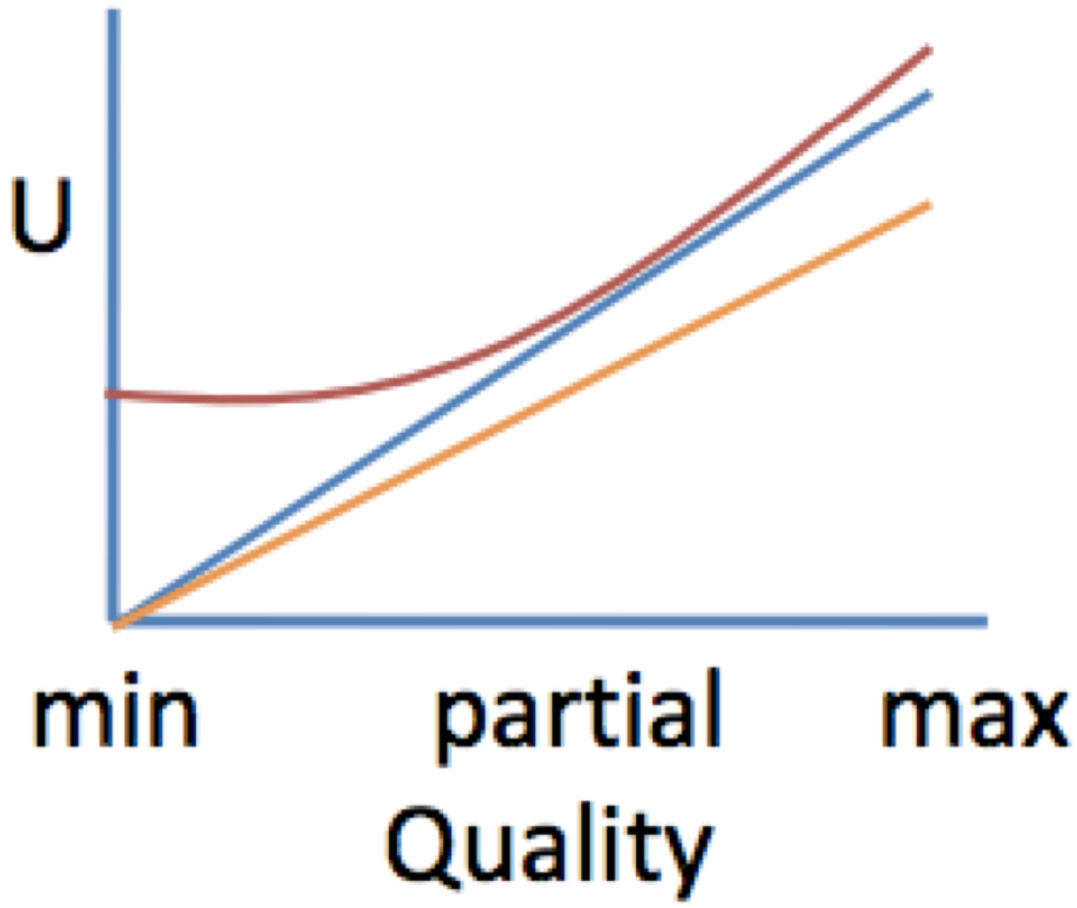
Learning Analytics

Dr. Bowen Hui

Computer Science

University of British Columbia Okanagan

Today: Eliciting a Utility Function



Previous Lectures

- Probabilistic inference (Bayes nets)
  - Maintain beliefs about the world
  - Probabilities can be estimated

- Utility theory
  - Quantify preferences
  - Where do utilities come from?

- Combination:
  - Take actions with maximum expected utility

Preference Over Lotteries

- Recall: action/choice outcomes can be stochastic
- Each action is a “lottery” over outcomes

- A simple lottery over X has the form:

$$l = [(p_1,x_1), (p_2,x_2), \ldots, (p_n,x_n)]$$

where $p_i \geq 0$ and $\sum p_i = 1.0$

- Outcomes are just trivial lotteries
  - One outcome has probability 1.0

Preference Over Lotteries

- Recall: action/choice outcomes can be stochastic
- Each action is a “lottery” over outcomes

- A simple lottery over X has the form:

  $$l = [(p_1,x_1), (p_2,x_2), \ldots, (p_n,x_n)]$$

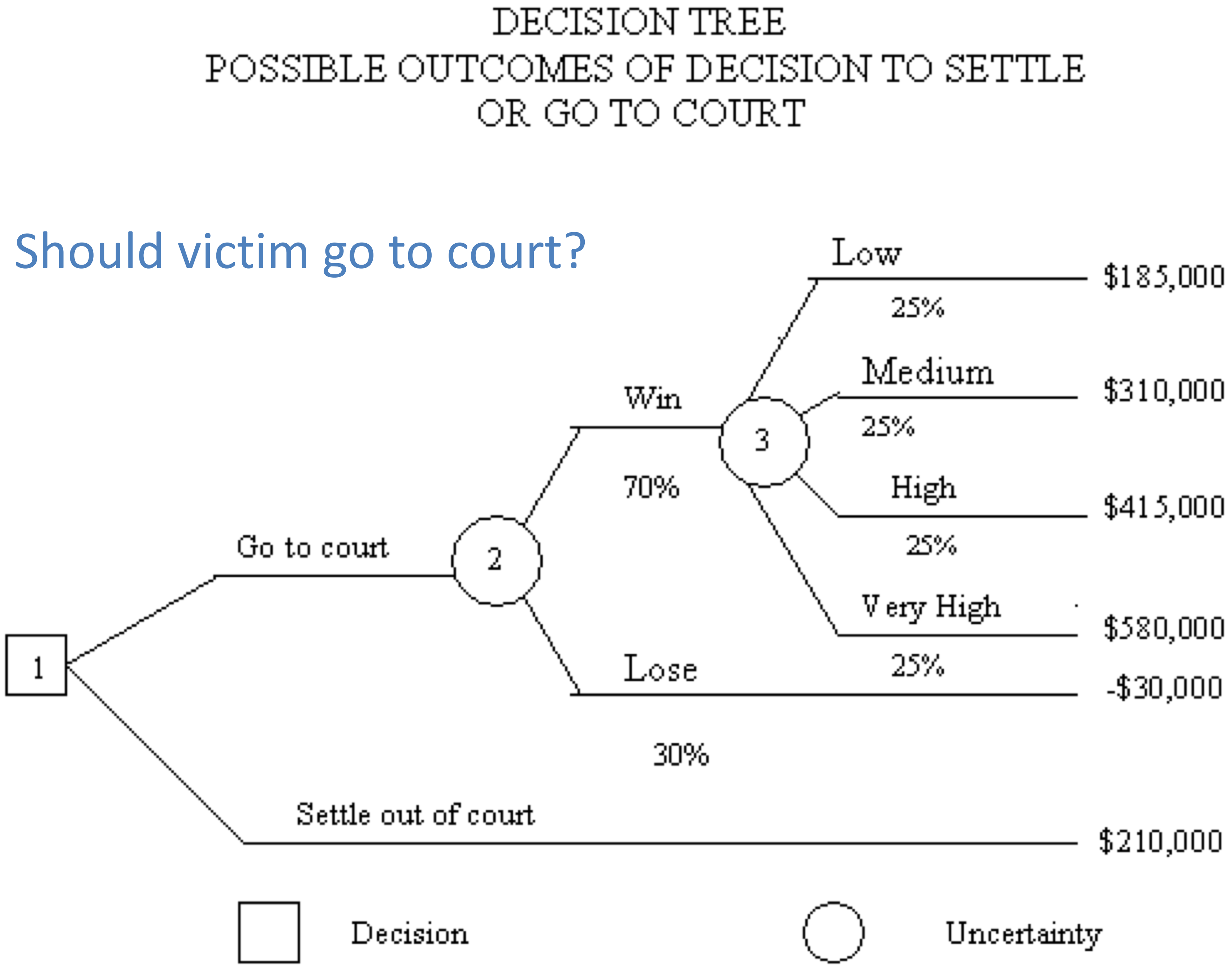
  where $p_i \geq 0$ and $\sum p_i = 1.0$

- A compound lottery allows outcomes to be lotteries:

  $$[(p_1,l_1), (p_2,l_2), \ldots, (p_n,l_n)]$$

  restrict to finite compounding

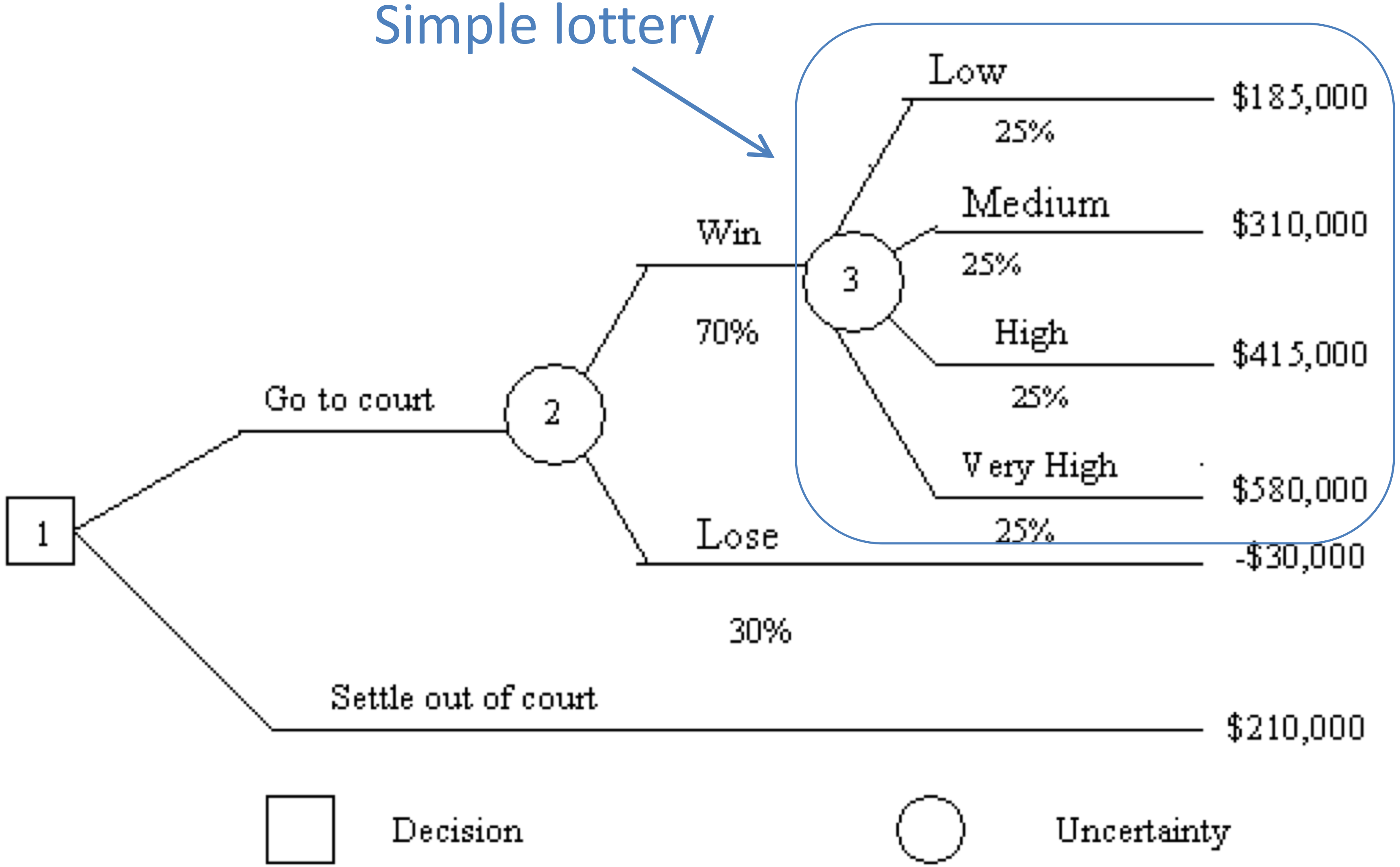
Should victim go to court?



DECISION TREE
POSSIBLE OUTCOMES OF DECISION TO SETTLE
OR GO TO COURT

Image taken from www.mediate.com

DECISION TREE
POSSIBLE OUTCOMES OF DECISION TO SETTLE OR GO TO COURT



Simple lottery

Image taken from www.mediate.com

DECISION TREE
POSSIBLE OUTCOMES OF DECISION TO SETTLE OR GO TO COURT

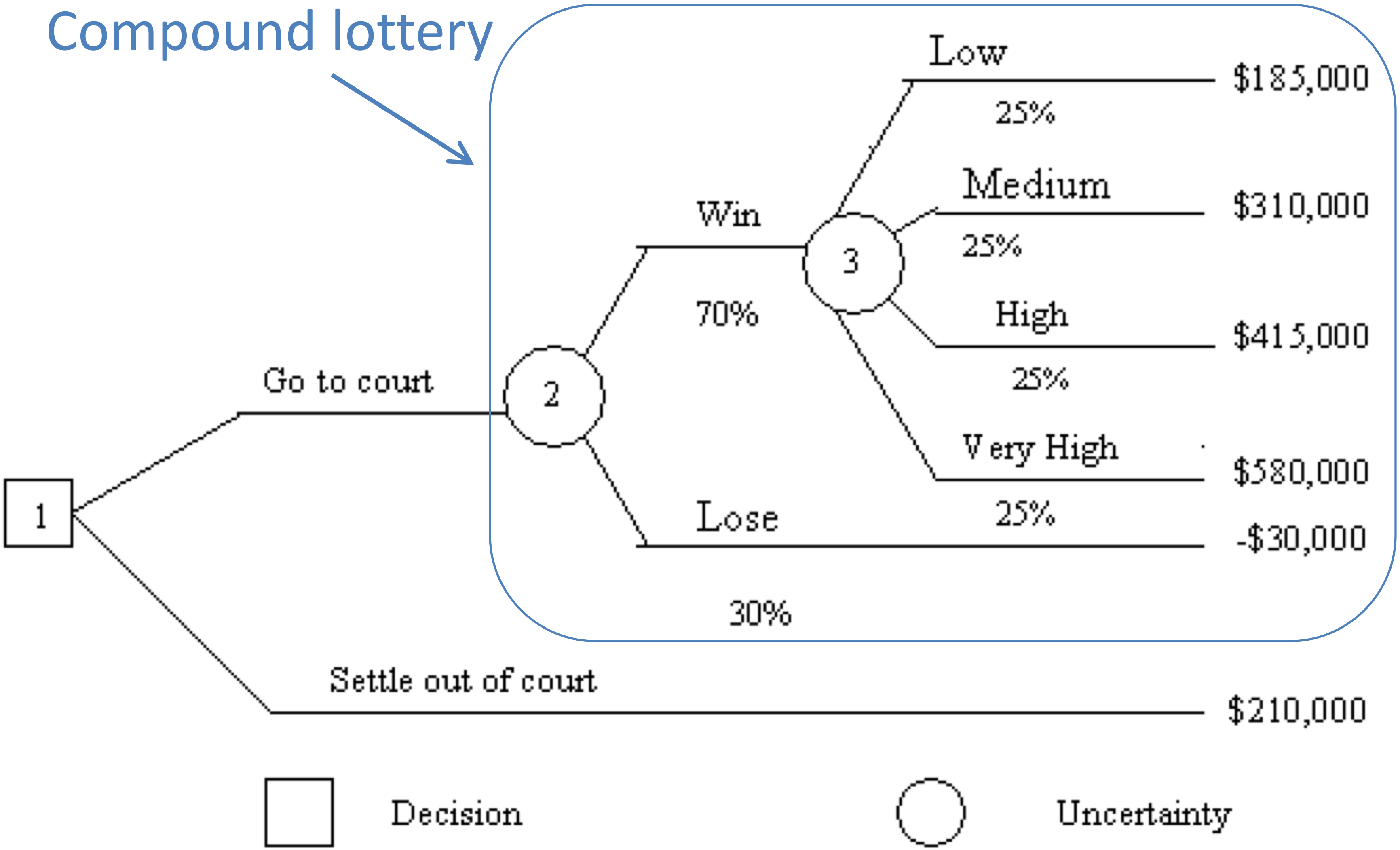

Compound lottery

Image taken from www.mediate.com

Which Would You Choose?

- Between:
  - [(0.5, $200,000), (0.5 $0)]　　　Gamble

Which Would You Choose?

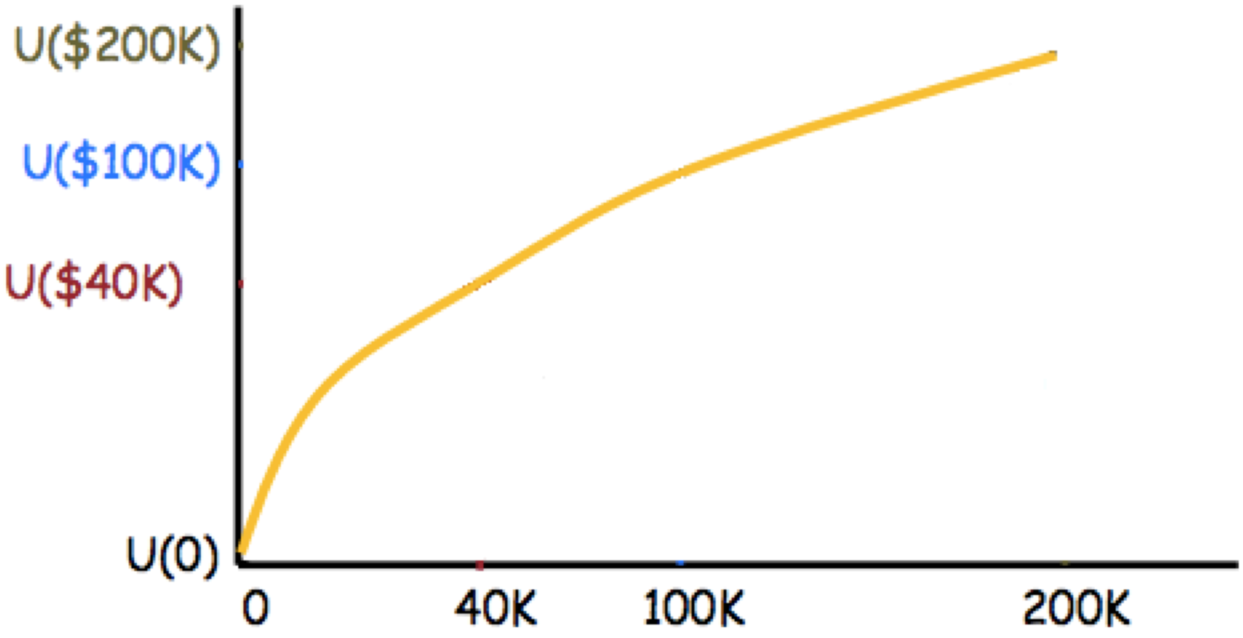
- Between:
  - [(0.5, \$200,000), (0.5 \$0)] Gamble
  - \$100,000 Sure Win

Which Would You Choose?

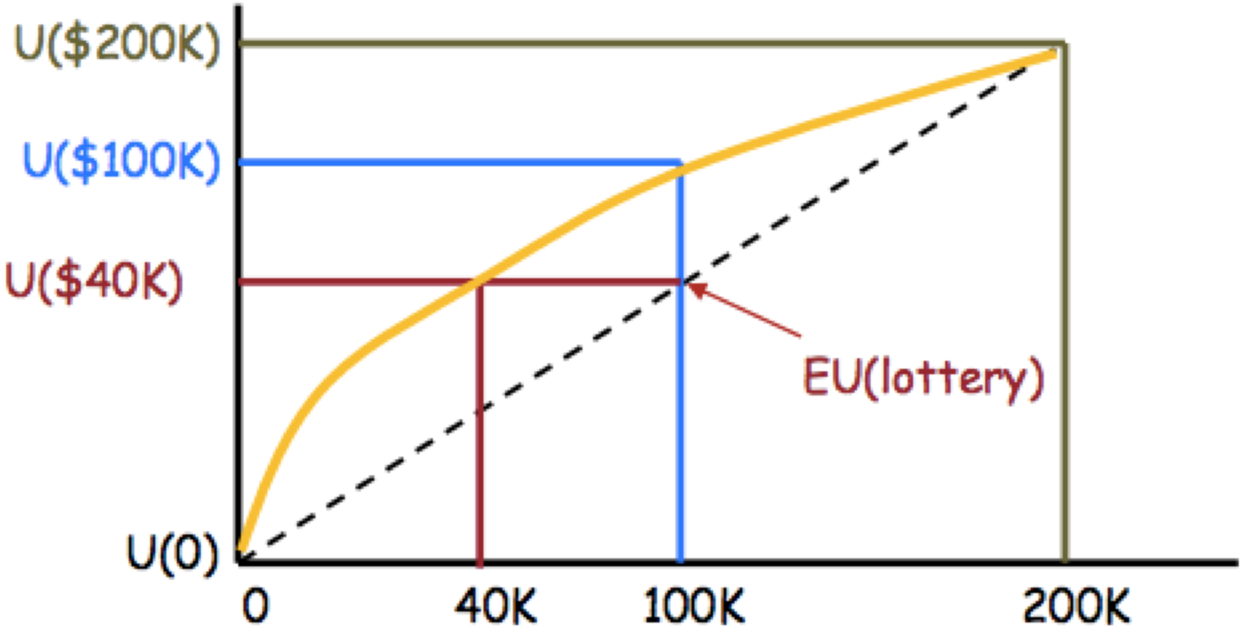
- Between:
  - [(0.5, \$200,000), (0.5 \$0)]
  - \$100,000

- Utility of money is nonlinear!
  - U(\$100K) > 0.5*U(\$200K) + 0.5*U(\$0)
  - What if lottery changed to: \$500K, \$1M,… p = 0.6, 0.7, 0.999, …?

Concave Utility Function



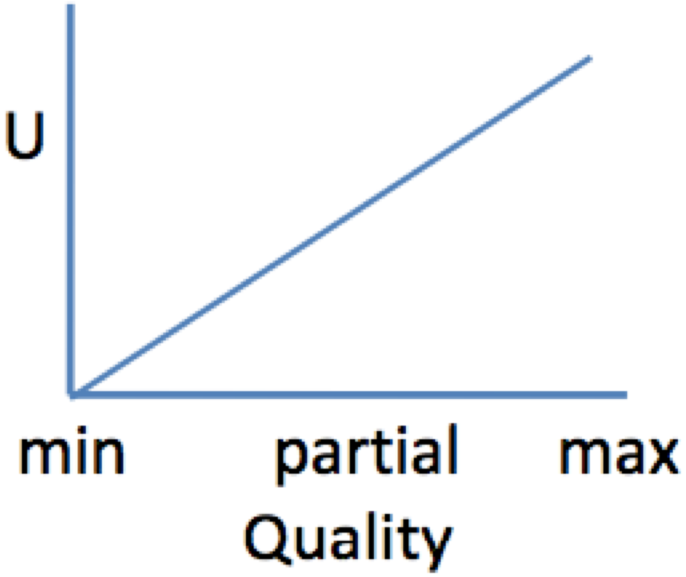
Concave Utility Function



Most people take a sure win of $40K over a gamble for $200K

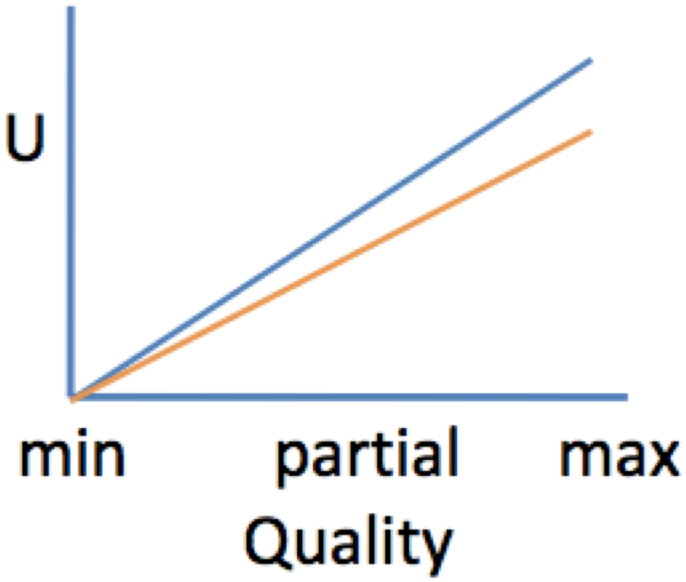
Abstract Utility Functions for Intelligent Tutoring Systems

- How valuable is the help you get as a function of …?



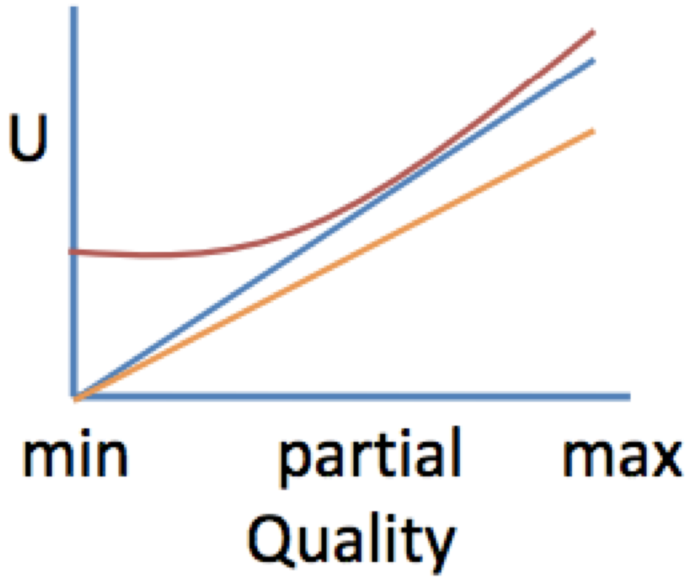
Abstract Utility Functions for Intelligent Tutoring Systems

- How valuable is the help you get as a function of …?



Abstract Utility Functions for Intelligent Tutoring Systems

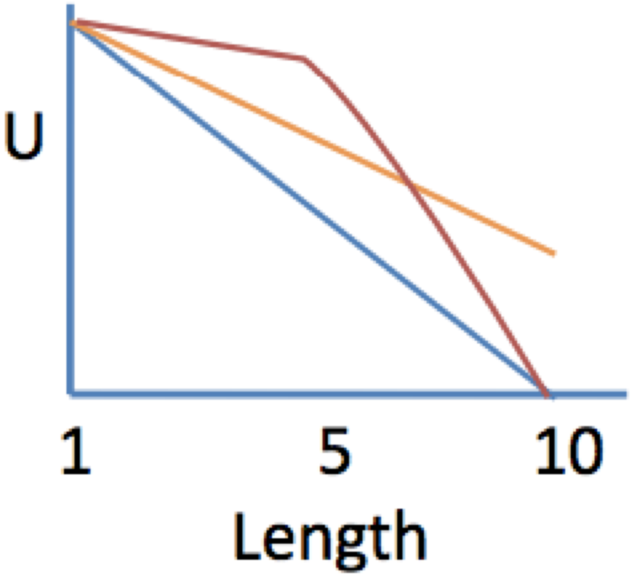
- How valuable is the help you get as a function of …?



Each function may represent different individual users

Abstract Utility Functions for Intelligent Tutoring Systems

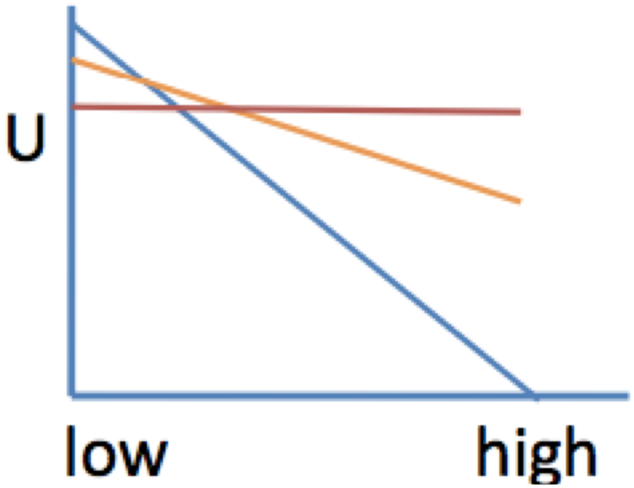
- How valuable is the help you get as a function of …?



Recall from HCI: interface bloat is bad design

Abstract Utility Functions for Intelligent Tutoring Systems

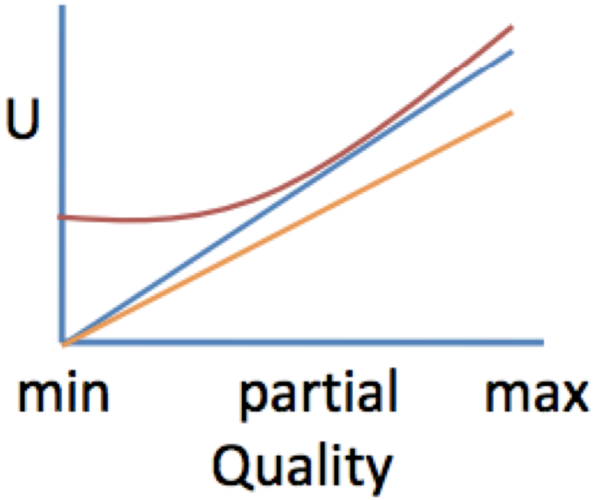
- How valuable is the help you get as a function of …?

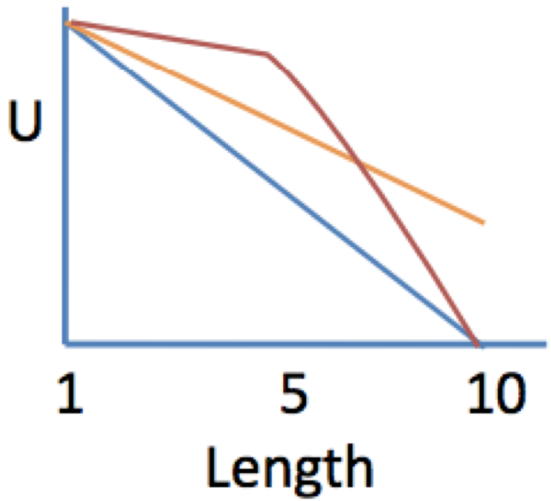


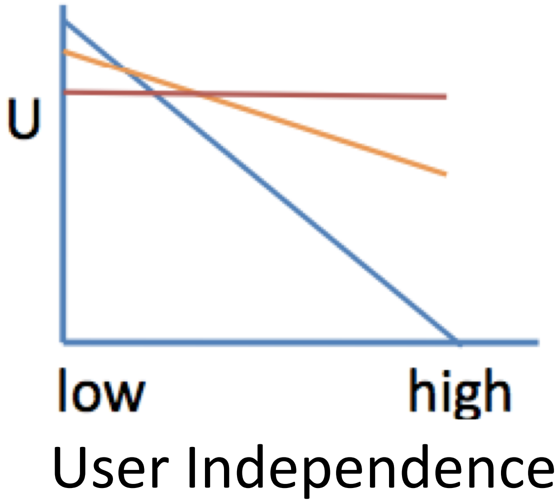
User Independence

Abstract Utility Functions for Intelligent Tutoring Systems

- How valuable is the help you get as a function of …?







- Additive decomposition?
  - – Roughly: U(help) = c_1*quality + c_2*length + c_3*indep

Elicitation Queries

- Decision problem setup:
  - Best outcome with utility 1.0
  - Worst outcome with utility 0.0
  - Other outcomes have utility values in [0,1]

Elicitation Queries

- Decision problem setup:
  - Best outcome with utility 1.0
  - Worst outcome with utility 0.0
  - Other outcomes have utility values in [0,1]

- Standard gamble
  - SG(pr) = [pr,o^{best}; 1-pr, o_{worst}]
  - Expected utility of SG = pr

Which would you choose:

- [(0.5, $200,000), (0.5 $0)]
- $100,000

Elicitation Queries

- Decision problem setup:
  - Best outcome with utility 1.0
  - Worst outcome with utility 0.0
  - Other outcomes have utility values in [0,1]

- Standard gamble
  - $SG(pr) = [pr, o^{best}; 1\text{-}pr, o_{worst}]$
  - Expected utility of SG = pr

Which would you choose:

- [(0.5, \$200,000), (0.5 \$0)]
- \$100,000

- Example:
  - Let pr = 0.5, then EU(SG) = 0.5(1) + 0.5(0) = 0.5
  - Let pr = 0.8, then EU(SG) = 0.8(1) + 0.2(0) = 0.8

What This Means

- Given an outcome
  - Call this o_i
- You want to know its utility value in [0,1]
  - Call this pr
- Can’t directly ask for pr
- Set it up a standard gamble with two options:
  - $SG(pr) = [pr, o^{best}; 1\text{-}pr, o_{worst}]$
  - Sure win of o_i
- Find the value of pr such that the person is indifferent between the two options
  - Then pr is the expected value of o_i

Standard Gamble Query

- Standard gamble
  - – SG(pr) = [pr,o^{best}; 1-pr, o_{worst}]
  - – Expected utility of SG = pr

- Standard gamble query
  - – Alternative A: SG(pr)
  - – Alternative B: o_i
  - – Question to ask: What is *pr* s.t. *SG(pr) = o_i*?
  - – Response to elicit from user: pr

Try It Yourself

- Suppose Quiz 2:
  - Choose between coin flip and a fixed score

- Find a partner and ask:
  - Alternative A: SG(pr) = [pr, 100%; 1-pr, 0%]
  - Alternative B: o_i = 51%
  - Question to ask: What is *pr* s.t. $SG(pr) = o_i$?
  - Response to elicit from user: pr

- Switch roles and replace:
  - Alternative B: o_i = 88%

How well did it go?

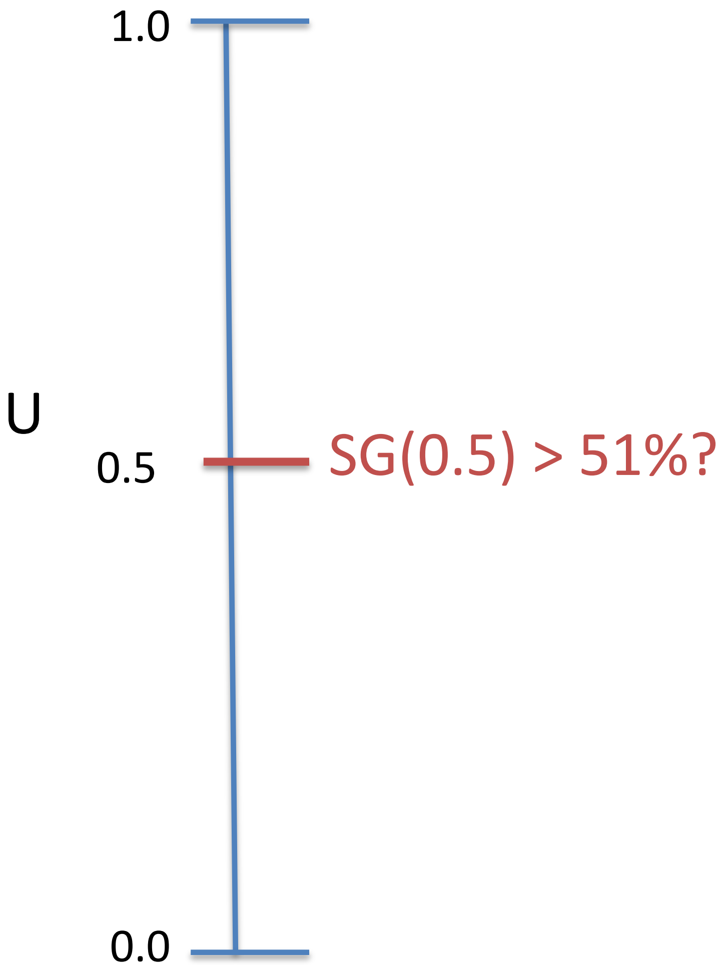
Bound Query

- Bound query
  - Alternative A: SG(pr)
  - Alternative B: o_i
  - Question to ask: Given *pr*, is *SG(pr)* > o_i?
  - Response to elicit from user: Yes/No

Like binary search

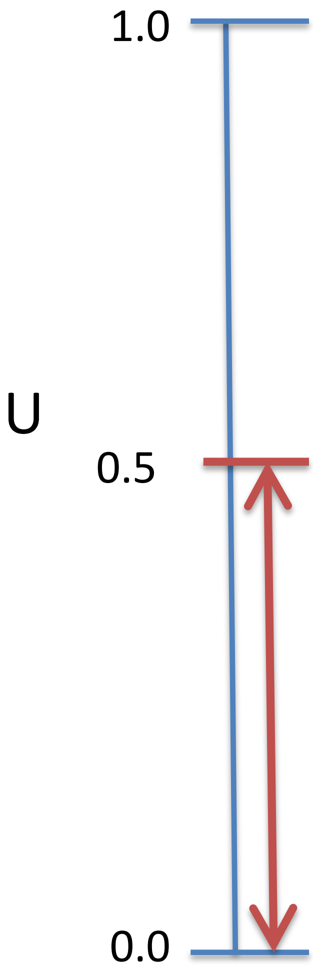
- Less information gain
  - Must ask sequence of queries to identify *pr* of o_i

[pr,100%;1-pr,0%] > 51% ?



SG(0.5) > 51%?

[pr,100%;1-pr,0%] > 51% ?



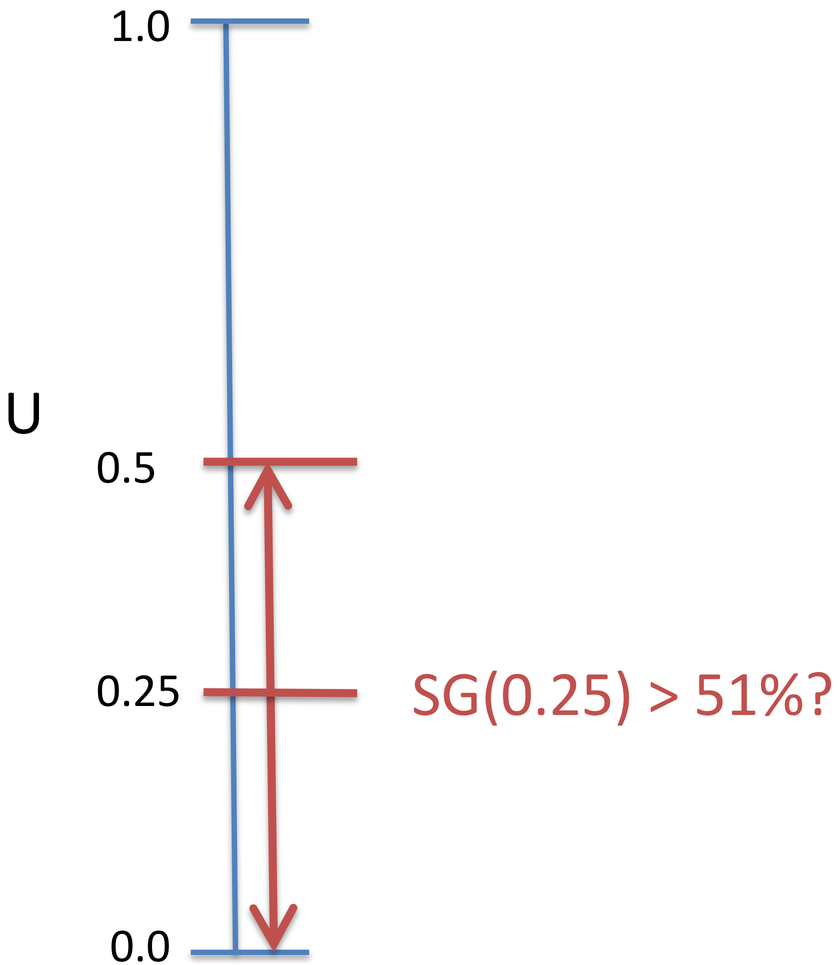
SG(0.75) > 51%?

Must still be “yes”

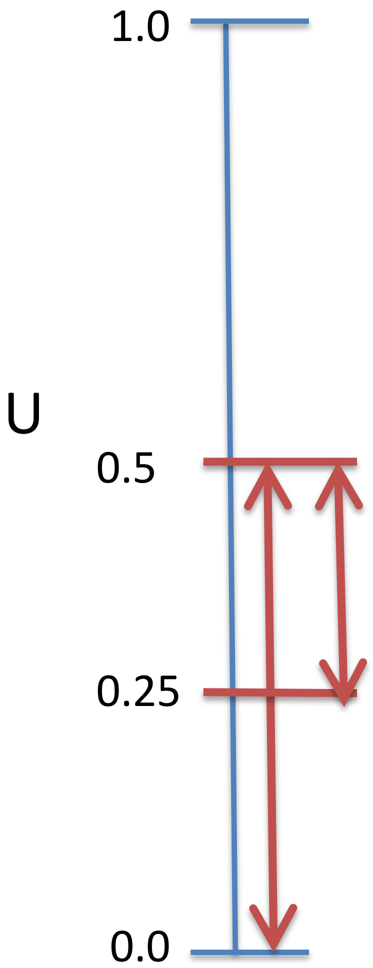
SG(0.5) > 51%? **yes**

- Search lower region

[pr,100%;1-pr,0%] > 51% ?



[pr,100%;1-pr,0%] > 51% ?

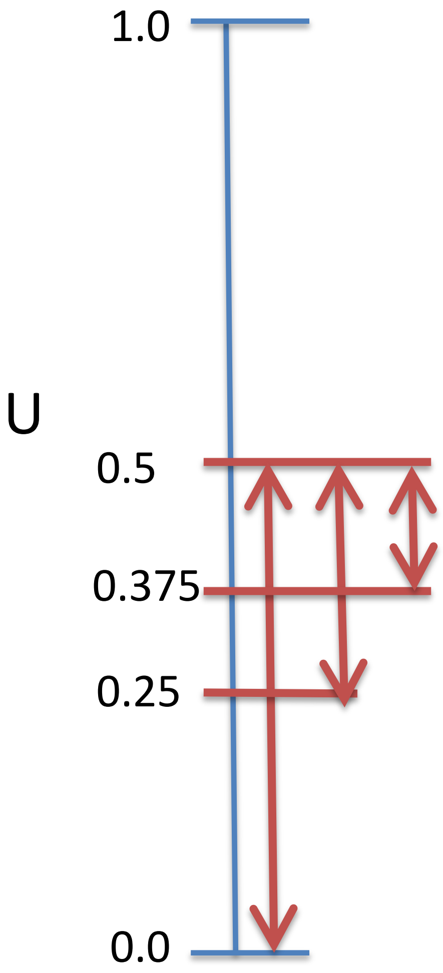


SG(0.25) > 51%? **no**

- Search upper region

SG(0.10) > 51%?
Must still be “no”

[pr,100%;1-pr,0%] > 51% ?



SG(0.375) > 51%? **no**

- Search upper region

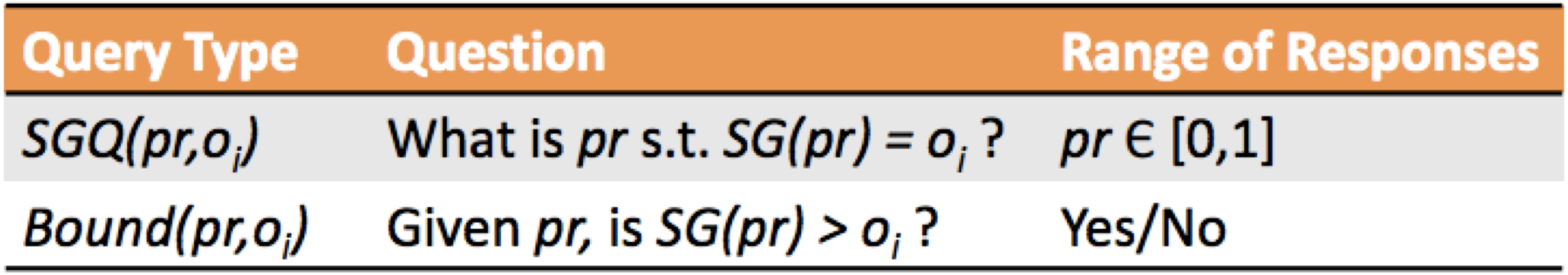
Continue until the response is "the two options are the same"

Try The New Version

- Same context (Quiz 2 score)
  - Construct query (choose reasonable *pr*)
  - Keep asking queries until you've identified *pr*

- Find a partner and ask:
  - Alternative A: SG(pr) = [pr, 100%; 1-pr, 0%]
  - Alternative B: o_i = 75%
  - Question to ask: Given *pr*, is $SG(pr) > o_i$?
  - Response to elicit from user: Yes/No

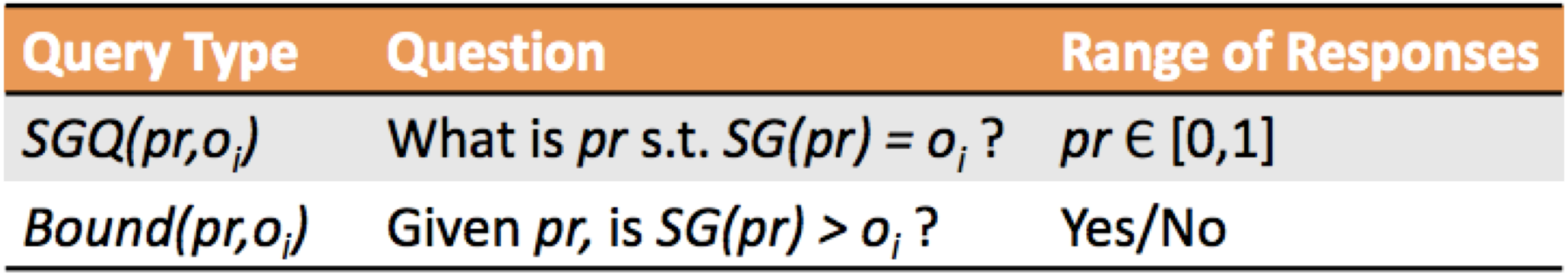
Comparison



Query Type	Question	Range of Responses
SGQ(pr,o_i)	What is *pr* s.t. *SG(pr)* = o_i ?	*pr* ∈ [0,1]
Bound(pr,o_i)	Given *pr*, is *SG(pr)* > o_i ?	Yes/No

- Which is better?

Comparison

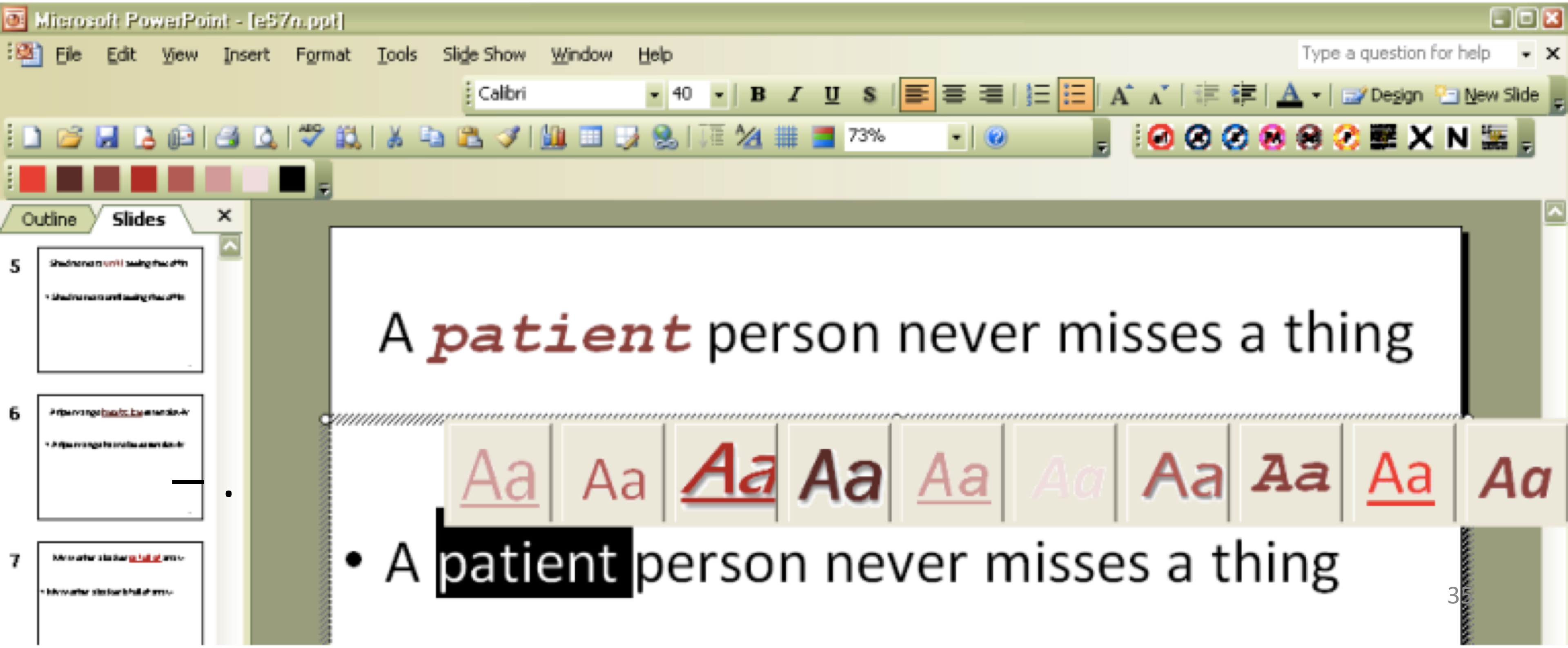


Query Type	Question	Range of Responses
SGQ(pr,o_i)	What is *pr* s.t. *SG(pr)* = o_i ?	*pr* ∈ [0,1]
Bound(pr,o_i)	Given *pr*, is *SG(pr)* > o_i ?	Yes/No

- Which is better?
  - – SGQ provides pinpoint answer, but mixture of outcomes is difficult to interpret
  - – Bound query offers more meaningful queries, but less meaningful as feasible regions get smaller

Case Study in Software Help (Hui & Boutilier 2008)

- Help: Macro suggestion in PowerPoint toolbar
- Variables: Length, Quality, User Independence
  - User independence proxy via task difficulty



Microsoft PowerPoint - [e57n.ppt]
File Edit View Insert Format Tools Slide Show Window Help
Type a question for help
Calibri 40
Design New Slide
73%
Outline Slides
5
6
7
Aa

A *patient* person never misses a thing

– .

- A patient person never misses a thing

Experiment Setup

- Series of controlled highlighting task in PowerPoint

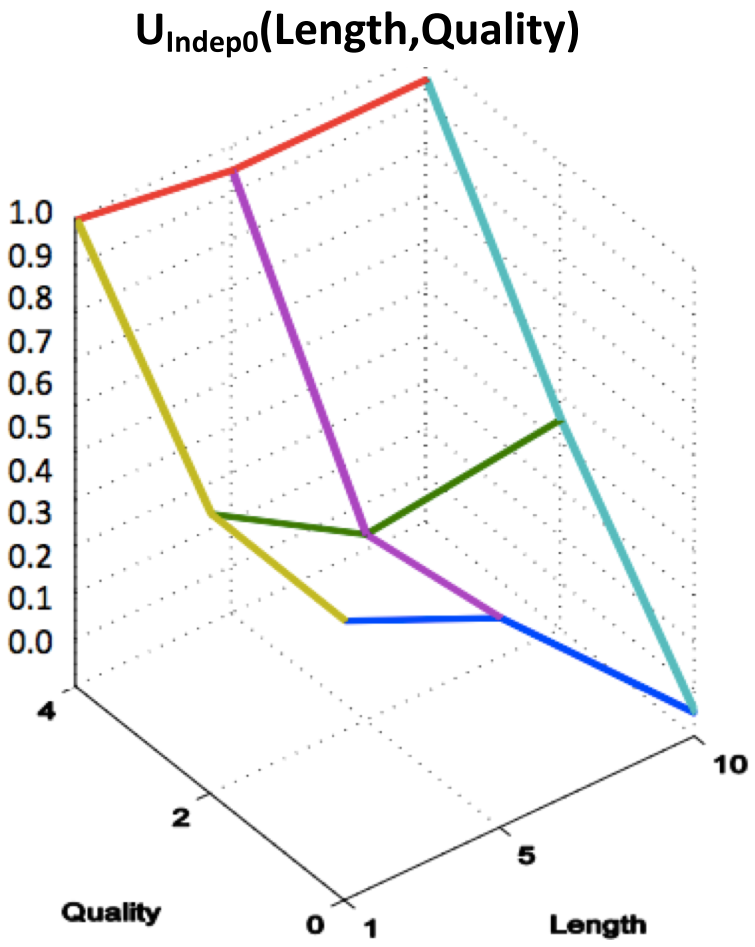
- Goal: To sample *U(I,L,Q)*
  - Indep, I = 0 (low), 1 (high)
  - Length, L = 1, 5, 10 icons
  - Quality, Q = 0 (wrong), 2 (partial), 4 (perfect)

- o^{best} = I0,L1,Q4
- o_{worst} = I1,L10,Q0

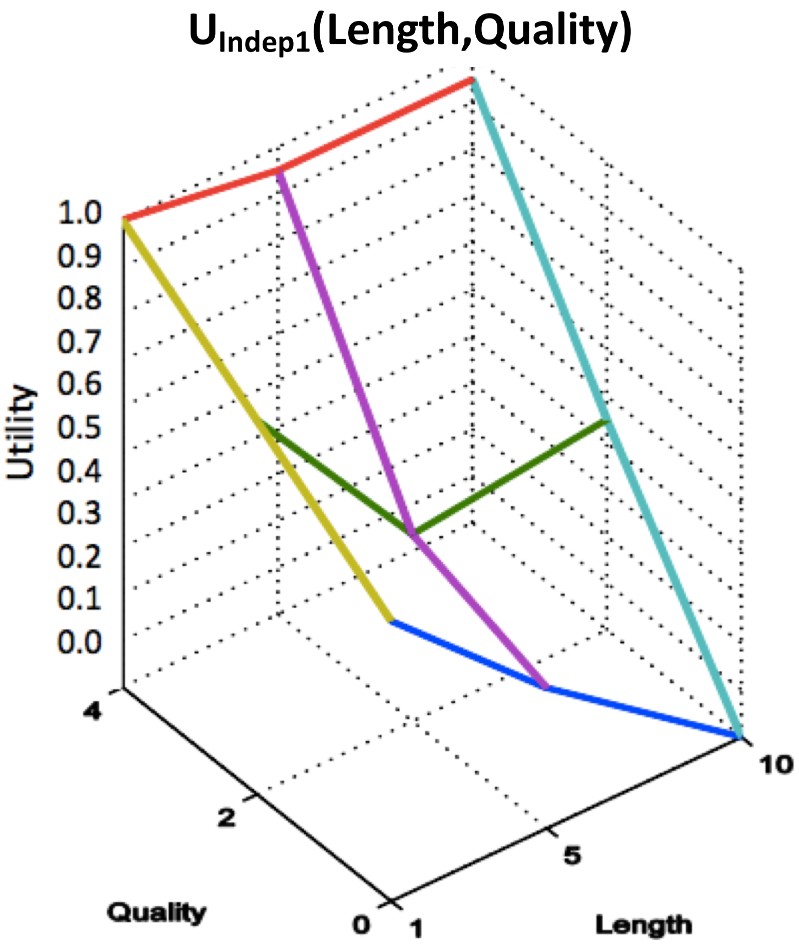
- Elicited until small “feasible” regions (pr ± 0.05)

User 11 (midpoints)

When: Task is difficult



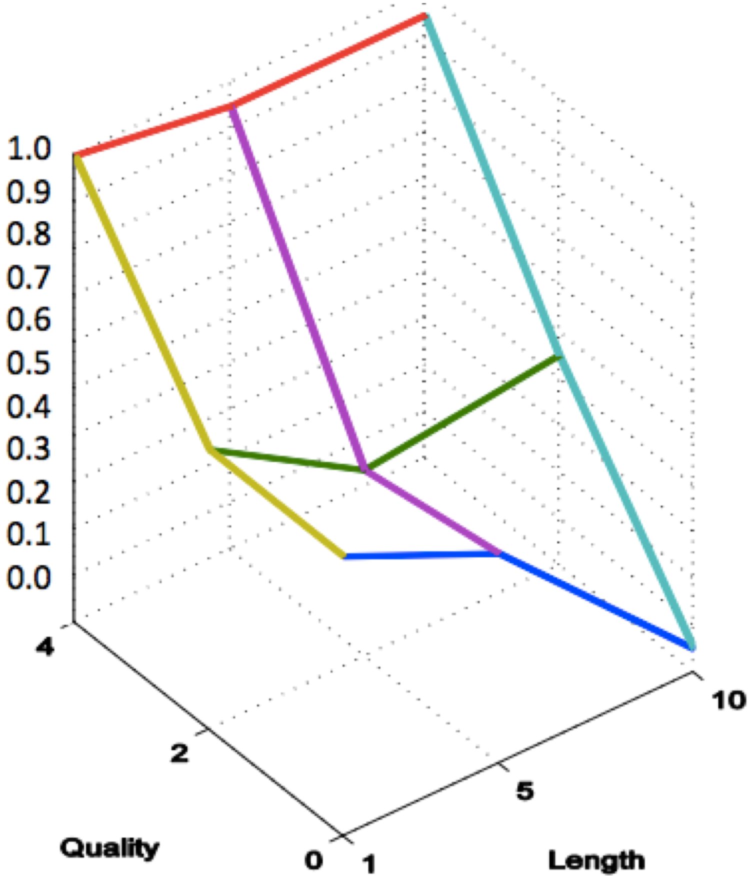
When: Task is easy



User 11 – role of length

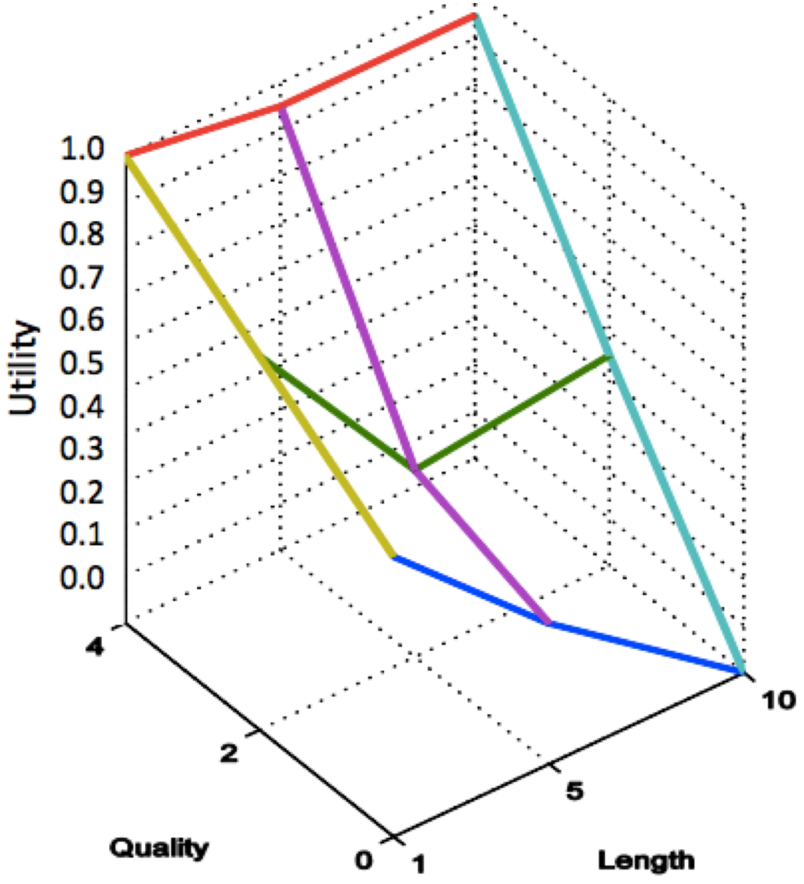
When: Task is difficult

U_{Indep0}(Length,Quality)



When: Task is easy

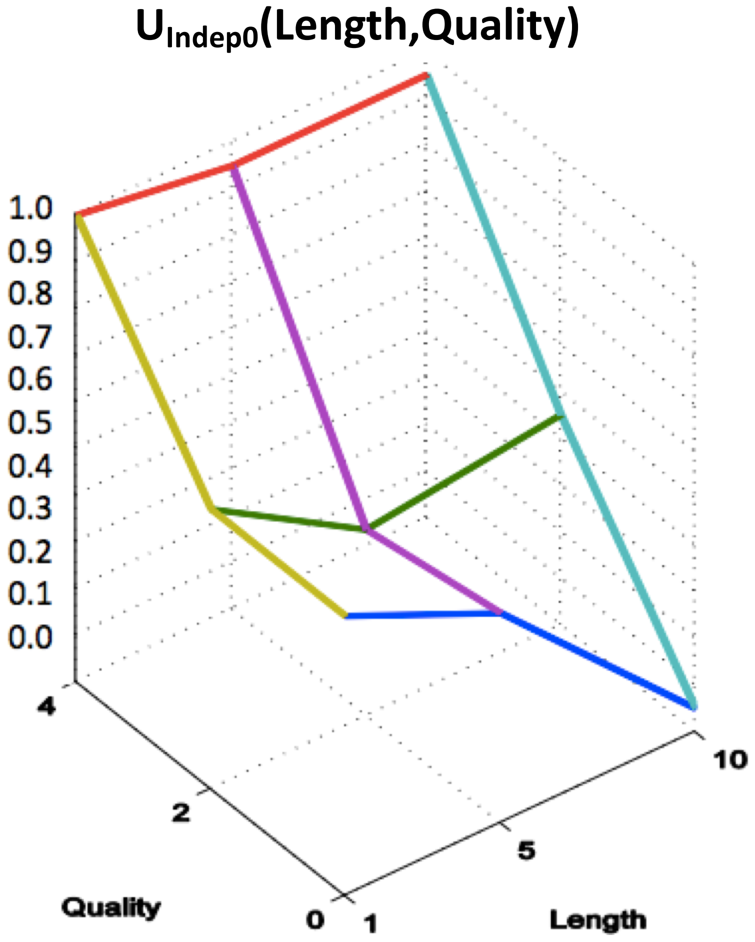
U_{Indep1}(Length,Quality)



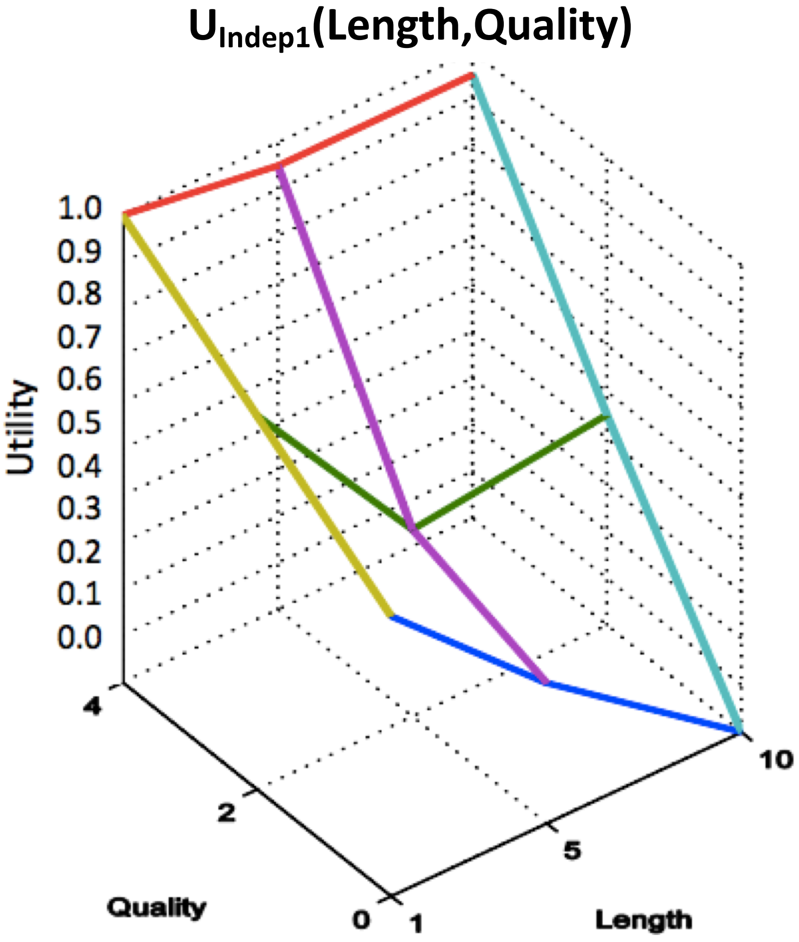
- High tolerance for length when Qual is also high

User 11 – value of non-perfect help

When: Task is difficult



When: Task is easy

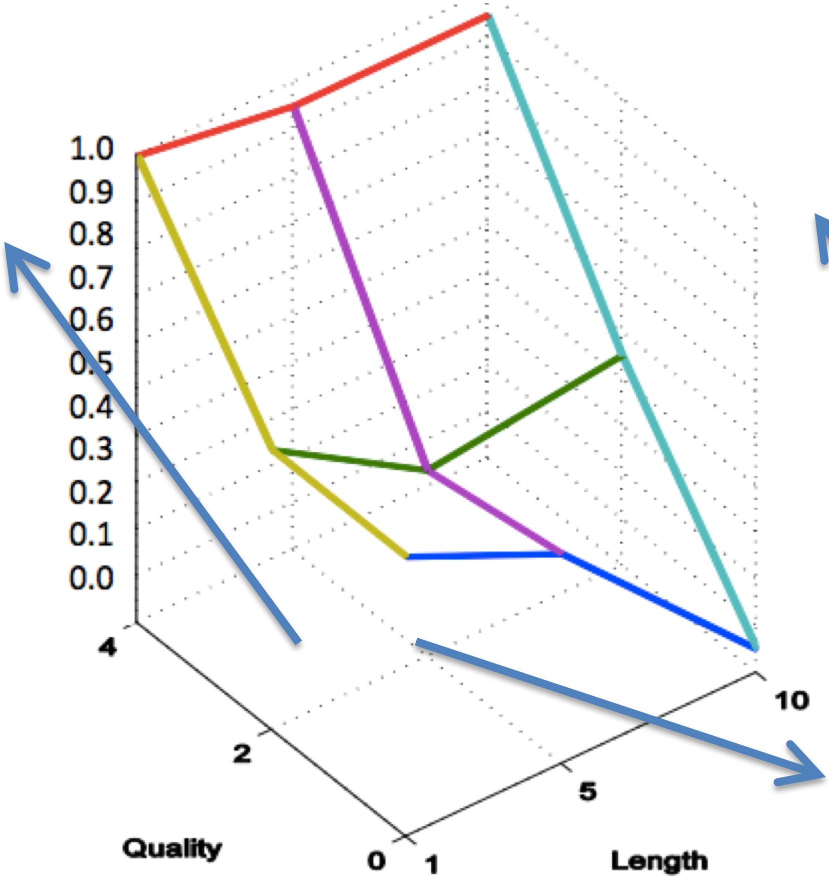


- Utility in Q2, even Q0
- Higher utility with easy task

User 11 – monotonicity

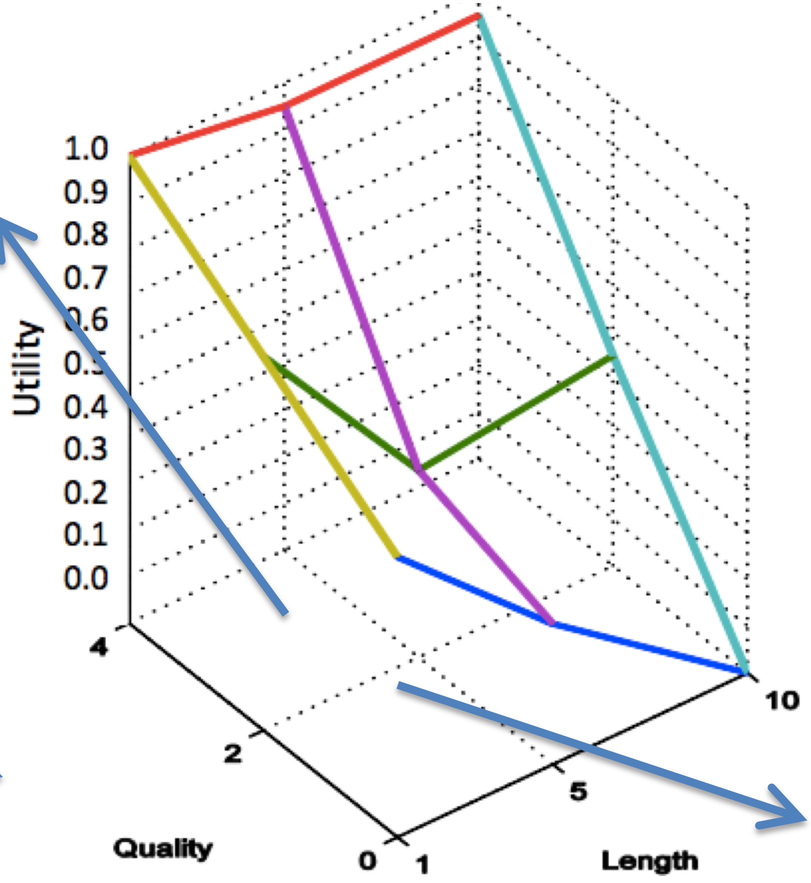
When: Task is difficult

U_{Indep0}(Length,Quality)



When: Task is easy

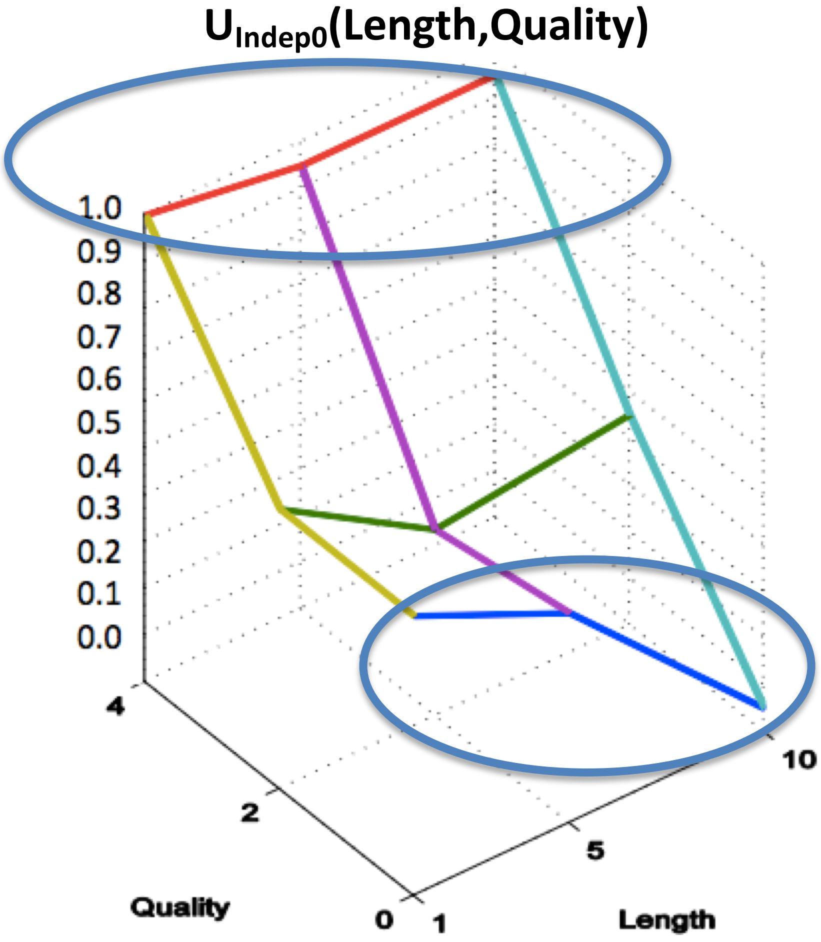
U_{Indep1}(Length,Quality)



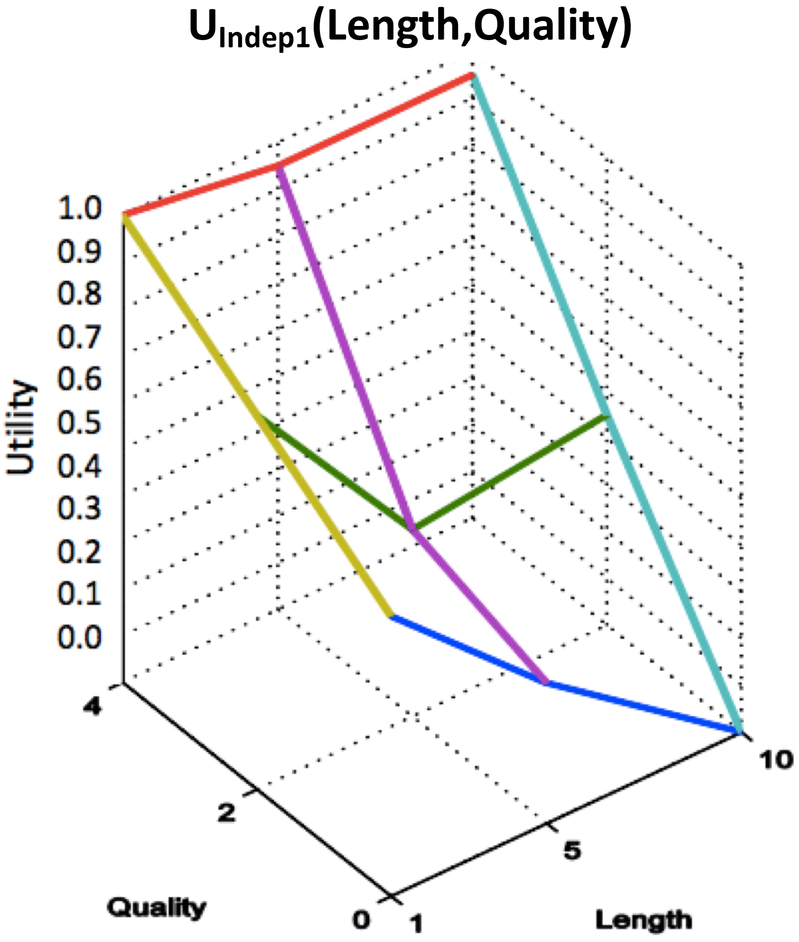
- Non-decreasing in Q and non-increasing in L

User 11 – curvature of partial function

When: Task is difficult



When: Task is easy



- Different structure as Q changes

Rest of the Study

- Imagine carrying out 50 highlighting tasks vs. doing 50 highlighting tasks

- Conceptual vs. experiential preferences
  - What you think you like isn’t what you actually like

- Adapting elicitation procedure to elicit experiential preferences

Key Ideas

- Main concepts
  - Eliciting a utility function via incremental constraints
  - Standard gamble query: What is *pr* s.t. $SG(pr) = o_i$?
  - Bound query: Given *pr,* is $SG(pr) > o_i$?
- Representation:
  - For simplicity, utility function may assume additive decomposition
- Algorithm:
  - Bound query for preference elicitation induces constraints for incremental refinement