Regret-based Optimal Recommendation Sets in Conversational Recommender Systems

Paolo Viappiani, Craig Boutilier
Department of Computer Science
University of Toronto

ACM Conference on Recommender Systems 2009
Recommendations Sets

- Show products that are both
  - Expected to be rated highly
  - Maximally informative should we have feedback

This work: **optimal recommendation set** given a *sound decision-theoretic semantics* of the user interaction
“Dynamic Critiquing” for navigation of a set of products with system-generated alternatives/critiques \cite{Smyth, McGuinty, Reilly}

product similarity + APRIORI alternatives

Evaluated on real users \cite{Reilly, Zhang, Smyth, Pu}
Recommendations with an Explicit Utility Model

- Associate user's actions with a precise, sound semantics
  - E.g. critique impose linear constraints on a user utility function

- Advantages of our approach
  - Suggest a set of products
  - Bound the difference in quality of the recommendation and the optimal option of the user
  - Determine which options and critiques carry the most information
  - Suggest when terminate the process

- We adopt the notion of minimax regret to face utility uncertainty
  - Extend it to the case of a set of joint recommendations
Minimax Regret definition

\[ W = \text{set of feasible utility parameters} \]
\[ X = \text{set of products} \]
\[ x = \text{recommendation} \]

- **Max regret**
  \[ \text{MR}(x; W) = \max_{y \in X} \max_{w \in W} u(y; w) - u(x; w) \]

- **Minimax regret and minimax regret optimal** \( x^*_w \) :
  \[ \text{MMR}(W) = \min_{x \in X} \text{MR}(x, W) \quad x^*_w = \arg\min_{x \in X} \text{MR}(x, W) \]
### Feature 1 vs Feature 2

<table>
<thead>
<tr>
<th></th>
<th>Feature 1</th>
<th>Feature 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>0.35</td>
<td>0.68</td>
</tr>
<tr>
<td>$O_2$</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td>$O_3$</td>
<td>0</td>
<td>0.75</td>
</tr>
<tr>
<td>$O_4$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$O_5$</td>
<td>0.5</td>
<td>0.3</td>
</tr>
</tbody>
</table>

**Utility Function**

$$U(x) = w_1 \cdot f_1(x) + (1-w_1) \cdot f_2(x)$$

where $w_1$ is unknown.

### Adversary MR

<table>
<thead>
<tr>
<th>Adversary</th>
<th>MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>$O_4$</td>
</tr>
<tr>
<td>$O_2$</td>
<td>$O_3$</td>
</tr>
<tr>
<td>$O_3$</td>
<td>$O_4$</td>
</tr>
<tr>
<td>$O_4$</td>
<td>$O_3$</td>
</tr>
<tr>
<td>$O_5$</td>
<td>$O_4$</td>
</tr>
</tbody>
</table>

$o_5$ minimax regret optimal
Regret-based recommender

$W$ set of feasible utility functions

1) Initialize $W$ with initial constraints
2) **DO** Generate current recommendations
3) Refine $W$ given user's feedback
4) **LOOP** until user stops OR regret < $\varepsilon$

**Initial minimax regret** = 0.5

User: o2 better than o1 → regret = 0.07

User: o4 better than o2 → regret = 0
Utility of a set

The value of a set is dependent on the elements of the set *jointly*. We assume:

\[
\text{Utility}(\begin{pmatrix} A \\ B \\ C \end{pmatrix}) = \max \left\{ \begin{array}{c} U(A) \\ U(B) \\ U(C) \end{array} \right\}
\]

- A recommendation set gives “shortlisted” alternatives
- Reasonable in practice: apartment search example
We choose the set of \( k \) options first, but delay the final choice from the slate after the adversary has chosen a utility function \( w \) in \( W \).

Minimum difference between options in the slate and (real) best option.

The setwise max regret \( \text{SMR}(Z; W) \) of a set \( Z \):

\[
\text{SMR}(Z; W) = \max_{y \in X} \max_{w \in W} \min_{x \in Z} u(y; w) - u(x; w)
\]

The setwise minimax regret \( \text{SMMR}(W) \) and the optimal set \( Z^*_W \):

\[
\text{SMMR}(W) = \min \text{SMR}(Z, W) \quad Z^*_W = \arg\min_{Z \subseteq X: |Z|=k} \text{SMR}(Z, W)
\]
<table>
<thead>
<tr>
<th>Set</th>
<th>Adversary</th>
<th>$w_1$</th>
<th>SMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>{o_1, o_4}</td>
<td>o_3</td>
<td>0</td>
<td>0.07</td>
</tr>
<tr>
<td>{o_1, o_2}</td>
<td>o_3</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>{o_3, o_2}</td>
<td>o_4</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>{o_3, o_4}</td>
<td>o_3</td>
<td>0.42</td>
<td>0.11</td>
</tr>
<tr>
<td>{o_5, o_4}</td>
<td>o_4</td>
<td>0</td>
<td>0.45</td>
</tr>
</tbody>
</table>

\{o_1, o_4\} setwise minimax regret optimal
Incorporating User Feedback

Slate Z of k options viewed as a “query set” - user picks one

- Worst-case Regret (wrt each possible answer)
  - $WR(Z) = \max [\text{MMR}(W^{1>2}), \text{MMR}(W^{2>1})]$ 

- To drive further elicitation, minimize WR
  - Relationship between SMR and WR ?
Incorporating User Feedback

- Slate $Z$ of $k$ options viewed as a “query set”
  - User picks one

- Consider $k$ possible cases
  - $1^{st}$ option preferred $\rightarrow W^{Z\rightarrow 1}$
  - $2^{nd}$ option preferred $\rightarrow W^{Z\rightarrow 2}$
  - …

- Worst-case Regret
  - $WR(Z) = \max [ MMR(W^{Z\rightarrow 1}), .., MMR(W^{Z\rightarrow k}) ]$

- To drive further elicitation, minimize $WR$
  - Relationship between SMR and $WR$?
Theorem

- The optimal recommendation set $Z^*_w$ is also the (myopically) optimal query set wrt worst-case regret (WR)

$\rightarrow$ “Best recommendation set = best query set”

- The optimal query set can be chosen without enumeration
  - If we can compute setwise regret efficiently (next slide)
Setwise Regret Computation

- Setwise minimax regret can be formulated as a MIP
  - Benders' decomposition + constraint generation techniques

\[
\begin{align*}
\min_{M, I_w^j, x^j, V_w^j} & \quad M \\
\text{s.t.} & \quad M \geq \sum_{1 \leq j \leq k} V_w^j \quad \forall w \in \text{Vert} \\
& \quad V_w^j \geq w \cdot (x_w^* - x^j) + (I_w^j - 1) m_{big} \\
& \quad \forall j \in [1, k] \land \forall w \in \text{Vert} \\
& \quad \sum_{1 \leq j \leq k} I_w^j = 1 \quad \forall w \in \text{Vert} \\
& \quad I_w^j \in \{0, 1\} \\
& \quad V_w^j \geq 0 \quad \forall j \in [1, k], \forall w \in \text{Vert}
\end{align*}
\]
Hillclimbing procedure
“minimax-regret rewriting”

Given a set $Z = \{x^1, \ldots, x^k\}$

DO

• Partition the utility space
• $X^1$ option preferred $\rightarrow$ new space $W^{Z\rightarrow 1}$
• ...
• $X^k$ option preferred $\rightarrow$ new space $W^{Z\rightarrow k}$
• Replace $x^i$ with $x^*_W$, the MMR-optimal in $W^i$

WHILE $\text{SMR}(Z^{\text{new}}) < \text{SMR}(Z)$

The inner replacement can be proved not to increase SMR

- Start with $\{o_5, o_4\}$
- Assume $o_4$ better than $o_5$
  • Compute MMR: this gives $o_2$
- Assume $o_5$ better than $o_4$
  • Compute MMR: this gives $o_1$
- New query $\{o_1, o_2\}$
Chain of Adversaries

- Current solution strategy (CSS) - only for $k=2$
  - Consider set $\{x^*_w, \text{Adv}(x^*_w)\}$
    \[
    \text{Adv}(x,W) = \arg\max_y \text{MR}(x,y,W)
    \]

- Setwise chain of adversaries (SCAS): $\{x^1, \ldots, x^k\}$
  - Use setwise notion of adversary
    \[
    \text{SMR-Adv}(Z,W) = \arg\max_y \text{SMR}(Z,y,W)
    \]

\[
\begin{cases}
x^1 = x^*_w \\
x^i = \text{SMR-Adv}(\{x^1, \ldots, x^{i-1}\})
\end{cases}
\]
Empirical Results

- Randomly generated *quasilinear* utility functions
- Real dataset (~200 options)
- User iteratively picks preferred option in a pair (k=2)
- Measure regret reduction
- SMMR recommendations are significantly better than CSS
- Hillclimbing (HCT) is as good as SMMR
Critiquing Simulation

- Simulate a critiquing session
  - Quasilinear utility model
  - Synthetic dataset (5000 options)
- “Optimizing” user chooses best critique wrt real utility
- Alternate btw
  - Selection of feature to improve ('unit critique')
  - Selection among a set of 3 suggestions
- HCT-based set recommendations gives best regret reduction
Real Loss

- Real loss (regret) is the difference to the actual optimum
- Set size $k=3$
- Regret-based recommender gives optimal recommendation in very few cycles
Conclusions

- Formalization of recommendations of a joint set of alternatives
  - We propose a new criterion *setwise regret*
    - Intuitive extension of regret criterion
    - Guarantee on the quality of the recommendation set
    - Efficient driver for further elicitation

- Optimal recommendations sets = optimal query sets
  - Computation & heuristics

- Application to critiquing systems

- Current and future works
  - User studies
  - “Noisy” models
  - Subjective features (see our poster!)
(Single Item) Minimax Regret Computation

- Configuration problems
  - Benders' decomposition and constraint generation to break minimax program

- Discrete datasets
  - Adversarial search with two pllys
  - Heuristics:
    - order to maximize pruning
    - Sample hypercube vectors
Constraint Generation

- Constraint generation: avoid enumeration of $V$
  - *REPEAT*
  - Solve minimization problem with a subset $GEN$ of $V$
    - The adversary's hands are tied to choose a couple $(w, y)$ 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
<table>
<thead>
<tr>
<th></th>
<th>Feature 1</th>
<th>Feature 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>0.35</td>
<td>0.68</td>
</tr>
<tr>
<td>$o_2$</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td>$o_3$</td>
<td>0</td>
<td>0.75</td>
</tr>
<tr>
<td>$o_4$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$o_5$</td>
<td>0.5</td>
<td>0.3</td>
</tr>
</tbody>
</table>

$U(x) = w_1 * f_1(x) + (1-w_1) * f_2(x)$

$w_1$ unknown

<table>
<thead>
<tr>
<th></th>
<th>Adversary</th>
<th>MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$o_4$</td>
<td>0.65</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$o_3$</td>
<td>0.55</td>
</tr>
<tr>
<td>$o_3$</td>
<td>$o_4$</td>
<td>1</td>
</tr>
<tr>
<td>$o_4$</td>
<td>$o_3$</td>
<td>0.75</td>
</tr>
<tr>
<td>$o_5$</td>
<td>$o_4$</td>
<td>0.5</td>
</tr>
</tbody>
</table>

$o_5$ minimax regret optimal