Quality}(x, S) = \text{Sim}(\text{target}) \times \text{Diversity}(x, S)$
- Model-based Suggestions
  - Probability that an option can become optimal
- Utility-based “diversity” (see later)
Accuracy and effort of Critiquing Interfaces

- A number of user studies showed that critiquing systems are easy to use, often converge to a “good” item
  - Accuracy measured by comparing to off-line in-depth choice analysis

- Model-based suggestions according to the “lookahead principle” effective in stimulating expression of correct preferences (avoiding anchoring effect)
  - Suggestions should not be optimal under the current preference model, but should provide a high likelihood of optimality when an additional preference is added

- User-based critiquing often more accurate than system-based critiques, but it requires more effort (and more motivation)
Effort Model: Interview vs Critiquing

- Interview-based model
- Example-based (critiquing)

Empirical explanation of the superiority of example-based tool (in particular critiquing) compared to questionnaire: users consistently perceive an improvement so they have an incentive in continuing the interaction until they found a high-quality product [Pu et al, 2011]
Inspired by multi attribute utility theory, this approach maintains a set of “weights” that are tweaked upon each user selection.

- Weight adaptation based on heuristic
Decision-theoretic Preference Acquisition

- Heuristic methods some time work reasonably well.
- Can we do better by adopting a principled decision-theoretic view?
- What AI methods can offer to preference elicitation (learning) for real time interactive systems?
Classic Utility Elicitation

- Assessment of multi attribute utility functions
  - (Usually) Long list of questions
  - Suitable for high stake decisions
- Queries
  - **Local**: focus on an attribute in isolation
  - **Global**: compare different attributes
  - First learn local utilities (for each attributes), then scaling factors

**STANDARD GAMBLE QUERIES**

Choose between option $x_{12}$ **for sure**

OR

a gamble $< x^T, l, x^\perp, 1-l >$ ?

Semantic of SGB is equivalent to ask: “$U(x_{12}) > l$ ?”
Preference Elicitation in AI

- AI systems need to recommend decisions on behalf of individuals (groups)

- The *preference elicitation bottleneck*
  - What is the objective function?
  - User preferences (or utilities) are unknown
  - Elicitation of preferences is expensive!

- Challenging questions
  - What are sources of preference information?
  - What preference info is *relevant* the task at hand?
  - Is the elicitation effort *worth the improvement* it offers in terms of decision quality
A General Framework for Elicitation and Interactive Decision Making

- **Bel**: beliefs about user’s utility function \( u \)
- **Opt(Bel)**: “optimal” decision given incomplete, noisy, and/or imprecise beliefs about \( u \)

- Repeat until **Bel** meets some termination condition
  - **Ask** user some query (propose some interaction) \( q \)
  - **Observe** user response \( r \)
  - **Update** **Bel** given \( r \)

- Return/recommend **Opt(Bel)**

Two main frameworks to represent uncertainty:

- **Strict Uncertainty** (constraints)
- **Bayesian** (update using Bayes rule)
Online Recommendation Systems

Options shown with dual goal of recommendation and elicitation

- User might end the interaction unpredictably
- A set can be interpreted as a query or as a recommendation
  - Are we losing anything?
  - Computational advantages?
- Challenges
  - Limited number of interactions
  - Large product catalogs
  - Small display set
  - User responses are noisy

[Reilly, Zhang, McGinty, Pu, Smyth]
Exploitation vs Exploration?

- Natural tension between **recommendation** and **elicitation**

- How to resolve this tension?
  - Focus on *choice queries*
    - A set can be interpreted **both** as a query and as a recommendation

**Goal**: exploit current information

**Note**: since utility is uncertain, there can be value in recommending a set

**Goal**: acquire further information in order to make better recommendation
How to Choose Recommendations and Queries?

- How to reason with an uncertain utility function?
- How to aggregate utility uncertainty?
- We extend classic decision criteria to sets

<table>
<thead>
<tr>
<th></th>
<th>Bayesian elicitation</th>
<th>Regret based elicitation (strict uncertainty)</th>
<th>Maximin elicitation (strict uncertainty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Recommendation</td>
<td>Expected Utility</td>
<td>Minimax Regret</td>
<td>Maximin Utility</td>
</tr>
<tr>
<td>Recommendation Set</td>
<td>Expected Utility of Selection</td>
<td>Setwise Regret</td>
<td>Setwise Maximin</td>
</tr>
<tr>
<td>Query Set</td>
<td>Expected Posterior Utility</td>
<td>Worst-case posterior Regret</td>
<td>Worst-case posterior maximin</td>
</tr>
</tbody>
</table>
Utility is parametric in $w$: we write $u(x; w)$ (assume form is given!)

For instance: $u(x; w) = w \cdot x$ (linear utility model)

$\theta$ is the current belief

$P(w; \theta)$ probability of parameter $w$

Expected utility of an outcome \emph{wrt} utility uncertainty

$$EU(x; \theta) = \int u(x; w) \, P(w; \theta) \, dw$$

$$x^* = \arg\max \ EU(x; \theta) \text{ is the optimal outcome \emph{wrt} } \theta$$

$$EU^* = \max \ EU(x; \theta)$$
Bayesian Elicitation: Criteria for Recommendation Sets and Queris

- **Expected Utility of Selection**

  I recommend you one of the following

  \[
  EUS_R(S; \theta) = \sum_{x \in S} P_R(S \sim x; \theta) EU(x; \theta | S \sim x)
  \]

  

  - Probability of response (selection)
  - Value of selection in updated belief given selection

<table>
<thead>
<tr>
<th>(0,0,1,1,0)</th>
<th>(1,0,1,1,0)</th>
<th>(1,1,0,1,1)</th>
</tr>
</thead>
</table>

- **Expected Posterior Utility**

  Which one do you prefer?

  \[
  EPUS_R(S; \theta) = \sum_{x \in S} P_R(S \sim x; \theta) EU^*(\theta | S \sim x)
  \]

  

  - Probability of response (selection)
  - Optimal value in updated belief given response

  | (0,0,1,1,0) | (1,0,1,1,0) | (1,1,0,1,1) |
Bayesian Elicitation: Criteria for Recommendation Sets and Queris

- **Expected Utility of Selection**

  \[
  EUS_R(S; \theta) = \sum_{x \in S} P_R(S \sim x; \theta) EU(x; \theta | S \sim x)
  \]

  I recommend you one of the following:
  
  \[(0,0,1,1,0), (1,0,0,1,0), (1,1,0,1,1)\]

- **Expected Posterior Utility**

  \[
  EPUR_R(S; \theta) = \sum_{x \in S} P_R(S \sim x; \theta) EU^*(\theta | S \sim x)
  \]

  Which one do you prefer?

  \[(0,0,1,1,0), (1,0,0,1,0), (1,1,0,1,1)\]
Response Model

- Selection probabilities $P_R$ depend on:
  - The underlying (true) utility
  - The user's cognitive ability to select the most preferred outcome

- Probability of selecting $x$ from $S$ given utility $w$
  - Response model gives $P_R(S \rightarrow x; w)$
  - Then, we can compute selection probability given belief $\theta$
    $$P_R(S \rightarrow x; \theta) = \int P_R(S \rightarrow x; w) P(w; \theta) \, dw$$

- Some common response models
  1) Noiseless responses
  2) Constant: probability $p$ of selecting a non-preferred outcome
  3) Logistic model (aka Luce-Sheppard, mixed-multinomial logit)
Logistic Noisy Response Model

Set A

- Selection probability: $P_R \approx 0$
- $P_R \approx 1$

Set B

- Selection probability: $P_R \approx 0.5$
- $P_R \approx 0.5$

Probability of selection

Logistic function with temperature parameter $\gamma$

$$P_R(S \sim x; w) = \frac{e^{\gamma u(x; w)}}{e^{\gamma u(x; w)} + e^{\gamma u(y; w)}}$$

utility difference

$\gamma$
“Operator T”

Define an operator $T_\theta : S \rightarrow S$

set $S$

Which one do you prefer?

set $S' = T_\theta(S)$

EU* optimal in $\theta'$

EU* optimal in $\theta''$
"Operator T"

Define an operator $T_\theta: S \rightarrow S$

set $S$

Which one do you prefer?

**Lemma** The new set $T_\theta(S)$ is better than $S$ both as query and as recommendation

$$EPU(\cdots) \geq EUS(\cdots) \geq EPU(\cdots) \geq EUS(\cdots) \text{ in } \theta$$

for the noiseless and constant noise model

set $S' = T_\theta(S)$

EU* optimal in $\theta'$

EU* optimal in $\theta''$
Sets can be viewed as both recommendation and choice queries

- *Expected Utility of Selection*: value of a set as recommendation
- *Expected Value of Information*: value of a set as a choice query

Different response/selection models: noiseless, constant noise, logistic (aka mixed multinomial logit, Luce-Sheppard)

**Theorem:**

Optimal recommendation sets are optimal choice queries

- Assuming noiseless responses or a constant noise model
- No particular assumption about prior distribution, methods of Bayesian inference.

**Theorem:** Optimal recommendation sets are near-optimal queries under the logistic noise model

- We provide the expression for the worst-case loss $\Delta_{\text{max}}$ (surprisingly small)
- Also, the optimal query assuming noiseless responses is a near-optimal query under logistic noise
Logistic Noise Model

- Expected Utility of Selection under a noisy model \( \leq \) than the same selection under noiseless model
  - \( \text{Loss} = EUS_{NL}(S; \theta) - EUS_{L}(S; \theta) \)

- Can we derive bounds under the logistic model?

Set A

- Selection probability: \( P_R \approx 0 \) and \( P_R \approx 1 \)
- Select item with highest utility
- \( \text{Loss} \rightarrow 0 \)

Set B

- Selection probability: \( P_R \approx 0.5 \)
- Both items have \( \sim \) same utility
- \( \text{Loss} \rightarrow 0 \)
Theorem (Logistic noise model)

1) The value of a recommendation set is at most \( \Delta_{\text{max}} \) lower than the noiseless

- \( EUS_L(S) \geq EUS_{NL}(S) - \Delta_{\text{max}} \)
- **Consequence**: we can optimize EUS without considering noise

2) The difference between the value of the optimal query and the optimal recommendation set is at most \( \Delta_{\text{max}} \)

- \( EPU_L^* \geq EUS_L^* - \Delta_{\text{max}} \)
- **Consequence**: we can optimize a set wrt EUS and use it as a query (with bounded loss)

\( \Delta_{\text{max}} \) can be expressed in function of the set size and the temperature parameter \( \gamma \) (surprisingly low in practice)
Efficient Algorithms for Bayesian Elicitation

- Consequences of our theoretical results: efficient algorithms to generate choice queries
  - Optimizing a recommendation set is simpler and *submodular*
  - Approximated *greedy* strategies with *worst-case guarantees*
  - *Noiseless* optimization quite effective in *noisy* settings
  - Query Iteration strategy particularly efficient for large datasets

<table>
<thead>
<tr>
<th>Computation time</th>
<th>Dataset 1 Size=187</th>
<th>Dataset 2 Size=506</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact EPU</td>
<td>1815s</td>
<td>~2 weeks</td>
</tr>
<tr>
<td>Exact EUS</td>
<td>405s</td>
<td>~2 hours</td>
</tr>
<tr>
<td>Greedy with lazy evaluation</td>
<td>1.02s</td>
<td>0.93 s</td>
</tr>
<tr>
<td>Query Iteration (local search)</td>
<td>0.15s</td>
<td>0.05 s</td>
</tr>
</tbody>
</table>

![Graph showing normalized average loss vs. number of queries]

- Exact EPU
- Greedy (EUS, NL)
- QI (sampling)
- QI (rand)
- random
How Many To Show?

- EVOI in function of the number of items in the query set
- Dataset
  - 506 items
- Logistic noise model
- Monte Carlo methods for Bayesian inference
How to Choose Recommendations and Queries?

- How to reason with an uncertain utility function?
- How to aggregate utility uncertainty?
- We extend classic decision criteria to sets

<table>
<thead>
<tr>
<th>Single Recommendation</th>
<th>Bayesian elicitation</th>
<th>Regret based elicitation (strict uncertainty)</th>
<th>Maximin elicitation (strict uncertainty)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Expected Utility</strong></td>
<td><strong>Minimax Regret</strong></td>
<td><strong>Maximin Utility</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recommendation Set</th>
<th>Expected Utility of Selection</th>
<th>Setwise Regret</th>
<th>Setwise Maximin</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Query Set</th>
<th>Expected Posterior Utility</th>
<th><strong>Worst-case posterior Regret</strong></th>
<th>Worst-case posterior maximin</th>
</tr>
</thead>
</table>
Minimax Regret

$W = \text{set of feasible utility parameters}$

Recommend product $x^*$ associated with *minimax regret*

Max regret:
$$MR(x; W) = \max_{y \in X} \max_{w \in W} u(x^a; w) - u(x; w)$$

Minimax regret:
$$MMR(W) = \min_{x \in X} MR(x, W); \quad x^*_w = \arg\min_{x \in X} MR(x, W)$$
Computation of Minimax Regret

Benders' decomposition + constraint generation techniques

**“Master” problem**

\[
\begin{align*}
\min_{x,R} & \quad R \\
\text{s.t.} & \quad R > wx^*_w - wx \quad \forall w \in \text{GEN}
\end{align*}
\]

**Constraint Generation**

(calculate max regret of the master's solution)

**“Slave” problem**

\[
\begin{align*}
\max_{w,y,Y} & \quad \sum_i Y_i - wx \\
\text{s.t.} & \quad \text{constraints}(W) \\
Y \text{ “encode” } w^*y
\end{align*}
\]

add the \( w \) and \( x^*_w \) to GEN
Regret-based Elicitation: Criteria for Recommendation Sets and Queries

- **Setwise Regret**

  \[ SMR(S; W) = \max_{x^a \in X} \max_{w \in W} u(x^a; w) - \max_{x \in S} u(x; w) \]

  I recommend you the following:
  
  (0,0,1,1,0)  (1,0,1,0,0)  (1,1,0,1,1)

- **Worstcase Regret**

  \[ WR(Z) = \max \left[ MMR(W^{Z \to 1}), \ldots, MMR(W^{Z \to k}) \right] \]

  Which one do you prefer?
  
  (0,0,1,1,0)  (1,0,1,0,0)  (1,1,0,1,1)
- Sets can be viewed as **both** recommendation and choice queries
  - Setwise Regret: value of a set as recommendation
  - Worstcase Regret: value of a set as a choice query

**Theorem:**

*Optimal recommendation sets are optimal choice queries*

- In the regret-based elicitation model
Comparison Query
Bound Query

You are asked to decide whether the apartment on the left is "closer" in value to the TOP apartment or the BOTTOM apartment.

Features that are not shown (including price) are the same for all three apartments. Note that any features shown in grey are also the same for all apartments.

You have previously indicated that BOTTOM has the worst combination of features, and TOP has the best combination of features. On the scale from 0 to 100 (shown on the right of the bins) BOTTOM is at 0, and TOP is at 100. You should consider of where the apartment in question falls on this scale. If its value is between 0 and the tip of the slider, please drag it to the bottom bin; otherwise, drag it to the top bin.
Regret-based Recommendation
Preference Elicitation with Subjective Features

- Usually focuses on “catalog” attributes
  - Fix vocabulary of features: e.g., *engine*, *size*, *color*, *fuel economy*, ...

- Preferences are often most naturally expressed in terms of **subjective features**
  - *Safe, cool, sporty, big, trendy, light, modern*...
  - User-specific
  - In our work, an unknown Boolean formula (conjunction)
Feature Elicitation

- Subjective definition means we have to elicit them
- Feature elicitation vs. classical concept learning
  - Learn just enough about a concept in order to make a good decision
  - Near optimal recommendation with weak concept knowledge
  - Minimize user queries

Example: preference for safe cars, BUT fuel economy more important.

If all “safe cars” have poor fuel economy, it is not worth continuing to learn more about safety!
Joint Elicitation with Minimax Regret

- Simultaneous elicitation of user \textit{features} and \textit{utility}
  - Doing one “completely” followed by other is wasteful
  - Optimal or near-optimal recommendation is possible with little concept and utility information

- Contributions
  1) Define a model that allows \textit{simultaneous} elicitation of user features and utility
  2) \textbf{Minimax regret optimization} in presence of both utility and feature uncertainty, providing robust recommendations
  3) Several heuristic techniques for eliciting concepts and utility that reduce \textit{regret} quickly
Abstract Model

- Product space $X \subseteq \text{Dom}\{X_1, \ldots, X_n\}$
  - Utility $u(x; w, c)$ function of unknown weights and feature; weight $w$ reflects tradeoffs between features
  - Concept $c(x)$ drawn from some hypothesis space $H$
  - Bonus weight $w_b$: additional utility for an $x$ satisfying $c(x)$
  - Goal: recommend product with highest utility

\[
u(x; w, c) = wx + w_bc(x) \quad \text{with } w \text{ in } W, c \text{ in } V \text{ (unknown)}\]

- $(V, W)$ consistent with prior knowledge
Query Types

Which one do you prefer?

Is this car SAFE?

\[ X_1 \land \neg X_2 \land .. \]

\[ \neg X_1 \land \neg X_2 \land .. \]

\[ u( X_1 \land \neg X_2 \land .. ) > u( \neg X_1 \land \neg X_2 \land .. ) \ ? \]

\[ c( X_1 \land \neg X_2 \land .. ) == True \ ? \]

Comparison queries

Membership queries

Answers impose constraints on V only

Answers impose constraints on W and V

\[ c( X_1 \land \neg X_2 \land .. ) \]

\[ c( X_1 \text{ and not } X_2 ) = 0 \]
Minimax Regret over Concepts and Utility

Let \((W, V)\) be current utility and version space

\[
MR(x; W, V) = \max_{w \in W} \max_{c \in V} \max_{x^a \in X} u(x^a; w, c) - u(x; w, c)
\]

\[
MMR(W, V) = \min_{x \in X} MR(x; W, V)
\]

Current solution \((x^*, x^a, w, c)\)
Elicitation

- **Aim**: reduce *minimax regret* quickly
  - Empirically, *actual regret* also reduces quickly

- **Strategies**
  1) Which type of query to ask
  2) What to ask

- **CSS comparisons**
  - Current solution \((x^*, x^a, w, c)\) immediately suggest query
  - Ask to compare \(x^*\) and \(x^a\)

Which one do you prefer?
Which Type of Query to Ask?

- Decompose the “source” of regret
- Given the current solution \((x^*, x^a, w, c)\)
- MaxRegret = RewardRegret (RR) + ConceptRegret (CR)

\[
\begin{align*}
\mathcal{w} \cdot (x^a - x^*) \\
\mathcal{w}_b (c(x^a) - c(x^*))
\end{align*}
\]

Strategies:

- **Interleaved strategies (I)** asks comparison query when \(RR > CR\)
- **Phased Strategies (Ph)** always ask membership when \(CR > 0\)
- **Combined comparison-membership query (CCM):** asks both comparison and membership queries about \(x^*\) and \(x^a\) (In general, counts as 3 queries)
Large concepts (defined over 10 attributes)

*Interleaved* elicitation strategies are better off than *phased* strategies

Our CSS-based heuristics better than *Halving-based* strategies
Conclusions

- Preference-handling systems
- Psychological biases in decision making and preference elicitation
- Preference elicitation with utility-based techniques
- Recommendation sets and optimal queries: bayesian framework, regret-based and maximin frameworks
- Elicitation of Subjective features

**Acknowledgements:** works done in collaboration with Craig Boutilier (utility-based recommendations), Boi Faltings, Pearl Pu (critiquing systems) and others
References 1

General on Preferences in AI


SmartClient


Teaching Salesman


Supple


PexA


Biases in Decision-making

- Kahneman, Daniel; Paul Slovic, Amos Tversky (1982), Judgment under Uncertainty: Heuristics and Biases, Cambridge University Press,
- Gilovich, Thomas; Dale Griffin, Daniel Kahneman (2002), Heuristics and biases: The psychology of intuitive judgment, Cambridge University Press

Foundations of Preference Reasoning


Example-critiquing

Dynamic Critiquing
- Kevin McCarthy, Yasser Salem, Barry Smyth: Experience-Based Critiquing: Reusing Critiquing Experiences to Improve Conversational Recommendation. ICCBR 2010: 480-494
- Monika Mandl, Alexander Felfernig: Improving the Performance of Unit Critiquing. UMAP 2012: 176-187

Minimax Regret
- Darius Braziunas, Craig Boutilier: Minimax regret based elicitation of generalized additive utilities. UAI 2007: 25-32

UTPREF (Decision-support system based on Regret)
- Darius Braziunas, Craig Boutilier: Assessing regret-based preference elicitation with the UTPREF recommendation system. ACM Conference on Electronic Commerce 2010: 219-228

Bayesian elicitation

Optimal Recommendation Sets

Feature Elicitation
- Craig Boutilier, Kevin Regan, Paolo Viappiani: Online feature elicitation in interactive optimization. ICML 2009
- Craig Boutilier, Kevin Regan, Paolo Viappiani: Simultaneous Elicitation of Preference Features and Utility. AAAI 2010