Choice Theory and Interactive Utility Elicitation

Paolo Viappiani <paolo.viappiani@lip6.fr>

Outline

1. Some notions of Choice theory
2. Interactive elicitation: Bayesian approach
3. Interactive elicitation: minimax regret

Probabilistic Choice Models

Assumption: choice behavior is probabilistic.

Consider the following selection probability:

$$P(\{\text{apple}, \text{banana}\} \rightarrow \text{apple}) = 0.8$$

Link between the utility of an option and the probability with which it is chosen.

Change of preferences OR noise in choice ?

Probabilistic Choices

It is not entirely sure what the decision maker will choose.

DM's choice process as a two-step process:

1. the decision maker assesses the utility of each option
2. she makes a choice by maximizing utility

Choice Models vs Random Utility Models

Luce's Choice Axiom (LCA)

Let $P(A \rightarrow S)$ with $S \subset A$ be the probability that the chosen alternative in A is an element of S

Luce's Choice Axiom

Let A be the set of all alternatives.

1. For any set $S \subset A$ such that there is no $y \in S$ such that $P(A \rightarrow y) = 0$:

$$P(A \rightarrow x) = P(A \rightarrow S)P(S \rightarrow x)$$

   for all $x \in S$.

2. For any set $S \subseteq A$ such that there is $y \in S$ such that $P(A \rightarrow y) = 0$:

$$P(S \rightarrow x) = P(S - \{y\} \rightarrow x)$$

   for all $x \in S$ such that $P(A \rightarrow x) > 0$.

Mr Simpson and his Choice of Wine

Possible options: set $A = R \cup W$ consists in two red wines $R = \{r_1, r_2\}$ and two white wines $W = \{w_1, w_2\}$.

- Indifferent between red or white wines: $P(A \rightarrow R) = P(A \rightarrow W) = 0.5$.
- Prefer r_1 to r_2 60% of the times: $P(R \rightarrow r_1) = 0.6$
- Prefer w_1 to w_2 70% of the times: $P(W \rightarrow w_1) = 0.7$

Probability $P(A \rightarrow r_1) = P(A \rightarrow R)P(R \rightarrow r_1) = 0.5 \cdot 0.6 = 0.3$.
Probability $P(A \rightarrow r_2) = P(A \rightarrow R) \cdot P(R \rightarrow r_2) = 0.2$.
Probability $P(A \rightarrow w_1) = P(A \rightarrow W) \cdot P(W \rightarrow w_1) = 0.35$.
Probability $P(A \rightarrow w_2) = P(A \rightarrow W) \cdot P(W \rightarrow w_2) = 0.15$.

Ratio Scale Representation

Main property: assuming LCA, then there exists $v(\cdot)$ such that, for any subset S of A:

$$P(S \to x) = \frac{v(x)}{\sum_{y \in S} v(y)}$$

with v that can be rescaled by a multiplicative factor.

Proof sketch

Let $S \subset A$. Assuming $P(A \to S) \neq 0$ then

$$P(S \to x) = \frac{P(A \to x)}{P(A \to S)}.$$

Assume $v(x) = k \cdot P(A \succ x)$ with $k > 0$. From probability calculus:

$$P(S \to x) = \frac{k \cdot P(A \to x)}{\sum_{y \in B} k \cdot P(A \to y)} = \frac{v(x)}{\sum_{y \in S} v(y)}$$

Example 1

set A	$P(A \to a)$	$P(A \to b)$	$P(A \to c)$
a,b,c	0.6	0.3	0.1
a,b	0.66	0.33	-
a,c	0.86	-	0.14
b,c	-	0.75	0.25

The choice probabilities are consistent with $v(a) = 6$, $v(b) = 3$, $v(c) = 1$. Indeed:

$$P(\{a, c\} \to a) = \frac{6}{6+1} \approx 0.86.$$

We check LCA directly:
$P(\{a, b, c\} \to a) = P(\{a, b, c\} \to \{a, c\})P(\{a, c\} \to a)$?

$$P(\{a, b, c\} \to a) = 0.6$$
$$P(\{a, b, c\} \to \{a, c\})P(\{a, c\} \to a) = [0.6 + 0.1]0.86 \approx 0.6$$

Example 2

set A	$P(A \to a)$	$P(A \to b)$	$P(A \to c)$
a,b,c	0.4	0.4	0.2
a,b	0.6	0.4	-
a,c	0.66	-	0.33
b,c	-	0.66	0.33

$$P(\{a,b,c\} \to a) = 0.4$$
$$P(\{a,b,c\} \to \{a,b\}) P(\{a,b\} \to a) = [0.4 + 0.4]0.6 = 0.48$$

$$\implies P(\{a,b,c\} \to a) \neq P(\{a,b,c\} \to \{a,b\}) P(\{a,b\} \to a)$$

Therefore, LCA does hold in this example!

Independence from irrelevant alternatives (IIA)

Constant Ratio Rule

Let $p_{x,y} = P(\{x, y\} \rightarrow x)$. We have, if LCA holds:

$$\frac{p_{x,y}}{p_{y,x}} = \frac{P(A \rightarrow x)}{P(A \rightarrow y)}.$$

This means that elements other than x and y do not have an impact in this ratio.
(also called *indipendence from context*)

Other Properties

Product rule

$$\frac{p_{x,y}}{p_{y,x}} \frac{p_{y,z}}{p_{z,y}} \frac{p_{z,x}}{p_{x,z}} = 1.$$

and equivalently:

$$\frac{p_{x,z}}{p_{z,x}} = \frac{p_{x,y}}{p_{y,x}} \frac{p_{y,z}}{p_{z,y}}.$$

The probability of choosing x over z is determined by the probabilities of preferring x over y and y over z.

Debreu's Counterexample

- Mr Simpson is indifferent between seafood and meat
- He is also indifferent between two different types of steak

x=lobster, y= steak1, z=steak2
$A = \{x, y, z\}$; $B = \{x, y\}$; $B' = \{y, z\}$

$$P(A \to x) = \frac{1}{2}$$

$$P(A \to B) = 1 - P(A \to z) = 1 - P(A \to B')P(B' \to z) = 1 - \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}$$

$$\Longrightarrow$$

$$P(A \to x) = P(A \to B)P(B \to x) = \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8}$$

incompatibility with *similarity effect*

LCA as a *normative* requirement for identifying the alternatives.

Logistic Models

Assume $u = log(v)$ and equivalently $v = e^{u(x)}$. Then the probability of selection is:

$$P(A \to x) = \frac{e^{u(x)}}{\sum_y e^{u(y)}}$$

- For two items this is a logistic: (connection with logistic regression)

$$P(\{x, y\} \to x) = \frac{e^{u(x)}}{e^{u(x)} + e^{u(y)}} = \frac{1}{1 + e^{-(u(x)-u(y))}}$$

  that means: $u(x) >> u(y) \implies P \to 1$
  $u(x) = u(y) \implies P = \frac{1}{2}$
  $u(x) << u(y) \implies P \to 0$

- linear in the log-odds

$$\log \frac{P(\{x, y\} \to x)}{P(\{x, y\} \to y)} = u(x) - u(y)$$

  Multinomial Logit Model: $u(x)$ scalar product of parameters and attributes

From Choices to Rankings

Choice models can be used to derive models about *rankings*.

- Ranking as a repeated choice: *multi stage* model
- Probability of object i ranked first

$$\frac{v_i}{v_1 + \ldots + v_m}$$

- Probability of ranking $(1, 2, 3, 4)$: probability of choosing 1 from $\{1, 2, 3, 4\}$, times the probability of choosing 2 from $\{2, 3, 4\}$, etc.

$$\frac{v_1}{v_1 + v_2 + v_3 + v_4} \cdot \frac{v_2}{v_2 + v_3 + v_4} \cdot \frac{v_3}{v_3 + v_4}$$

Probabilistic Ranking Models

- Space of possible rankings of m objects: $m!$ (permutation polyhedron)
- Specifying a probability distrubution needs to assign a probability value to each possible ranking $\Longrightarrow$ need for a compact representation

Notation:

- m objects: $1, 2, \ldots, m$
- A ranking is a permutation $\pi : \{1, \ldots, m\} \rightarrow \{1, \ldots, m\}$
  $\pi(i)$ means "the position of object i"
  $\pi^{-1}(i)$ means "the object in position i"

Plackett-Luce model

- Defined by a vector of m parameters $v_1, \ldots, v_m$ with $v_i \geq 0$ and $\sum_{i=1}^{m} v_i = 1$.
- $P(\sigma) = \prod_{i=1}^{m} \frac{v_{\sigma^{-1}(i)}}{\sum_{j=i}^{m} v_{\sigma^{-1}(j)}}$

Distance-based Probabilistic Ranking Models

Distance on rankings

Assume d distance (metric) between permutations such that

- $d(\sigma, \tau) \geq 0 \; \forall \sigma, \tau$
- $d(\sigma, \tau) = 0 \iff \sigma = \tau$
- Triangular inequality: $d(\sigma, \rho) \leq d(\sigma, \tau) + d(\tau, \rho)$
- Simmetry: $d(\tau, \rho) = d(\rho, \tau)$

Assume a "central" ranking τ_0, formalize the probability of a ranking τ as:

$$P(\tau) = \frac{1}{Z} e^{-\theta d(\theta, \tau_0)}$$

where θ is a parameter to control the "dispersion" of the distribution.

Distances between Permutations

- Hamming distance: number of items that are not in the same position
  $d_H(\pi, \rho) = \sum_1^m I[\pi(i) \neq \rho(i)]$
- Spearman metric: sum of squares of the displacements

$$\hat{d}_S(\pi, \rho) = \sum_1^m (\pi(i) - \rho(i))^2$$

  to obtain a metric, $d_S = \sqrt{\hat{d}_S(\pi, \rho)}$.
- Spearman's footrule: sum of absolute value of the displacements

$$d_F(\pi, \rho) = \sum_1^m |\pi(i) - \rho(i)|$$

- Kendall tau: number of pairwise disagreement
  (minimum number of adjacent swaps)
- Cayley metric
- ...

Spearman distance

Spearman distance is defined as taking the squares of the differences:

$$d_S(\pi,\sigma) = \sum_{j=1}^{n}[\pi(j) - \sigma(j)]^2. \tag{1}$$

Note that Spearman can be expressed as follows:

$$d_S(\pi,\sigma) = \frac{n(n+1)(2n+1)}{3} - 2\sum_{i=1}^{n}\pi(i)\sigma(i).$$

Observation

Spearman distance d_S is a semimetric.

The connection between Borda and Spearman Distance

The Spearman distance characterizes the Borda rule:
$\pi^*_{Borda} = \arg\min_{\pi \in S_n} \sum_{u=1}^{m} d_S(\pi,\sigma_u)$.

Cfr. Theorem 2.2 in [John I Marden. Analyzing and modeling rank data. CRC Press, 1996].

Proof

$$d_S(\pi,\sigma) = \frac{n(n+1)(2n+1)}{3} - 2\sum_{i=1}^{n}\pi(i)\sigma(i).$$

Sum of distances with respect to input rankings $\sigma_1, ..., \sigma_m$:

$$D_S(\pi;\sigma_1,...,\sigma_m) = \sum_{u=1}^{m} d_S(\pi,\sigma_u) = 2\left(m\,C_n - \sum_{u=1}^{m}\sum_{i=1}^{n}\pi(i)\sigma_u(i)\right).$$

Therefore the ranking with minimum total Spearman distance with respect to a set of rankings $\sigma_1, .., \sigma_m$ is

$$\arg\min_{\pi} D_S(\pi;\sigma_1,...,\sigma_m) = \arg\max_{\pi}\sum_{i=1}^{n}\sum_{u=1}^{m}\pi(i)\sigma_u(i) =$$
$$= \arg\max_{\pi}\sum_{i=1}^{n}\pi(i)\sum_{u=1}^{m}\sigma_u(i).$$

Hence, π^* should be such that i precedes j if $\sum_{u=1}^{m}\sigma_u(i) < \sum_{u=1}^{m}\sigma_u(j)$. But that is exactly the ordering induced by Borda count!

Positional Spearman

We define Positional Spearman as a generalization of Spearman distance giving different weights to rank positions, computed as

$$d_{PS}(\pi, \sigma) = \sum_{i=1}^{n} [w(\pi(i)) - w(\sigma(i))]^2 \tag{2}$$

parametrized by a vector w.

Observation

The positional ranking distance d_{PS} is a semimetric.

Characterization

Let w such that all weights are different; $w(r) \neq w(s)$ if $r \neq s$ with $r, s \in \{1, ..., n\}$. The positional Spearman distance with weights w characterizes the scoring rule with the same weights: $\pi^*_{SR} = \arg\min_{\pi \in S_n} \sum_{u=1}^{m} d_{PS}(\pi, \sigma_u)$.

The problem of non strictly decreasing weights

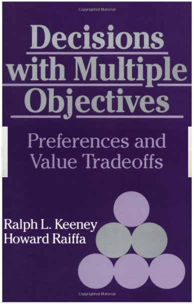
- The characterization only holds for scoring rules with distinct weights.
- Why identical weights are problematic?
  - Consider, for instance, $w = (3, 2, 2, 1)$
  - $d_{PS}(\langle 1, 2, 3, 4\rangle, \langle 1, 3, 2, 4\rangle) = 0.$
- The characterization fails for plurality, veto and top-k represented as scoring rules with weights $(1, 0, ..., 0)$, $(0, ..., 0, 1)$ and $(1, ..., 1, 0, .., 0)$.
- We therefore look for an alternative characterization for these rules.
- Using plurality in our framework for clustering basically means to put together rankings based on the first preferred item.
  - Aggregation will be made by ordering items according to the number of "votes" (number of rankings placing an item first); when assigning rankings to clusters, a ranking with item i in the first position will be assigned to the cluster whose centroid put item i in the highest position.

Elicitation: Overview

- Traditional approaches
- Bayesian utility elicitation
- Elicitation with minimax regret

Classic Approaches for Utility Elicitation

- Assessment of multi attribute utility functions
  - Typically long list of questions
  - Focus on high risk decision
  - Goal: learn utility parameters (weights) up to a small error
- Which queries?
  - Local: focus on attributes in isolation
  - Global: compare complete outcomes
- Standard gamble queries, bound queries, ...



Basic Elicitation: Flat Representation

- Typical approach to assessment
  - Normalization: for best outcome $\mathbf{x}^\top$ and worst outcome $\mathbf{x}^\perp$, set: $u(\mathbf{x}^\top) = 1$ and $u(\mathbf{x}^\perp) = 0$.
  - Standard Gamble Queries (SGQ)
    - Choose between option x_0 for sure or a gamble $SG(l) =< x^\top, l, x_\perp, 1-l >$ (best option $\mathbf{x}^\top$ with probability l, worst option $\mathbf{x}_\perp$ with probability $1-l$)
    - $u(\mathbf{x}) = lu(\mathbf{x}^\top) + (1-l)u(\mathbf{x}^\perp) = l.$
    - *"Would you prefer to have car number 3 or a lottery where you get car number 10 with probability* 0.85 *and car number 22 with probability* 0.15 *?"*
    - Require precise numerical assessments
- Bound queries: fix a probability l, ask if $\mathbf{x}$ preferred to $SG(l)$
  - Yes/no responses place a lower/upper bound on utility.
  - Less informative

Additive Utility Functions

- Additive representation (common in MAUT)
- Sum of local utility functions u_i over attributes (or local value functions v_i multiplied by scaling weights)
- Exponential reduction in the number of needed parameters

$$u(\mathbf{x}) = \sum_{i=1}^{n} u_i(x_i) = \sum_{i=1}^{n} \alpha_i v_i(x_i) \tag{3}$$

Color	v_1
red	1.0
blue	0.7
green	0.0

Shape	v_2
round	0
square	0.2
star	1

Importance for attribute "color": $\alpha_1 = 0.2$, for "shape": $\alpha_2 = 0.3$.
Notice: many algorithms in the recommender system community (for example matrix factorization techniques) implicitly *assume an additive model!*

Generalized Additive Utility

- Sum of local utility functions u_I over *sets of attributes* (or local value functions v_i multiplied by scaling weights)
- Higher descriptive power than strictly additive utilities, while still having a manageable number of parameters

$$u(\mathbf{x}) = \sum_{i=1}^{m} u_{J_i}(x_{J_i}) = \sum_{i=1}^{n} \alpha_{J_i} v_{J_i}(x_{J_i}) \tag{4}$$

where J_i is a set of indices, x_{J_i} the projection of $\mathbf{x}$ on J_i and m the number of factors.

Color	**Shape**	$v_{color,shape}$
red	round	0.9
red	square	1.0
red	star	0.5
blue	round	0.4
...	...	...

Position	$v_{position}$
top	1
bottom	0

Importance for factor "color+shape": $\alpha_{J_1} = 0.2$, for "position": $\alpha_{J_2} = 0.3$.

(Standard) Elicitation of Additive Models

- Local standard gamble queries assess local value functions
  - Answers asses the values of the local value function on attribute i:

$$x_i \approx < l, x_i^{\top}; 1-l, x_i^{\perp} > \iff v_i(x_i) = l$$

  - For example $\text{blue} \approx < 0.85, \text{ red}; 0.15, \text{ grey} >$.
- Local bound queries refine intervals on local utility.

(Standard) Elicitation of Additive Models: Scaling Factors

- Scaling factors are assessed with "global queries"
  - One needs to define a *reference* outcome
  - For example, the user's current car: (red, 2doors, 150hp,...)
  - Define $\mathbf{x}^{\top_j}$ by setting attribute X_j to the best value, keeping the other at the reference value.
    - E.g. for doors: (red, 4doors, 150hp,...)
    - By independence, best value must be fixed
  - Compute scaling factors $\lambda_j = u(\mathbf{x}^{\top_j}) - u(\mathbf{x}^{\perp_j})$.
  - Assess these utility values with global SG queries.
- This gives $u(\mathbf{x}) = \sum_1^n u_i(x_i) = \sum_{i=1}^n \lambda_i v_i(x_i)$.

(Standard) Elicitation of Additive Models

- Consider an attribute (for example, color)
  - Ask for the best value (say, red)
  - Ask for worst value (gray)
  - Ask local standard gable for each remaining color to assess it local utility value (*value function*)
- Scaling factors
  - Define reference outcome
  - Ask global queries in order to assess the difference in utility occurring when, starting from the reference outcome, “moving” a particular attribute to the best / worst

Automated Elicitation vs Classic Elicitation

Problems with the classic view

- Standard gamble queries (and similar queries) are difficult to respond
- Large number of parameters to assess
- Unreasonable precision required
- Cognitive or computational cost may outweigh benefit

Automated Elicitation and Recommendation

Important points:

- Cognitively plausible forms of interaction
- Incremental elicitation until a decision is possible
- We can often make optimal decisions without full utility information
- Generalization across users

Adaptive Utility Elicitation

Utility-based Interactive Recommender System:

- *Bel*: belief about the user's utility function u
- $Opt(Bel)$: optimal decision given incomplete beliefs about u

Algorithm: Adaptive Utility Elicitation

1. **Repeat** until *Bel* meets some termination condition
   1. Ask user some query
   2. Observe user response r
   3. Update *Bel* given r
2. Recommend $Opt(Bel)$

Types of Beliefs

- *Probabilistic Uncertainty:* distribution of parameters, updated using Bayes
- *Strict Uncertainty:* feasible region (if linear constraints: convex polytope)

Which Recommendations? Which Queries?

Several Framewors

The questions:

- How to represent the uncertainty over an utility function?
- How to aggregate this uncertainty? Make recommendations ?
- How to choose the next query? How to recommend a set, a ranking, ...

	Regret-based	Bayesian
Knoweldge representation	constraints	prob. distribtion
Which option to recommend?	minimax regret	expected utility
Which query to ask next?	worst-case regret reduction	expected value of information

Possible other choices: Hurwitz criterion, and others.
Hybrid models: minimax regret with expected regret reduction, ...

The Bayesian view

Probabilities reflect the subjective belief about the user's utility function; Utilities are random variables

- $P(u)$: distribution over possible utility functions
  (in practice: assume u has parametric form and a distribution over parameters)

We can use expected utility to sort the alternatives

$$EU(x) = \int_x u(x)P(u)du$$

We recommend the item x^* with highest expected utility x^*

$$EU^* = \max_{x \in A} EU(x); \qquad x^* = \arg\max_{x \in A} EU(x)$$

The *expected loss* can be used as stopping criterion for elicitation:

$$\int [\max_{x \in X}\{u(x)\} - u(x)]P(u)du \tag{5}$$

Note: the item with highest expected utility achieves minimum expected loss

Response Models

Model the user's cognitive ability of answering correctly to a preference query

- $P(r|u)$: probability of an answer r given utility u (for a specific query)
- By marginalization we can obtain the a-priori estimate of an answer according to the current belief $P(r) = \int P(r|u)P(u)du$
- Update the distribution: $P(u|r) = \frac{P(r|u)P(u)}{P(r)} = \frac{P(r|u)P(u)}{\int P(r|u)P(u)du}$

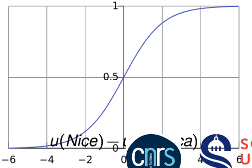
Example: query "do you prefer to go on vacation to Nice or in Corsica?"

- Noiseless response: $P(Nice|u) = I[u(Nice) > u(Corsica)]$
- Constant noise:

$$P(Nice|u) = \begin{cases} 1-\rho, & \text{if } u(Nice) > u(Corsica) \\ 0.5, & \text{if } u(Nice) = u(Corsica) \\ \rho, & \text{otherwise} \end{cases}$$

- Logistic noise:

$$P(Nice|u) = \frac{e^{\gamma u(Nice)}}{e^{\gamma u(Nice)} + e^{\gamma u(Corsica)}} = \\ = \frac{1}{1 + e^{-\gamma[u(Nice) - u(Corsica)]}}$$



Expected Utility of Selection

- It is typical to provide a set of recommended items to the user
- Decision-theoretic "diversity": recommended options should be highly preferred relative to a wide range of probable user utility functions
- $S \to x$: event of choosing x among items in set S

EUS is the expected utility of the chosen item (in expectation over utility and over the randomness in the selection) *[Viappiani and Boutilier, 2010]*

$$EUS(S) = \sum_{x \in S} p(S \to x) EU(x|S \to x)$$

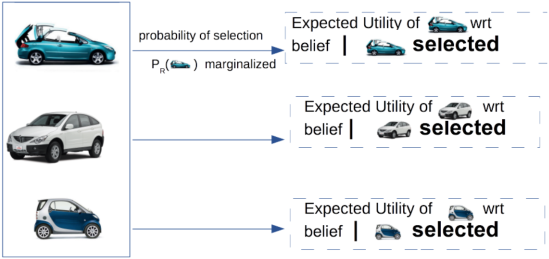
or alternatively

$$EUS(S) = \int [\sum_{x \in S} p(S \to x|u) u(x)] p(u) du$$

EUS is submodular with no noiseless reponse model and constaint noise, but not with the logistic noise model

Greedy strategies give strong worst-case guarantees [Nemhauser] [Krause & Guestrin]; very efficient optimization with lazy-evaluation [Minou...]

Expected Utility of Selection



$$EUS(S) = \sum_{x \in S} p(S \rightarrow x) EU(x|S \rightarrow x)$$

$$= EUS(S) = \int [\sum_{x \in S} p(S \rightarrow x|u) u(x)] p(u) du$$

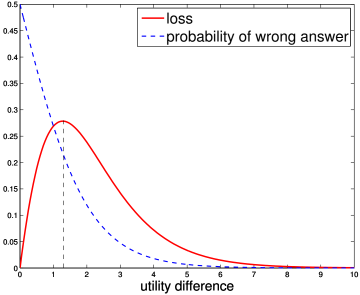
A Bound on the Effect of Logistic Noise

Question: "among $S = \{\text{Nice}, \text{Corsica}\}$ which one do you pref?"
Under logistic noise:

$$P(S \to \text{Nice}|u) = \frac{1}{1 + e^{-\gamma[u(Nice) - u(Corsica)]}}$$

Now assume that the user prefers Nice to Corsica ($u(\text{Nice}) > u(\text{Corsica})$), the probability of a "mistake" is:

$$P(S \to \text{Corsica}|u) = 1 - \frac{1}{1 + e^{-\gamma[u(Nice) - u(Corsica)]}} = \frac{1}{1 + e^{\gamma[u(Nice) - u(Corsica)]}}$$



- Let $d = u(Nice) - u(Corsica)$.
- When $d \nearrow$ the probability of a mistake $\searrow$.
- When $d \nearrow$ the "impact" of a mistake $\nearrow$.
- The loss $d \cdot \frac{1}{1+e^{\gamma d}}$ is maximized for a specific point $\hat{d}$.
- The maximum loss is $\frac{1}{\gamma} W(\frac{1}{e}) \approx 0.279 \frac{1}{\gamma}$ (where W is Lambert-W function).

What Query to Ask Next?

- The problem can be modeled as a POMDP *[Boutilier, AAAI 2002]*, however impractical to solve for non trivial cases
- Idea: use Entropy to score potential queries [Abbas, 2004]
- Better Idea: ask query with highest "value", a posteriori improvement in decision quality

Value of Information

In a Bayesian approach (Myopic) *Expected Value of Information*

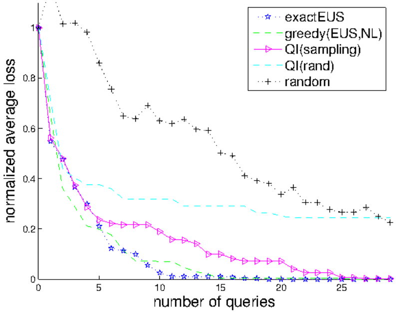
$$EVOI(q) = \sum_{r \in R_q} P(r) EU^*_{u \sim P(u|r)} - EU^*_{u \sim P(u)}$$

where R is the set of possible responses (answers)

Ask query $q^* = \arg\max EVOI(q)$ with highest EVOI

Efficient computation of the myopically optimal query *[Viappiani and Boutilier, 2010]* for choice queries exploiting the link with the criterion *EUS*

Experimental Results (Bayesian Elicitation)



We know that the myopic value of information of the queries posed by the greedy strategies are close to the optimum. Empirical results show that they are also effective when evaluated in an iterative process; the recommended item converges to the "true" best item in few cycles.

Which Recommendations? Which Queries?

Several Framewors

The questions:

- How to represent the uncertainty over an utility function?
- How to aggregate this uncertainty? Make recommendations ?
- How to choose the next query? How to recommend a set, a ranking, ...

	Minimax Regret	Bayesian
Knoweldge representation	constraints	prob. distribtion
Which option to recommend?	expected regret	utility
Which query to ask next?	worst-case regret reduction	expected value of information

Possible other choices: Hurwitz criterion, and others.
Hybrid models: minimax regret with expected regret reduction, ...

Strict Utility Function Uncertainty

- User's utility parameters w unknown
- Assume a feasible set W

$$u(downtown, 2rooms, parking) > 0.4$$
$$u(downtown, 2rooms, parking) > u(eastside, 4rooms, no - parking)$$

- Allow for unquantified or "strict" uncertainty (if linear constraints, the representation is a polytope)

Minimax Regret under Utility Uncertainty

- The insight is to use minimax regret in order to make recommendations in face of *utility uncertainty*.
- The idea is to recommend the item/choice that has lowest regret with respect to the true best choice.
- We have seen how decision tables can model decisions where outcomes are dictated by both our choice and "nature's" pick of the state of the world.
- Since many possible utility functions are usually possible (often continuous); we'll define minimax regret in terms of pairwise max regrets.

Minimax Regret

Intuition

Adversarial game; the recommender selects the item reducing the "regret" wrt the "best" item when the unknonw parameters are chosen by the adversary

- Robust criterion for decision making under uncertainty [Savage; Kouvelis]
- Showed to be effective when used for decision making under utility uncertainty [Boutilier et al., 2006] and as driver for elicitation

Advantages

- No heavy Bayesian updates
- No prior assumption required
- MMR computation suggests queries to ask to the user

Limitations

- No account for noisy responses
- Formulation of the optimization depends on the assumption about the utility

Minimax Regret

Assumption: a set of feasible utility functions W is given

The **pairwise max regret**

$$\mathrm{PMR}(\mathbf{x}, \mathbf{y}; W) = \max_{w \in W} \; u(\mathbf{y}; w) - u(\mathbf{x}; w)$$

The **max regret**

$$MR(\mathbf{x}; W) = \max_{\mathbf{y} \in \mathbf{X}} \; \mathrm{PMR}(\mathbf{x}, \mathbf{y}; W)$$

The **minimax regret**

MMR(W) of W and the **minimax optimal item $\mathbf{x}^*_W$:**

$$\begin{aligned} \mathrm{MMR}(W) &= \min_{\mathbf{x} \in \mathbf{X}} \; \mathrm{MR}(\mathbf{x}, W) \\ \mathbf{x}^*_W &= \arg\min_{\mathbf{x} \in \mathbf{X}} \; \mathrm{MR}(\mathbf{x}, W) \end{aligned}$$

Properties of Minimax Regret

- $\text{PMR}(\mathbf{x}, \mathbf{x}; W) = 0$
- $\text{PMR}(\mathbf{x}, \mathbf{y}; W)$ is in general different from $\text{PMR}(\mathbf{y}, \mathbf{x}; W)$
- $\text{PMR}(\mathbf{x}, \mathbf{y}; W) > 0$ means that **y** can be preferred to **x**.
- If $\text{PMR}(\mathbf{x}, \mathbf{y}; W) > 0$ and $\text{PMR}(\mathbf{y}, \mathbf{x}; W) < 0$ than **y** is necessarily preferred to **x**
- If $\text{PMR}(\mathbf{x}, \mathbf{y}; W) = 0$ and $\text{PMR}(\mathbf{y}, \mathbf{x}; W) = 0$ than **y** necessarily have the same utility
- $\text{MMR}(W) = 0$ iff the best choice has been found with certainty

Example

item	*feature1*	*feature2*
a	10	14
b	8	12
c	7	16
d	14	9
e	15	6
f	16	0

Linear utility model with normalized utility weights ($w_1 + w_2 = 1$);
$u(x; w) = (1 - w_2)x_1 + w_2 x_2 = (x_2 - x_1)w_2 + x_1$

Notice: it is a 1 dimensional problem

Initially, we only know that $w_2 \in [0, 1]$

$\mathrm{PMR}(a, f; w_2) = \max_{w_2} u(f; w_2) - u(a; w_2)$
$= \max_{w_2} 6(1 - w_2) - 14w_2 = \max_{w_2} 6 - 20w_2$
$= 6$ (for $w_2 = 0$)

$\mathrm{PMR}(a, b; w_2) = \max_{w_2} u(b; w_2) - u(a; w_2) < 0$
(a dominates b; there can't be regret in choosing a instead of b!)

$\mathrm{PMR}(a, c; w_2) = \max_{w_2} -3(1 - w_2) + 2w_2$
$= \max_{w_2} 5w_2 - 3 = 2$ (for $w_2 = 1$)

....

Example (continued)

item	*feature1*	*feature2*
a	10	14
b	8	12
c	7	16
d	14	9
e	15	6
f	16	0

Linear utility model with normalized utility weights ($w_1 + w_2 = 1$);
$u(x; w) = (1 - w_2)x_1 + w_2 x_2 = (x_2 - x_1)w_2 + x_1$

Notice: it is a 1 dimensional problem

Computation of the pairwise regret table.

PMR$(\cdot, \cdot)$	*a*	*b*	*c*	*d*	*e*	*f*	MR
a	0	-2	2	4	5	**6**	**6**
b	2	0	4	6	7	**8**	8
c	3	1	0	7	8	**9**	9
d	5	3	**7**	0	1	2	7
e	8	6	**10**	3	0	1	10
f	14	12	**16**	9	6	0	16

The MMR-optimal solution is *a*, adversarial choice is *f*, and minimax regret value is 6.

In reality no need to compute the full table (tree search methods) [Braziunas, PhD Thesis, 2011]

Minimax Regret Computation

Computation of Pairwise Max Regret as Linear Program

- Objective function for computing PMR: $\max_{w \in W} w \cdot (\mathbf{y} - \mathbf{x})$
- Usually W expressed by linear constraints such as $w \cdot \mathbf{z}^{+} \geq w \cdot \mathbf{z}^{-}$ for $(\mathbf{z}^{+}, \mathbf{z}^{-}) \in \mathcal{D}_{pref}$ (a set of comparisons)

Computation of Minimax Regret

- Naive approach: test all $n^2 - n$ combinations of choices
- Better idea: implement a *search* problem [Braziunas, 2012]
  - i=choice of recommender, j=choice of adversary
  - UB: upper bound on minimax regret (max regret of best solution found)
  - LB_i: lower bound on the max regret of option i
  - After testing i against j: $LB_i \leftarrow \max(LB_i, \text{PMR}(i, j))$
  - Whenever $LB_i \geq UB$: prune option i
- Empirically, a small number of PMR checks is needed

Example of Minimax Regret Computation

Example

Complete "pairwise max regret" table

$PMR(i,j)$	j=1	j=2	j=3	j=4	$MR(i)$
i=1	0	1	2	3	3
i=2	2	0	2	2	2
i=3	4	1	0	1	4
i=4	3	2	3	0	3

Evaluation (0 *PMR* checks)

$PMR(i,j)$	j=1	j=2	j=3	j=4	LB_i
i=1	0	?	?	?	0
i=2	?	0	?	?	0
i=3	?	?	0	?	0
i=4	?	?	?	0	0

UB
+Inf

Example of Minimax Regret Computation

Example

Complete "pairwise max regret" table

PMR(i,j)	j=1	j=2	j=3	j=4	MR(i)
i=1	0	1	2	3	3
i=2	2	0	2	2	2
i=3	4	1	0	1	4
i=4	3	2	3	0	3

Evaluation (1 *PMR* checks)

PMR(i,j)	j=1	j=2	j=3	j=4	LB_i
i=1	0	1	?	?	1
i=2	?	0	?	?	0
i=3	?	?	0	?	0
i=4	?	?	?	0	0

UB
+Inf

Example of Minimax Regret Computation

Example

Complete "pairwise max regret" table

$PMR(i, j)$	j=1	j=2	j=3	j=4	$MR(i)$
i=1	0	1	2	3	3
i=2	2	0	2	2	2
i=3	4	1	0	1	4
i=4	3	2	3	0	3

Evaluation (3 *PMR* checks)

$PMR(i, j)$	j=1	j=2	j=3	j=4	LB_i
i=1	0	1	2	3	3
i=2	?	0	?	?	0
i=3	?	?	0	?	0
i=4	?	?	?	0	0

UB
3

Example of Minimax Regret Computation

Example

Complete “pairwise max regret" table

$\mathrm{PMR}(i,j)$	j=1	j=2	j=3	j=4	$\mathrm{MR}(i)$
i=1	0	1	2	3	3
i=2	2	0	2	2	2
i=3	4	1	0	1	4
i=4	3	2	3	0	3

Evaluation (6 *PMR* checks)

$\mathrm{PMR}(i,j)$	j=1	j=2	j=3	j=4	LB_i
i=1	0	1	2	3	3
i=2	2	0	2	2	2
i=3	?	?	0	?	0
i=4	?	?	?	0	0

UB
2

Example of Minimax Regret Computation

Example

Complete "pairwise max regret" table

PMR(i,j)	j=1	j=2	j=3	j=4	MR(i)
i=1	0	1	2	3	3
i=2	2	0	2	2	2
i=3	4	1	0	1	4
i=4	3	2	3	0	3

Evaluation (7 *PMR* checks)

PMR(i,j)	j=1	j=2	j=3	j=4	LB_i
i=1	0	1	2	3	3
i=2	2	0	2	2	2
i=3	4	?	0	?	4
i=4	?	?	?	0	0

UB
2

Example of Minimax Regret Computation

Example

Complete "pairwise max regret" table

PMR(i, j)	j=1	j=2	j=3	j=4	MR(i)
i=1	0	1	2	3	3
i=2	2	0	2	2	2
i=3	4	1	0	1	4
i=4	3	2	3	0	3

Evaluation (8 *PMR* checks)

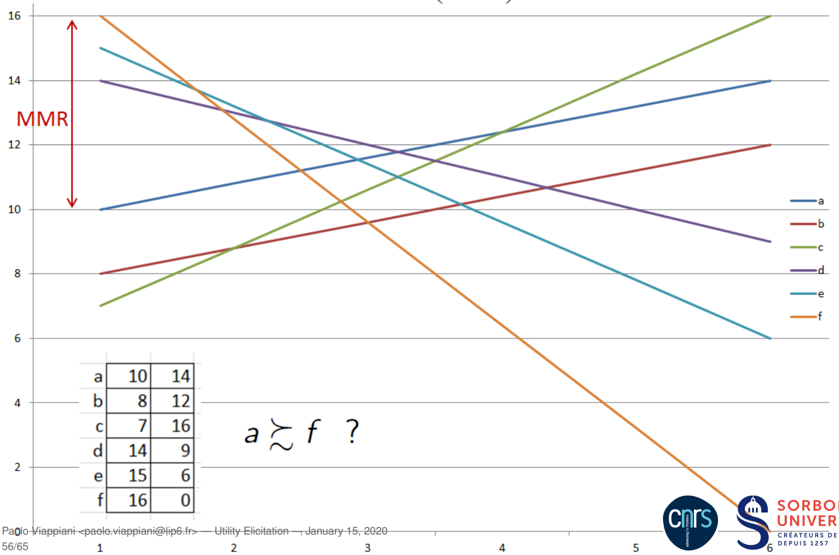
PMR(i, j)	j=1	j=2	j=3	j=4	LB_i
i=1	0	1	2	3	3
i=2	2	0	2	2	2
i=3	4	?	0	?	4
i=4	3	?	?	0	3

UB
2

Interactive Elicitation with Minimax Regret

- Assume, we want to ask a new query to improve the decision.
- A very successful strategy (thought generally not optimal!) is the *current solution strategy*: ask user to compare the option $\mathbf{x}^*$ minimising maxregret and its adversary $\mathbf{y}^a$.
- In the example, we would ask to compare a and f.

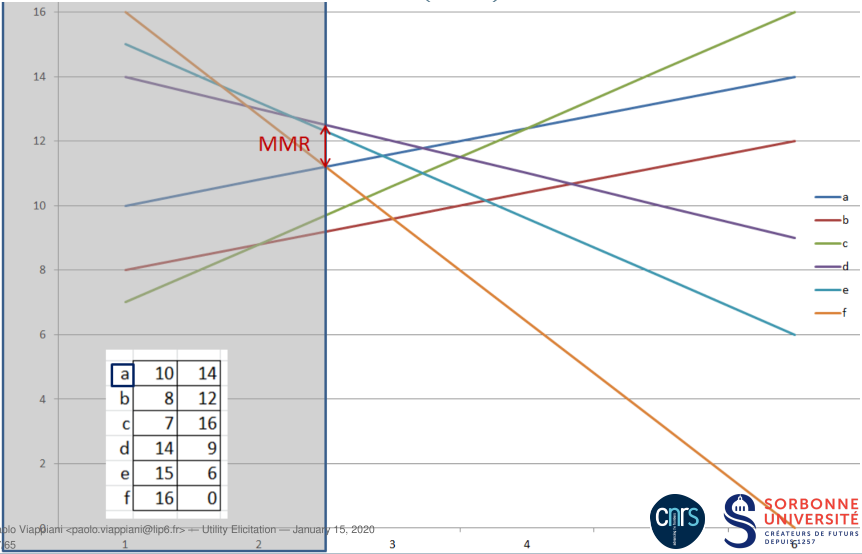
A graphical illustration for linear utility model (1/3)



a	10	14
b	8	12
c	7	16
d	14	9
e	15	6
f	16	0

$a \succsim f$?

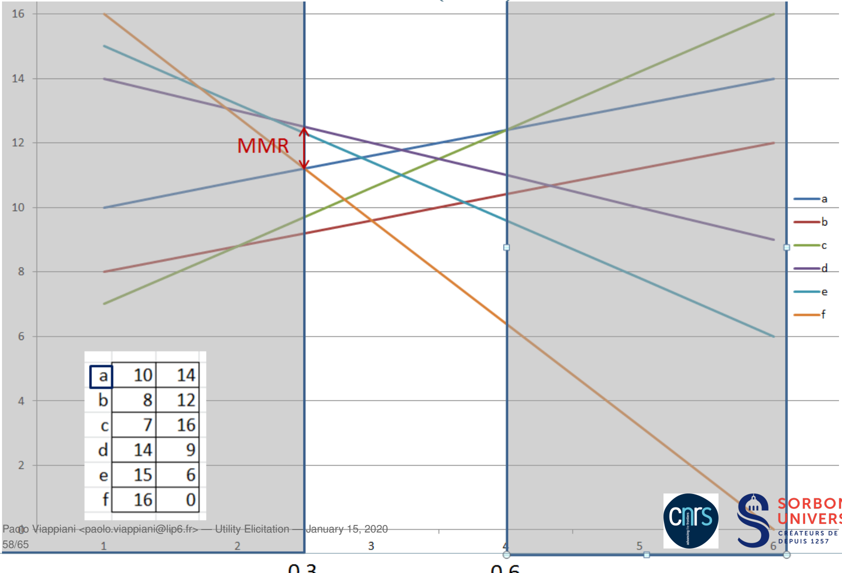
A graphical illustration for linear utility model (2/3)



$\boxed{a}$	10	14
b	8	12
c	7	16
d	14	9
e	15	6
f	16	0

0.3 0.6

A graphical illustration for linear utility model (3/3)



a	10	14
b	8	12
c	7	16
d	14	9
e	15	6
f	16	0

Criteria for Regret-based Elicitation

- Given $\mathbf{x}^*_W$ (the minimax regret option) and $\mathbf{x}^a$ (the adversary's choice),
- the current solution strategy asks the user to compare $\mathbf{x}^*$ and $\mathbf{x}^a$.
- One can define a criterion analogous to EVOI:
  - Let be W the current parameter space, q the query, $r_1, \ldots, r_k$ the possible ansers
  - $WR(q) = \max_i \mathrm{MMR}(W \cap [r_i])$
  - Where $[r_i]$ is the set of parameters satisfying r_i
- Note also that adding new constraints, the minimax regret cannot increase $\mathrm{MMR}(W) \geq \mathrm{MMR}(W \cap [r_i])$
- Therefore one can adopt $\mathrm{MMR}(W) < \tau$ as a threshold.

Minimax Regret in Combinatorial Domains

- Now, assume space of possible decisions **X** is combinatorial (for example, it may be implicitly defined as a constraint satisfaction problem)
- Set of choices/items **X** defined by integer constraints: for example $X_1 + X_2 < 1$
- Given u, recommendation can be formulated as an optimization problem.
- Issues with the computation of minimax regret: quadratic objective, minimax objective.

Benders' Decomposition

- Problem: how to write an optimization program that can handles minimax ?
- Decomposition: minimize R with R constrained to be *greater than* all regret values.

Here we consider a linear utility model $u(\mathbf{x};\mathbf{w}) = \mathbf{w}\cdot\mathbf{x}$.

$$\min_{x_1,\ldots,x_n,R} R$$
$$\text{s.t. } R \geq \mathbf{w}^a\mathbf{x}^a - \mathbf{w}^a\mathbf{x} = \sum w_i^a x_i^a - w_i^a x_i \quad \forall(\mathbf{w}^a,\mathbf{x}^a)$$
$$\mathbf{x} \in \mathbf{X}$$

- $\mathbf{w}^a\mathbf{x}^a$ is a constant from the point of view of the optimization.
- $\mathbf{x}^a = \max \mathbf{w}\cdot\mathbf{x}$ with $\mathbf{x}\in\mathbf{X}$ (the adversary will always choose the item with highest utility wrt $\mathbf{w}^a$).
- Problem not solvable as such (∞ number of constraints)
  - But in fact we need to consider only the $\mathbf{w}^a$ that are vertices of the polytope.
  - Still infinite, but we'll use constraint generation.

Constraint Generation

Intuition: solve a relaxed Integer Program (IP) with a subset of constraints; If any constraint violated at solution, add it and repeat.

Constraint Generation

- Constraint generation: avoid enumeration of V
  - **REPEAT**
  - Solve minimization problem with a subset GEN of Vertex(W)
    - The adversary's hands are tied to choose a couple $(\mathbf{w}, \mathbf{x}^a)$ from this subset
    - LB of minimax regret
    - $\mathbf{x}^*$ is the minimax-regret optimal of this relaxed problem
  - Find max violated constraint computing $MR(x)$
    - Adv chooses $\mathbf{w}^a$ in W and $\mathbf{x}^a \in X$ to maximize pairwise max regret of $\mathbf{x}$
    - UB of minimax regret
  - Add the $(\mathbf{w}^a, \mathbf{x}^a)$ to GEN
- Terminate **WHEN** UB = LB

The "Master" Problem

Note that utility parameters w_i are not a decision variable in this program!

$$\min_{x_1,\ldots,x_n,R} R$$

$$\text{s.t. } R \geq \mathbf{w}^a\mathbf{x}^a - \mathbf{w}^a\mathbf{x} = \sum w_i^a x_i^a - w_i^a x_i \quad \forall(\mathbf{w}^a, \mathbf{x}^a) \in \text{GEN}$$

$$\mathbf{x} \in \mathbf{X}$$

- Initialization: for instance pick a random **w**
- Solve "master" problem, obtain $\mathbf{x}^*$ minimizing regret with choice of adv limited in GEN
- Solve "slave" problem, maximizing regret when adv chooses in GEN
- If regret value is the same, stop, otherwise repeat.

The "Slave" Problem: Max Regret

- Max regret MR($\mathbf{x}$, W) computed as a mixed integrer linear program.
- Here the adversary can choose any $\mathbf{w}^a \in W$.
- The adversary chooses $\mathbf{x}^a = (X_1^a, \ldots, X_n^a)$. $\mathbf{x}$ is given.
- Integer programming tricks in order to linearize quadratic constraints.

$$\max_{\mathbf{w},\mathbf{x}^a} \sum_{i=1}^{n} W_i^a X_i^a - W_i^a x_i$$

linearised as:

$$\begin{aligned}
\max_{\mathbf{w}\in W,\mathbf{x}^a} \sum_{i=1}^{n} & Z_i^a - W_i^a x_i \\
\text{s.t. } & Z_i \leq X_i^a \quad \forall i \in \{1, \ldots, n\} \\
& Z_i \leq W_i^a \quad \forall i \in \{1, \ldots, n\} \\
& \mathbf{w}^a \in W \\
& \mathbf{x}^a \in \mathbf{X} \\
& X_i^a \in \{0, 1\}; 0 \leq Z_i \leq 1
\end{aligned}$$

References

R.L. Keeney, H. Raiffa. "Decisions with Multiple Objectives - Preferences and Value Tradeoffs", Cambridge University Press, 1993

E. Jacquet-Lagrèze, Y. Siskos. *"Assessing a set of additive utility functions for multicriteria decision making: the UTA method"*, European Journal of Operational Research, Vol.10, pp. 151-164, 1982.

U. Chajewska, D. Koller, R. Parr. *"Making Rational Decisions using Adaptive Utility Elicitation"*, Proceedings de AAAI-00, pp. 363-369, 2000

C. Boutilier, R. Patrascu, P. Poupart, D. Schuurmans (2006) *"Constraint-based Optimization and Utility Elicitation using the Minimax Decision Criterion"*, Artificial Intelligence, Vol. 170, 8-9, 2006

D. Braziunas, C. Boutilier. *"Minimax Regret-based Elicitation of Generalized Additive Utilities"*, Proceedings of UAI-07, pp. 25-32, 2007

P. Viappiani, C. Boutilier. *"Regret-based optimal recommendation sets in conversational recommender systems"*. Proc. of RecSys-2009: 101-108, 2009.

P. Viappiani, C. Boutilier. *"Optimal Bayesian Recommendation Sets and Myopically Optimal Choice Query Sets"*. Proc. of NIPS-2010, pp. 2352-2360, 2010.