

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

Instructions: Please use this system to find the laptop that you want to buy. You can either click the button for each feature on the left panel, or select one of the recommended products below. [Click here for more instructions...](#)

Laptop features

Brand:
 Apple

ProcessorType:
 Core Duo

ProcessorSpeed(GHz):

ScreenSize(inches):

Memory(MB):

HardDriveCapacity(GB):


Weight(lbs):

OperatingSystem:
 MacOSX 10

BatteryLife(hours):

Price(\$):

We recommend this laptop for you



Price: 2199 USD
1759.2 EUR
2748.75 CHF

Main Features:

- ProcessorType:Core Duo
- ProcessorSpeed(GHz):1.83
- ScreenSize(inches):15.4
- Memory(MB):1024
- HardDriveCapacity(GB):100
- Weight:5.5lbs (2.5kg)
- OperatingSystem:MacOS X 10.4
- BatteryLife(hours):5.6

Product Description:
You've seen improvements in notebook performance before - but never on this scale. The Intel Core Duo powering MacBook Pro is actually two processors built into a single chip. This, combined with myriad other engineering leaps, boosts performance up to four times higher than the PowerBook G4. With this awesome power, it's a breeze to render complex 3D models, enjoy smooth playback of HD video, or host a four-way video conference.

Not satisfied with the result? you may select other recommendations listed below

- 1. Faster CPU.**
But with More Expensive.
[>>see product detail<<](#)
- 2. Faster CPU and Cheaper.**
But with Less Memory and Smaller Hard-Disk.
[>>see product detail<<](#)
- 3. Lighter and Cheaper.**
But with Different type of CPU, Slower CPU, Smaller Screen, Less Memory, Smaller Hard-Disk and Shorter Battery Life.
[>>see product detail<<](#)
- 4. Larger Screen and Larger Hard-Disk.**
But with Different Type of CPU, Slower CPU, Less Memory, Heavier, Shorter Battery Life and More Expensive.
[>>see product detail<<](#)
- 5. Lighter and Longer Battery Life.**
But with Different Brand, Slower CPU, Smaller Screen, Different OS and More Expensive.
[>>see product detail<<](#)

“Dynamic Critiquing” for navigation of a set of products with system-generated alternatives/critiques [Smyth, McGuinty, Reilly]

product similarity + APRIORI alternatives

Evaluated on real users [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 = set of feasible utility parameters

X = set of products

x = recommendation

■ Max regret

$$MR(x; W) = \max_{y \in X} \max_{w \in W} u(y; w) - u(x; w)$$

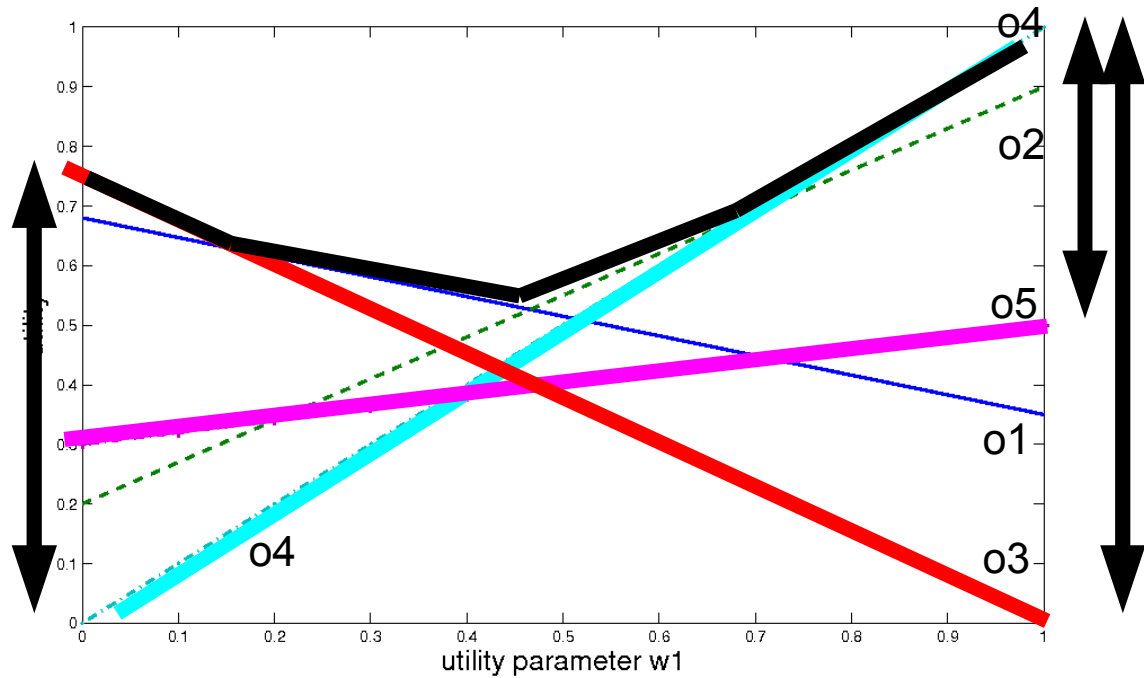
■ Minimax regret and minimax regret optimal x_w^* :

$$MMR(W) = \min_{x \in X} MR(x, W) \quad x_w^* = \operatorname{argmin}_{x \in X} MR(x, W)$$

	Feature 1	Feature 2
O_1	0.35	0.68
O_2	0.9	0.2
O_3	0	0.75
O_4	1	0
O_5	0.5	0.3

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

w_1 unknown



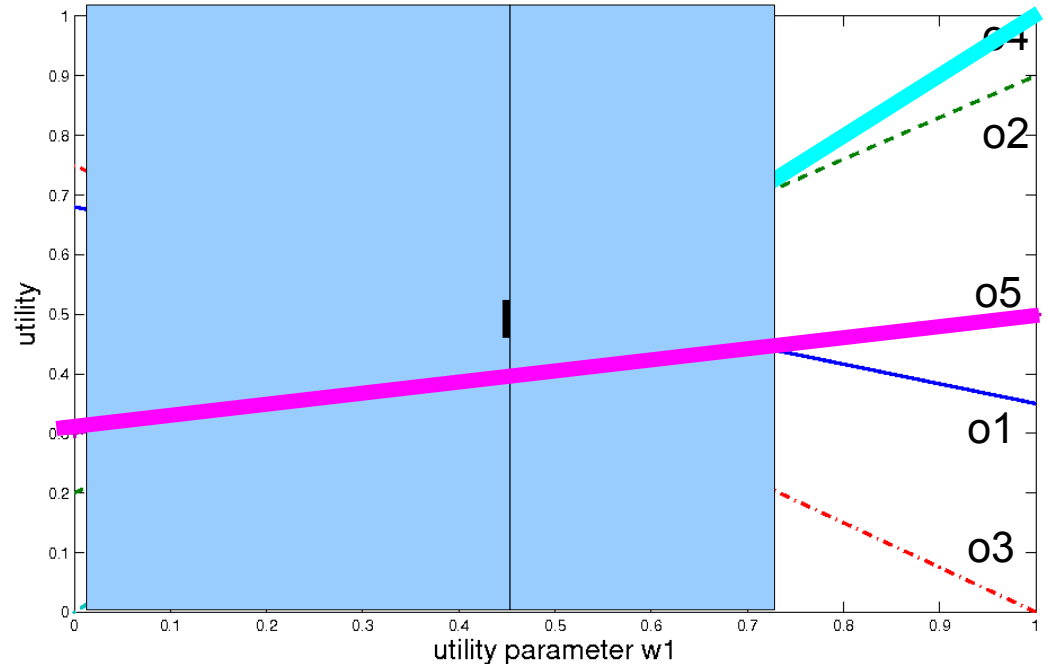
	Adversary	MR
O_1	O_4	0.65
O_2	O_3	0.55
O_3	O_4	1
O_4	O_3	0.75
O_5	O_4	0.5

O_5 minimax regret optimal 6

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 $< \epsilon$



Initial minimax regret = 0.5

User: o2 better than o1 \rightarrow regret = 0.07

User: o4 better than o2 \rightarrow regret = 0

Utility of a set



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

We assume:

$$\blacksquare \text{Utility} \left(\begin{matrix} A \\ B \\ C \end{matrix} \right) = \max \left\{ \begin{matrix} U(A) \\ U(B) \\ U(C) \end{matrix} \right\}$$

- A recommendation set gives “shortlisted” alternatives
- Reasonable in practice: apartment search example

Regret \rightarrow Setwise Regret

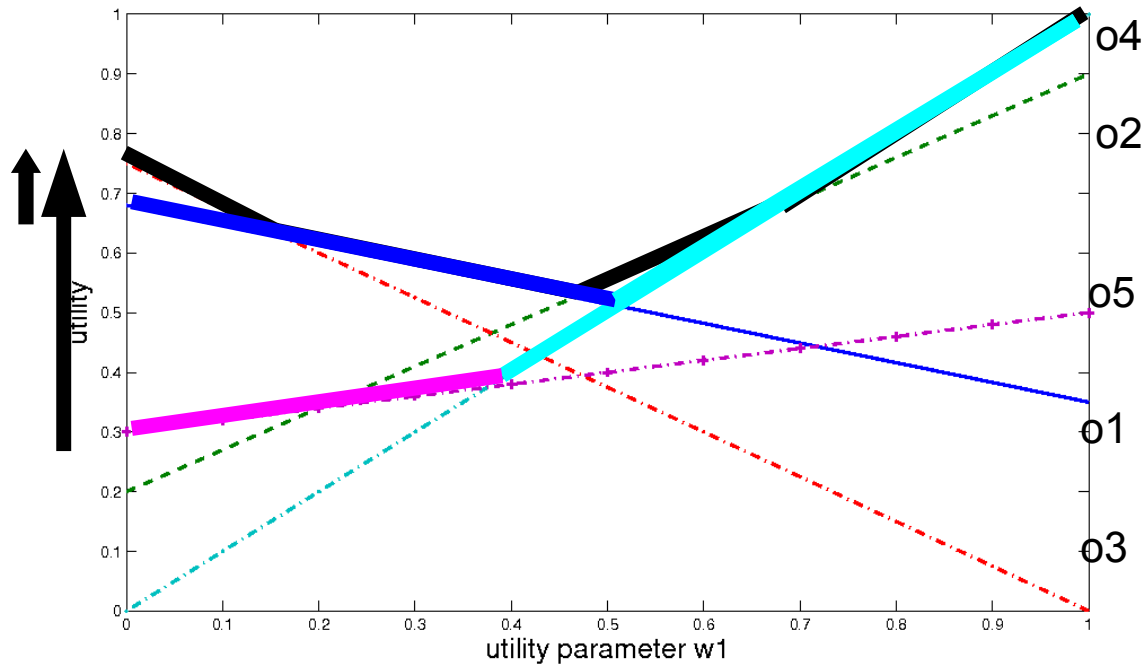
- We choose the set of k options first, but *delay* the final choice from the slate after the adversary has chosen a utility function w in W
- Minimum difference btw options in the slate and (real) best option
- The setwise max regret $\text{SMR}(Z; W)$ of a set Z :

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

- The *setwise* minimax regret $\text{SMMR}(W)$ and the optimal set Z_w^* :

$$\text{SMMR}(W) = \min_{Z \subset X: |Z|=k} \text{SMR}(Z, W)$$

$$Z_w^* = \operatorname{argmin}_{Z \subset X: |Z|=k} \text{SMR}(Z, W)$$

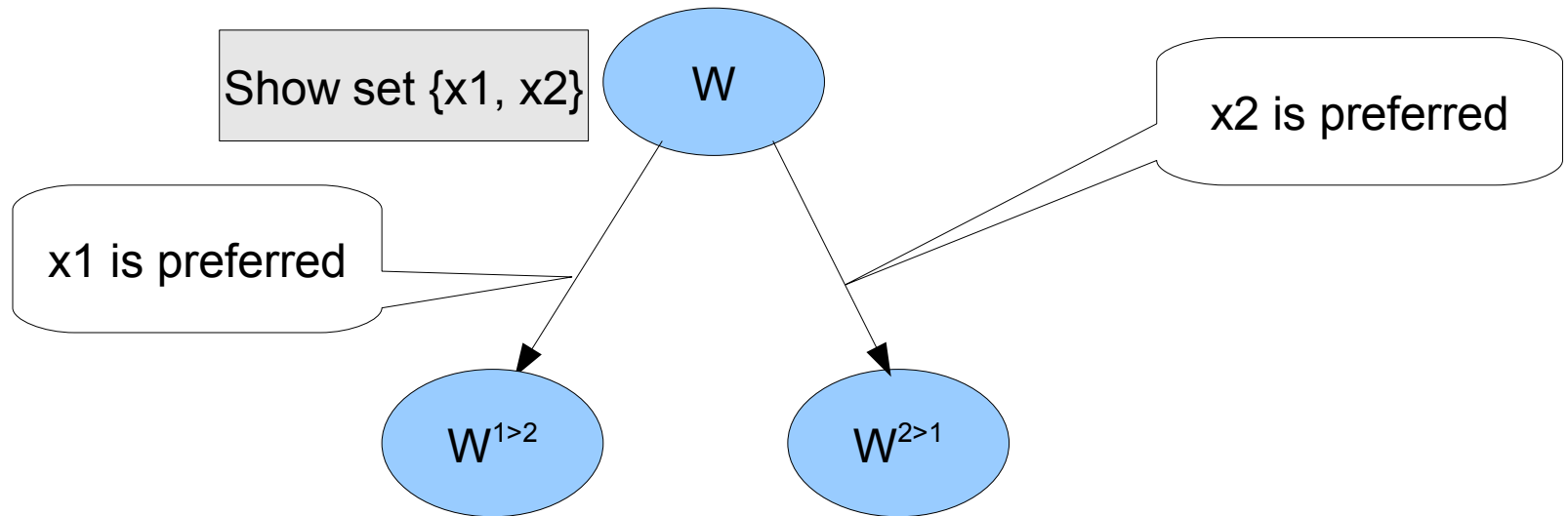


Set	Adversary	w_1	SMR
$\{o_1, o_4\}$	o_3	0	0.07
$\{o_1, o_2\}$	o_3	1	0.1
$\{o_3, o_2\}$	o_4	1	0.1
$\{o_3, o_4\}$	o_3	0.42	0.11
$\{o_5, o_4\}$	o_4	0	0.45

$\{o_1, o_4\}$ *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)
 - $WR(Z) = \max [MMR(W^{1>2}), MMR(W^{2>1})]$
- 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
 - 1st option preferred $\rightarrow W^{Z \rightarrow 1}$
 - 2nd option preferred $\rightarrow W^{Z \rightarrow 2}$
 - ...
- Worst-case Regret
 - $WR(Z) = \max [MMR(W^{Z \rightarrow 1}), \dots, MMR(W^{Z \rightarrow k})]$
- 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)
→ ***“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

$$\begin{aligned} & \min_{M, I_{\mathbf{w}}^j, \mathbf{X}^j, V_{\mathbf{w}}^j} && M \\ & \text{s.t. } M \geq \sum_{1 \leq j \leq k} V_{\mathbf{w}}^j && \forall \mathbf{w} \in Vert \\ & V_{\mathbf{w}}^j \geq \mathbf{w} \cdot (\mathbf{x}_{\mathbf{w}}^* - \mathbf{X}^j) + (I_{\mathbf{w}}^j - 1)m_{big} && \forall j \in [1, k] \wedge \forall \mathbf{w} \in Vert \\ & \sum_{1 \leq j \leq k} I_{\mathbf{w}}^j = 1 && \forall \mathbf{w} \in Vert \\ & I_{\mathbf{w}}^j \in \{0, 1\} \\ & V_{\mathbf{w}}^j \geq 0 && \forall j \in [1, k], \forall \mathbf{w} \in Vert \end{aligned}$$

Hillclimbing procedure

“minimax-regret rewriting”

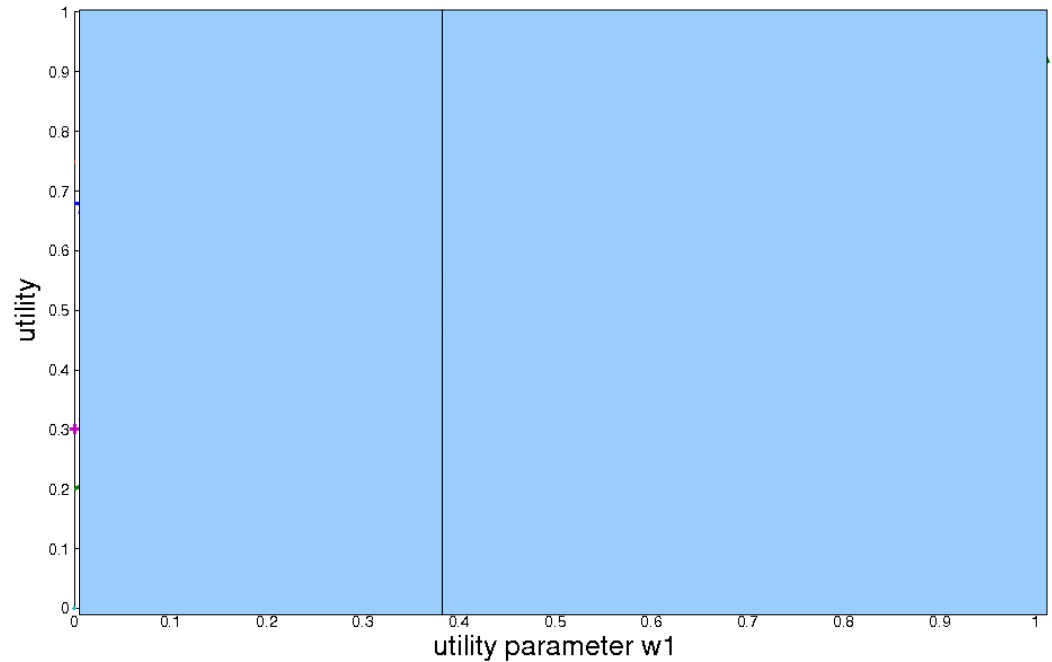
Given a set $Z = \{x^1, \dots, x^k\}$

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 $x_{w_i}^*$, the MMR-optimal in W^i

■ **WHILE** $SMR(Z^{new}) < SMR(Z)$

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 $\{x^*_w, \text{Adv}(x^*_w)\}$

$$\text{Adv}(x, W) = \text{argmax}_y \text{MR}(x, y, W)$$

- Setwise chain of adversaries (SCAS): $\{x^1, \dots, x^k\}$

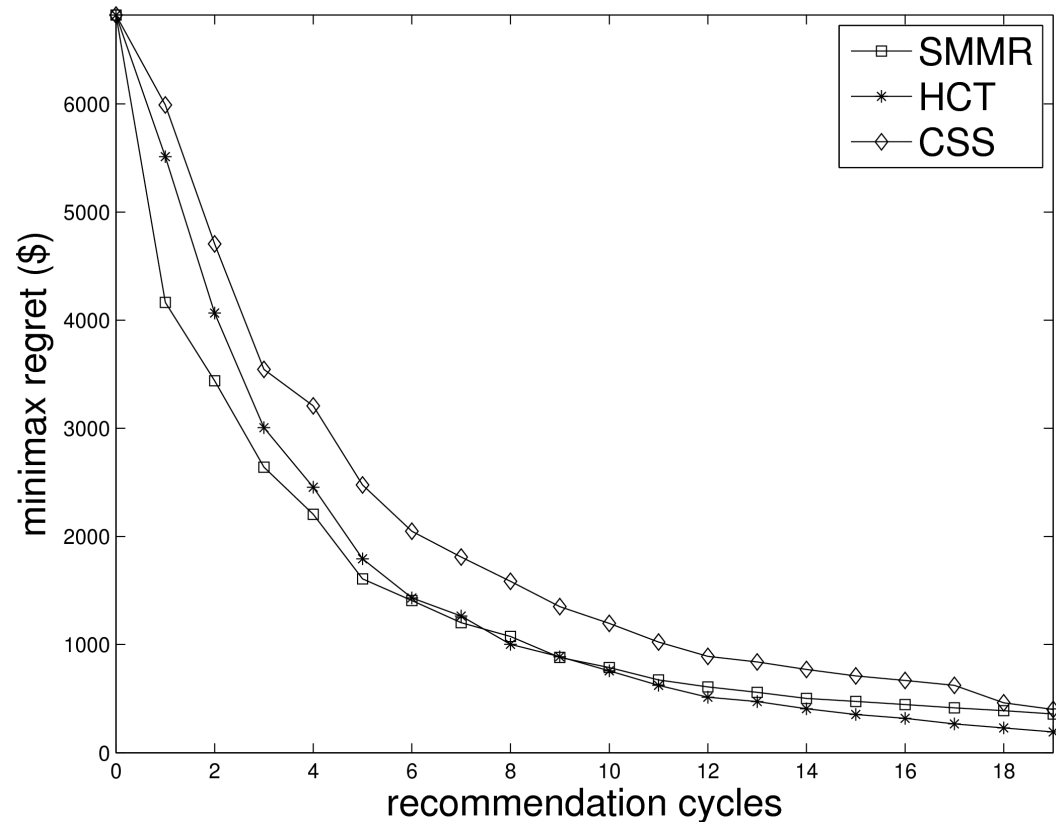
- Use *setwise* notion of adversary

$$\text{SMR-Adv}(Z, W) = \text{argmax}_y \text{SMR}(Z, y, W)$$

$$\left\{ \begin{array}{l} x^1 = x^*_w \\ x^i = \text{SMR-Adv}(\{x^1, \dots, x^{i-1}\}) \end{array} \right.$$

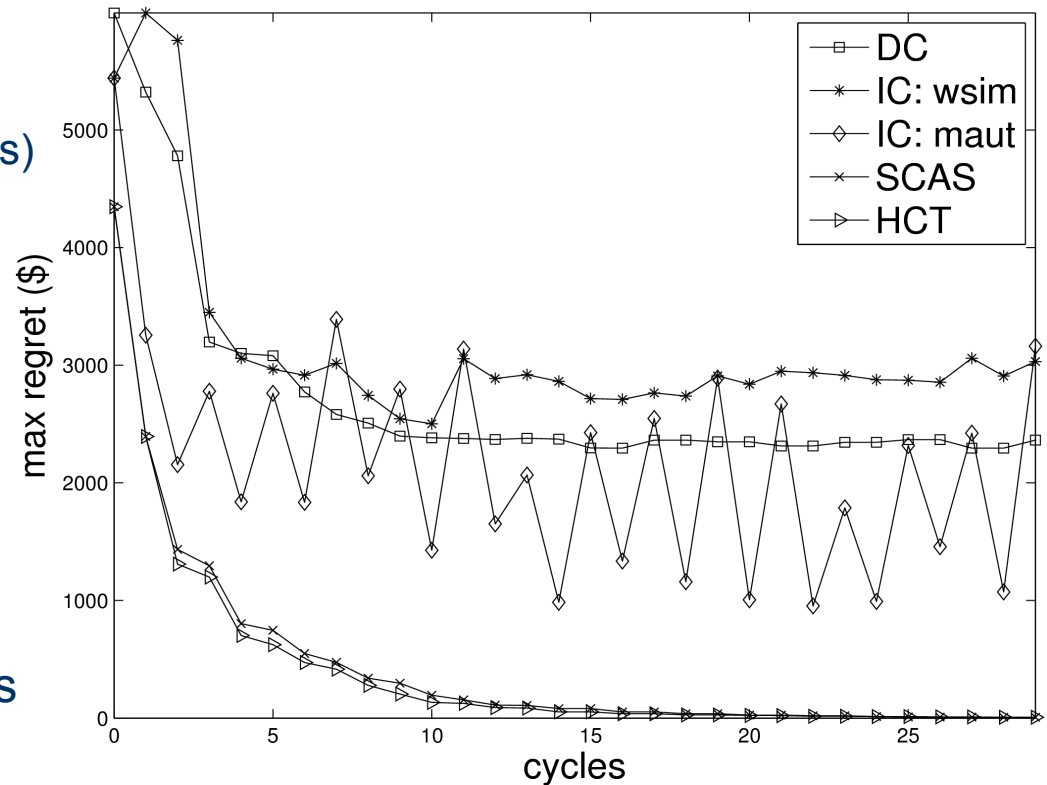
Empirical Results

- 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



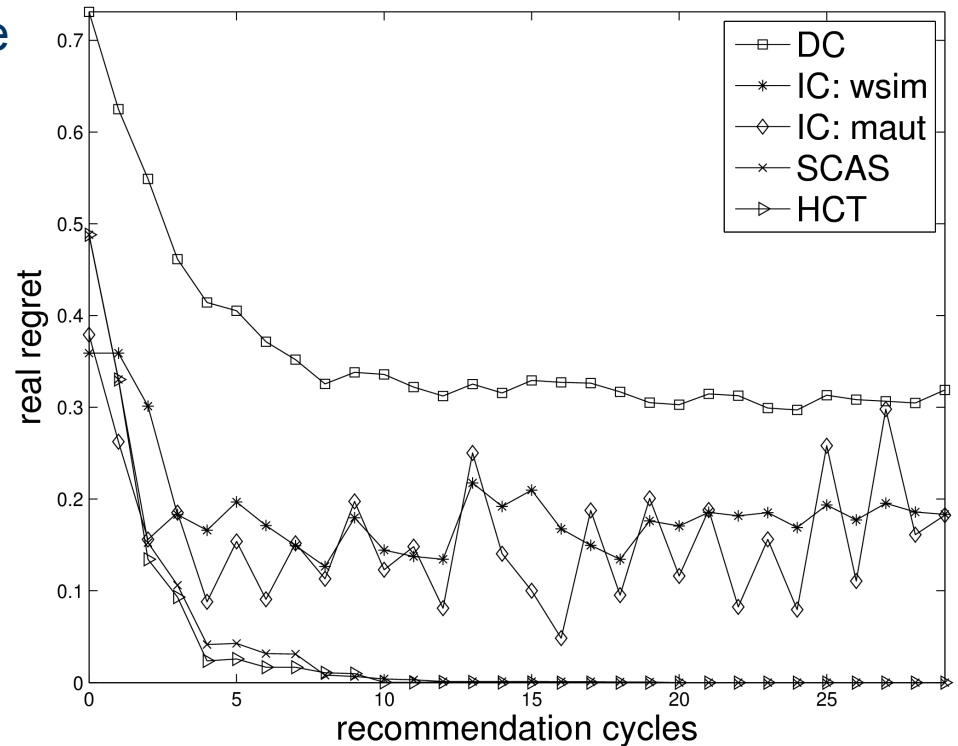
Critiquing Simulation

- 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 give 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!)

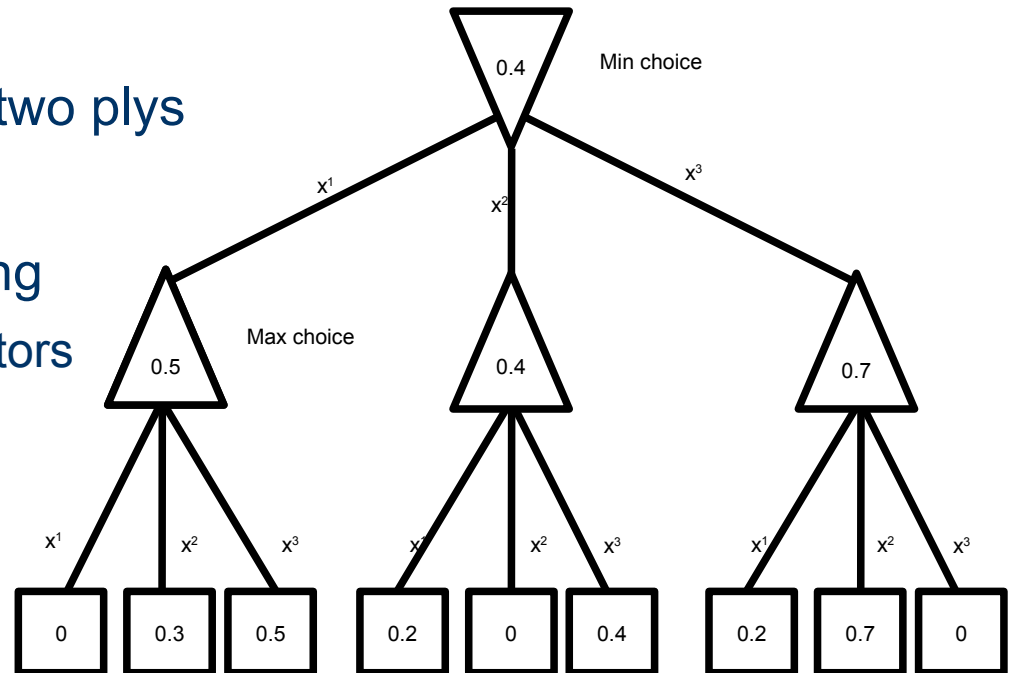
(Single Item) Minimax Regret Computation

■ Configuration problems

- Benders' decomposition and constraint generation to break minimax program

■ Discrete datasets

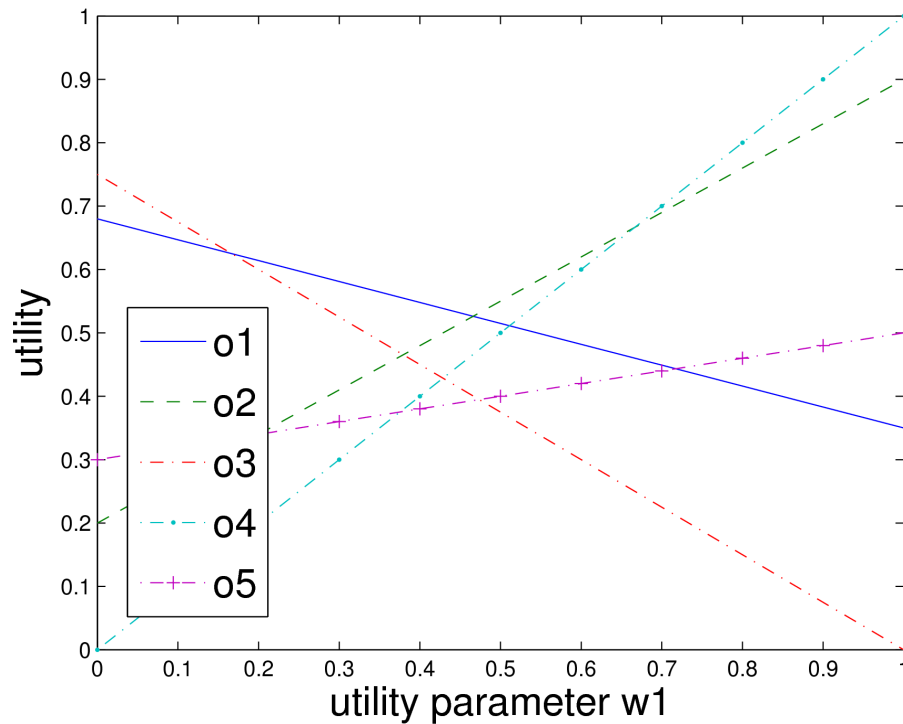
- Adversarial search with two plys
- Heuristics:
 - order to maximize pruning
 - Sample hypercube vectors



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 $UB = LB$

	Feature 1	Feature 2
o_1	0.35	0.68
o_2	0.9	0.2
o_3	0	0.75
o_4	1	0
o_5	0.5	0.3



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

w_1 unknown

	Adversary	MR
o_1	o_4	0.65
o_2	o_3	0.55
o_3	o_4	1
o_4	o_3	0.75
o_5	o_4	0.5

o_5 minimax regret optimal