Representing, Eliciting, and Reasoning with Preferences

AAAI-07 Tutorial Forum

Ronen Brafman

Ben-Gurion University (Israel)

Carmel Domshlak

Technion (Israel)

Outline

1. Introduction:
   1. Why preferences?
   2. The Meta-Model: Models, Languages, Algorithms

2. Preference Models, Languages, and Algorithms
   1. Total orders and Value Functions
   2. Partial orders and Qualitative Languages
   3. Preference Compilation
   4. Gambles and Utility functions

3. From Preference Specification to Preference Elicitation

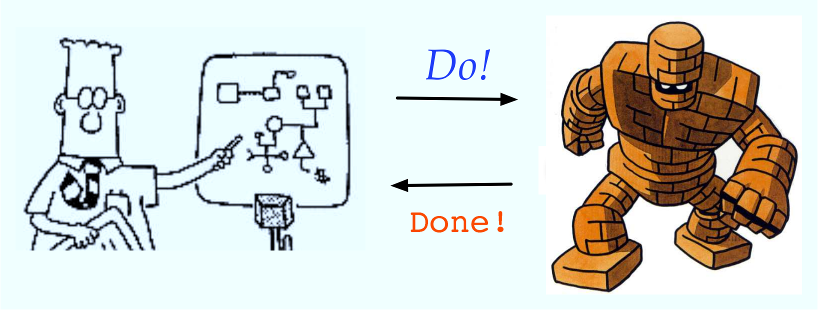
Outline

1. Introduction:
   1. Why preferences?
   2. The Meta-Model: Models, Languages, Algorithms

2. Preference Models, Languages, and Algorithms
   1. Total orders and Value Functions
   2. Partial orders and Qualitative Languages
   3. Preference Compilation
   4. Gambles and Utility functions

3. From Preference Specification to Preference Elicitation

Autonomous Agent Acts on Behalf of a User



Do!

Done!

When Would We Need *Communicating* Our Preferences?

What's wrong with simple goals?

Goals are **rigid**— "do or die"

The world can be highly uncertain

We can't tell ahead of time if our ultimate goal is achievable

When Would We Need *Communicating* Our Preferences?

Our application realizes that the goal is unachievable

What should we do?

Sometimes we give up ...

- Example: Solving a puzzle
- Example: DARPA Grand Challenge (not very convincing)

Most times we don't!

- Can't get the isle seat on KLM's morning flight to Vancouver
- Conclusion(?):
  I'll stay at home. You can read the tutorial online

When Would We Need *Communicating* Our Preferences?

Our application realizes that the goal is unachievable

What should we do?

We go for the **second best** alternative

- What is “second best”?
- What if ”second best” is infeasible?

Preference Specification

How complicated can/should it be?

Easy – if you find an easy way to rank alternatives

Single objective with natural order

- Optimize cost, optimize quality
- Optimize both? ...

Very small set of alternatives

- Hyatt $\succ$ Best-Western $\succ$ Student Housing $\succ$ A bench in Stanly Park

Preference Specification

But ...

Task: *Find the best (for me) used car advertised on the web!*

1. **large space of alternative outcomes**
   - lots of different used cars advertised online for sale
   - I don't want to explicitly view or compare all of them
2. (possibly involved) **multi-criteria objective**
   - my choice would be guided by color, age, model, milage, ...
3. (again) **uncertainty** about which outcomes are feasibile
   - Is there a low-milage Ferrari for under $5000 out there?

Preference Specification

But ...

Task: *Find the best (for me) used car advertised on the web!*

1. **large space of alternative outcomes**
2. (possibly involved) **multi-criteria objective**
3. (again) **uncertainty** about which outcomes are feasibile

And in face of this, we still need to

1. realize the preference order *to ourselves*
   - Easy? Try choosing one of some 20+ used cars on sale
2. communicate this order to an agent working for us
   - Annoying even for small sets of outcomes
     (e.g., 20+ alternative car configurations)
   - What if the space of alternative outcomes is
     (combinatorially) huge?

Bottom Line

We hope all the above have convinced you that ...

To “do the right thing” for the user, the agent must be provided with a *specification* of the user’s preference ordering over outcomes.

Questions of Interest

- How can we minimize the cognitive effort and time required to attain information about the user's preferences?

- How can we efficiently represent and reason with such information?

Outline

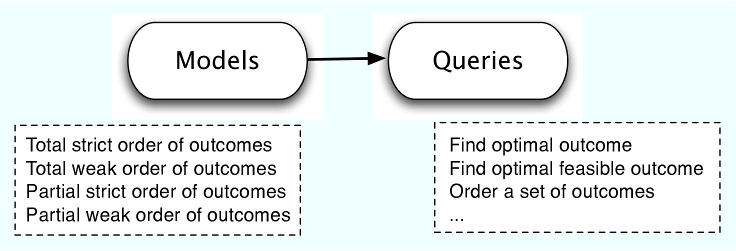
1. Introduction:
   1. Why preferences?
   2. The Meta-Model: Models, Languages, Algorithms

2. Preference Models, Languages, and Algorithms
   1. Total orders and Value Functions
   2. Partial orders and Qualitative Languages
   3. Preference Compilation
   4. Gambles and Utility functions

3. From Preference Specification to Preference Elicitation

The Meta-Model

Models and Queries



Models

Queries

Total strict order of outcomes
Total weak order of outcomes
Partial strict order of outcomes
Partial weak order of outcomes

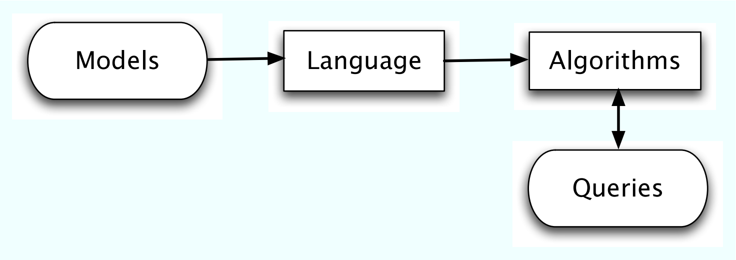
Find optimal outcome
Find optimal feasible outcome
Order a set of outcomes
...

Framework

- **models** for **defining**, classifying, and understanding the *paradigm of preferences*
- **queries** to capture questions of interest about the models
  - what queries are of interest depends on the task in hand

The Meta-Model

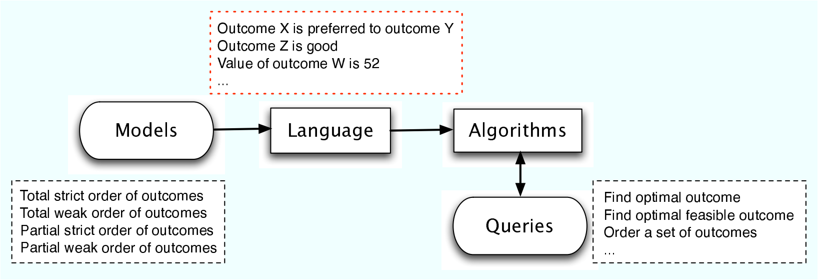
Languages + Algorithms



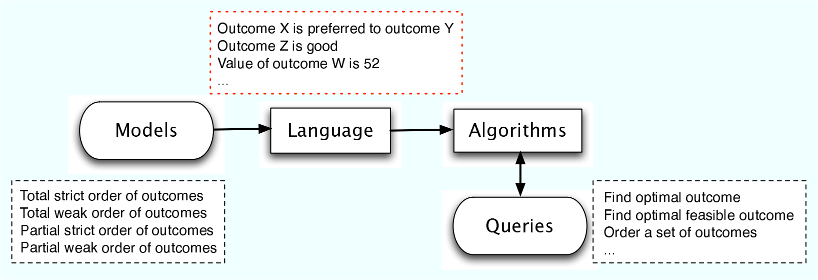
Framework

- **models** for **defining**, classifying, and understanding preferences
- **languages** for **communicating** and **representing** the models
- **algorithms** for **reasoning** (answering **queries**) about the models

Preferences: Languages


Outcome X is preferred to outcome Y
Outcome Z is good
Value of outcome W is 52
...
Models
Language
Algorithms
Queries
Total strict order of outcomes
Total weak order of outcomes
Partial strict order of outcomes
Partial weak order of outcomes
Find optimal outcome
Find optimal feasible outcome
Order a set of outcomes
...

Preferences: Languages


Outcome X is preferred to outcome Y
Outcome Z is good
Value of outcome W is 52
...
Models
Language
Algorithms
Queries
Total strict order of outcomes
Total weak order of outcomes
Partial strict order of outcomes
Partial weak order of outcomes
Find optimal outcome
Find optimal feasible outcome
Order a set of outcomes
...

The realm of real users

1. Incomplete and/or noisy model specification
2. System uncertain about the true semantics of the user's statements
3. Language constrained by system design decisions

Practical Shortcomings

Problem no. 1

Incomplete and/or noisy model specification

- Cognitive limitations
  - Users have great difficulty effectively elucidating their preference model even to themselves
- Typically, requires a time-intensive effort
- Example
  - Imagine having to compare various vacation packages
  - 4-star with a health club near the beach breakfast included in Cuba vs.
    5-star with four swimming pools in the center of Barcelona

We have an information elicitation problem

Practical Shortcomings

Problem no. 2

What does she mean when she says ...

- Natural language statements often ambiguous
  - ... and this is not a matter of syntax
- Not a problem when statements compare completely specified outcomes
- Problematic with *generalizing statements*
  - “I prefer going to a restaurant.”
  - “I prefer red cars to blue cars.”

We have an information decoding problem

Practical Shortcomings

Problem no. 3

Subjective language constraints

- Different users may have different **criteria** affecting their preferences over the same set of outcomes
  - Some camera buyers care about convenience (i.e., weight, size, durability, etc.)
  - Other care about picture quality (i.e., resolution, lens type and make, zoom, image stabilization, etc.)
- Any system comes with a **fixed alphabet** for the language
  - attributes of a catalog database
  - constants used by a knowledge base
  - ...

Practical Shortcomings

Problem no. 3

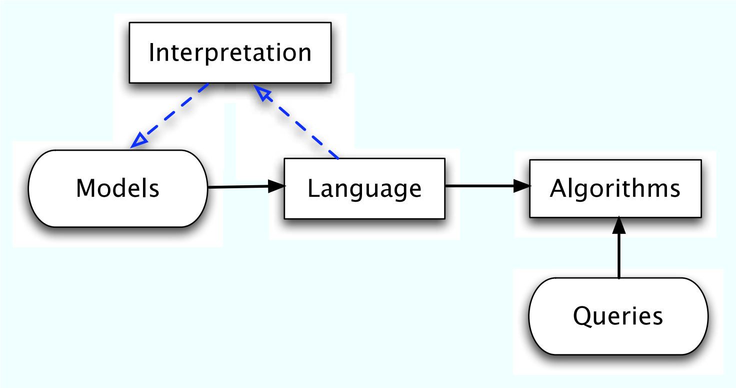
Subjective language constraints

- Different users may have different **criteria** affecting their preferences over the same set of outcomes
    - Some camera buyers care about convenience (i.e., weight, size, durability, etc.)
    - Other care about picture quality (i.e., resolution, lens type and make, zoom, image stabilization, etc.)
- Any system comes with a **fixed alphabet** for the language
    - attributes of a catalog database
    - constants used by a knowledge base
    - ...

♠ Hard to make preference specification (relatively) comfortable for all potential users

The information decoding problem gets even more complicated

Conclusion: Need for Language Interpretation



Interpretation

An interpretation maps the language into the model.
It provides *semantics* to the user's statements.

The Language
Intermediate summary

What would be an ”ultimate” language?

1. Based on information that’s
   - cognitively easy to reflect upon, and
   - has a common sense interpretation semantics
2. Compactly specifies natural orderings
3. Computationally efficient reasoning
   - complexity = F(language, query)

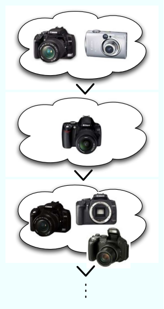
Outline

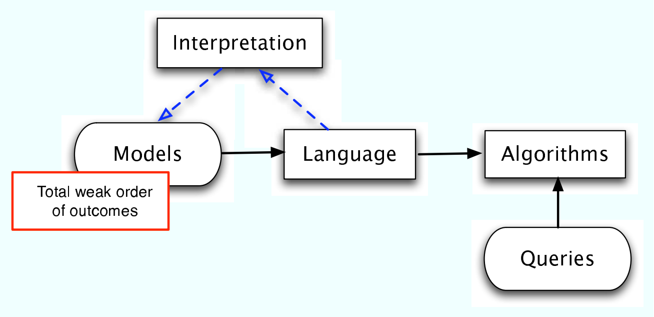
1. Introduction:
   1. Why preferences?
   2. The Meta-Model: Models, Languages, Algorithms
2. Preference Models, Languages, and Algorithms
   1. Total orders and Value Functions
   2. Partial orders and Qualitative Languages
   3. Preference Compilation
   4. Gambles and Utility functions
3. From Preference Specification to Preference Elicitation

Model = Total (Weak) Order

Simple and Natural Model

- Clear notion of optimal outcomes
- Every pair of outcomes comparable



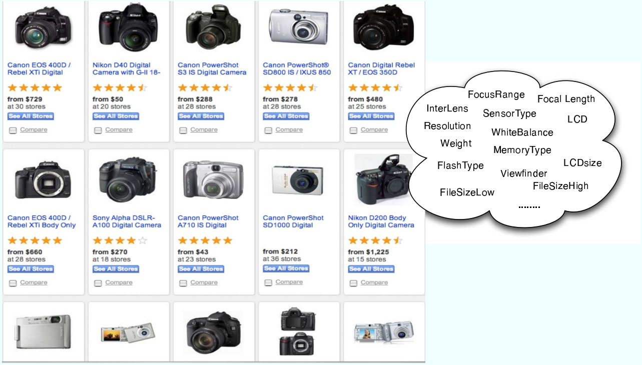


Model = Total (Weak) Order, Language = ??

Language = Model (i.e., an explicit ordering)

- Impractical except for small outcome spaces
- Cognitively difficult when outcomes involve many attributes we care about

2,707 digital cameras at shopping.com (May, 2007)

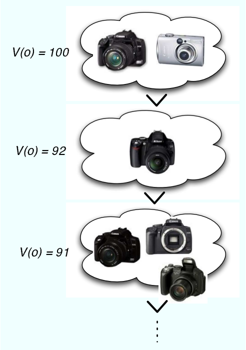


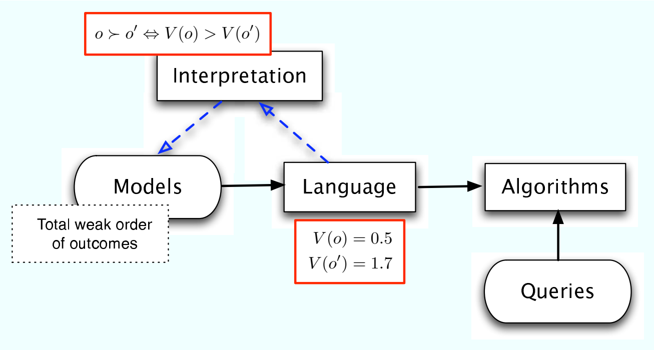
Canon EOS 400D / Rebel XTi Digital ★★★★★ from $729 at 30 stores See All Stores □ Compare	Nikon D40 Digital Camera with G-II 18- ★★★★½ from $50 at 20 stores See All Stores □ Compare	Canon PowerShot S3 IS Digital Camera ★★★★½ from $288 at 28 stores See All Stores □ Compare	Canon PowerShot® SD800 IS / IXUS 850 ★★★★½ from $278 at 28 stores See All Stores □ Compare	Canon Digital Rebel XT / EOS 350D ★★★★½ from $480 at 25 stores See All Stores □ Compare
Canon EOS 400D / Rebel XTi Body Only ★★★★★ from $660 at 28 stores See All Stores □ Compare	Sony Alpha DSLR-A100 Digital Camera ★★★★☆ from $270 at 18 stores See All Stores □ Compare	Canon PowerShot A710 IS Digital ★★★★★ from $43 at 23 stores See All Stores □ Compare	Canon PowerShot SD1000 Digital from $212 at 36 stores See All Stores □ Compare	Nikon D200 Body Only Digital Camera ★★★★½ from $1,225 at 15 stores See All Stores □ Compare

Model = Total (Weak) Order, Language = ??

Language = Value Function $V : \Omega \rightarrow R$

- Value function assigns real value (e.g, \$ value) to each outcome
- **Interpretation**: $o \succ o' \Leftrightarrow V(o) > V(o')$





$o \succ o' \Leftrightarrow V(o) > V(o')$

Interpretation

Models

Total weak order of outcomes

Language

$V(o) = 0.5$
$V(o') = 1.7$

Algorithms

Queries

Model = Total Order, Language = Value Function

Difficulties? Potential?

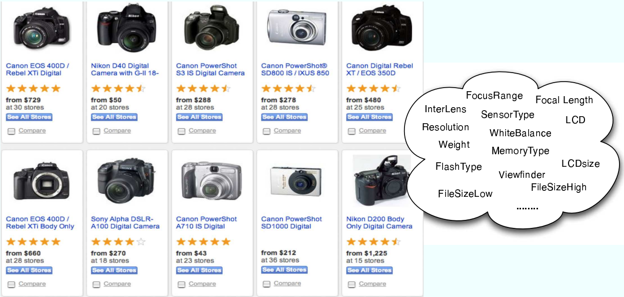
- Same difficulties as an ordering
- But ... hints at how things could be improved
- ... Could V have a compact form?
- ... Could the user's preference have some special **structure**?

Structure

Structured outcomes

1. Typically, physical outcomes Ω are described in terms of a finite set of attributes $\mathbf{X} = \{X_1, \ldots, X_n\}$
   - Attribute domains are often finite, or
   - Attribute domains continuous, but naturally ordered
2. The outcome space Ω becomes $\mathcal{X} = \times Dom(X_i)$

2,707 digital cameras at shopping.com (May, 2007)



Canon EOS 400D / Rebel XTi Digital
from $729
at 30 stores
See All Stores
□ Compare

Nikon D40 Digital Camera with G-II 18-
from $50
at 20 stores
See All Stores
□ Compare

Canon PowerShot S3 IS Digital Camera
from $288
at 28 stores
See All Stores
□ Compare

Canon PowerShot® SD800 IS / IXUS 850
from $278
at 28 stores
See All Stores
□ Compare

Canon Digital Rebel XT / EOS 350D
from $480
at 25 stores
See All Stores
□ Compare

Canon EOS 400D / Rebel XTi Body Only
from $660
at 28 stores
See All Stores
□ Compare

Sony Alpha DSLR-A100 Digital Camera
from $270
at 18 stores
See All Stores
□ Compare

Canon PowerShot A710 IS Digital
from $43
at 23 stores
See All Stores
□ Compare

Canon PowerShot SD1000 Digital
from $212
at 36 stores
See All Stores
□ Compare

Nikon D200 Body Only Digital Camera
from $1,225
at 15 stores
See All Stores
□ Compare

Structure

Structured outcomes

1. Typically, physical outcomes Ω are described in terms of a finite set of attributes $\mathbf{X} = \{X_1, \ldots, X_n\}$
   - Attribute domains are often finite, or
   - Attribute domains continuous, but naturally ordered
2. The outcome space Ω becomes $\mathcal{X} = \times Dom(X_i)$

Structured preferences

Working assumption

Informally User preferences have a lot of *regularity* (*patterns*) in terms of **X**

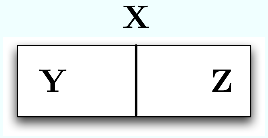
Formally User preferences induce a significant amount of preferential independence over **X**

Preferential Independence

- What is preferential independence?
  - Is it similar to probabilistic independence?
- What kinds of preferential independence?

Preferential Independence

Definitions (I)



$PI(\mathbf{Y};\mathbf{Z})$

Preferential Independence (PI)

Preference over the value of **Y** is independent of the value of **Z**

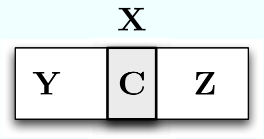
$$\forall \mathbf{y}_1, \mathbf{y}_2 \in Dom(\mathbf{Y}) :$$
$$(\exists \mathbf{z} : \mathbf{y}_1\mathbf{z} \succ \mathbf{y}_2\mathbf{z}) \;\Rightarrow \forall \mathbf{z} \in Dom(\mathbf{Z}) : \mathbf{y}_1\mathbf{z} \succ \mathbf{y}_2\mathbf{z}$$

Example: *Preferences over used cars*

Preference over **Y** $= \{$color$\}$ is independent
of the value of **Z** $= \{$mileage$\}$

Preferential Independence

Definitions (II)



$$PI(\mathbf{Y};\mathbf{Z} \mid \mathbf{C})$$

Conditional Preferential Independence (CPI)

Preference over the value of **Y** is independent of the value of **Z** given the value of **C**

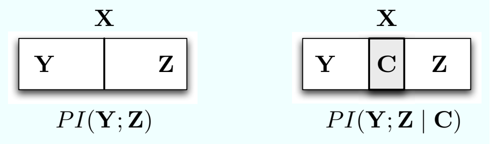
$$\forall \mathbf{y}_1, \mathbf{y}_2 \in \mathit{Dom}(\mathbf{Y}) :$$
$$(\exists \mathbf{z} : \mathbf{y}_1\mathbf{cz} \succ \mathbf{y}_2\mathbf{cz}) \;\Rightarrow \forall \mathbf{z} \in \mathit{Dom}(\mathbf{Z}) : \mathbf{y}_1\mathbf{cz} \succ \mathbf{y}_2\mathbf{cz})$$

Example: *Preferences over used cars*

Preference over **Y** $= \{$brand$\}$ is independent of **Z** $= \{$mileage$\}$ given **C** $= \{$mechanical-inspection-report$\}$.

Preferential Independence

Definitions (III)


X
Y
Z
$PI(\mathbf{Y};\mathbf{Z})$
X
Y
C
Z
$PI(\mathbf{Y};\mathbf{Z} \mid \mathbf{C})$

(Conditional) Preferential Independence

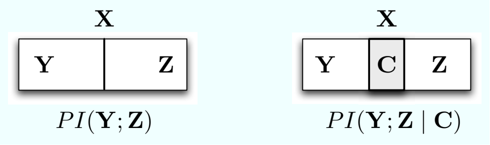
- PI/CPI are *directional*: $PI(\mathbf{Y};\mathbf{Z}) \not\Rightarrow PI(\mathbf{Z};\mathbf{Y})$
  - Example with cars: $\mathbf{Y} = \{\text{brand}\}$, $\mathbf{Z} = \{\text{color}\}$
- Strongest case: *Mutual Independence*

$$\forall \mathbf{Y} \subset \mathbf{X} : \quad PI(\mathbf{Y};\mathbf{X} \setminus \mathbf{Y})$$

- Weakest case?

Preferential Independence

How can PI/CPI help?


X
Y
Z
$PI(\mathbf{Y};\mathbf{Z})$
X
Y
C
Z
$PI(\mathbf{Y};\mathbf{Z} \mid \mathbf{C})$

Independence ⇒ Conciseness

1. Reduction in effort required for model specification
   - If $PI(\mathbf{Y};\mathbf{Z})$, then a *statement* $\mathbf{y}_1 \succ \mathbf{y}_2$ communicates $\forall \mathbf{z} \in Dom(\mathbf{Z}) : \mathbf{y}_1\mathbf{z} \succ \mathbf{y}_2\mathbf{z}$
2. Increased efficiency of reasoning?

Structure, Independence, and Value Functions

If $\Omega = \mathcal{X} = \times Dom(X_i)$ then $V : \mathcal{X} \to R$

Independence = Compact Form

- Compact form: $V(X_1, \ldots, X_n) = f(g_1(\mathbf{Y}_1), \ldots, g_k(\mathbf{Y}_k))$.
    - Potentially fewer parameters required: $O(2^k \cdot 2^{|\mathbf{Y}_i|})$ vs. $O(2^n)$.
    - OK if
        - $k \ll n$, and all $\mathbf{Y}_i$ are small subsets of $\mathbf{X}$, OR
        - f has a convenient special form

Structure, Independence, and Value Functions

If $\Omega = \mathcal{X} = \times Dom(X_i)$ then $V : \mathcal{X} \rightarrow R$

Independence = Compact Form

- Compact form: $V(X_1, \ldots, X_n) = f(g_1(\mathbf{Y}_1), \ldots, g_k(\mathbf{Y}_k))$.
  - Potentially fewer parameters required: $O(2^k \cdot 2^{|\mathbf{Y}_i|})$ vs. $O(2^n)$.
  - OK if
    - $k \ll n$, and all $\mathbf{Y}_i$ are small subsets of $\mathbf{X}$, OR
    - f has a convenient special form
- If $V(X, Y, Z) = V_1(X, Z) + V_2(Y, Z)$ then X is preferentially independent of Y given Z.

Structure, Independence, and Value Functions

If $\Omega = \mathcal{X} = \times Dom(X_i)$ then $V : \mathcal{X} \rightarrow R$

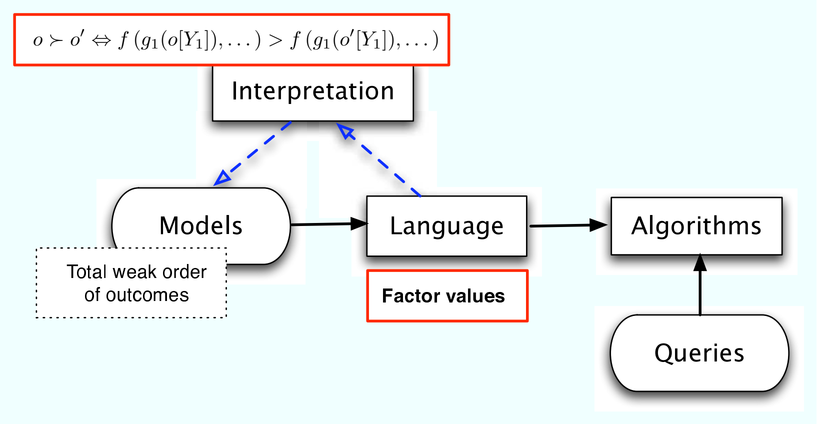
Independence = Compact Form

- Compact form: $V(X_1, \ldots, X_n) = f(g_1(\mathbf{Y}_1), \ldots, g_k(\mathbf{Y}_k))$.
  - Potentially fewer parameters required: $O(2^k \cdot 2^{|\mathbf{Y}_i|})$ vs. $O(2^n)$.
  - OK if
    - $k \ll n$, and all $\mathbf{Y}_i$ are small subsets of $\mathbf{X}$, OR
    - f has a convenient special form
- If $V(X, Y, Z) = V_1(X, Z) + V_2(Y, Z)$ then X is preferentially independent of Y given Z.
- If X is preferentially independent of Y given Z then $V(X, Y, Z) = V_1(X, Z) + V_2(Y, Z)$
  - Would be nice, but requires stronger conditions
  - In general, certain independence properties may lead to the existence of simpler form for V

Structure, Independence, and Value Functions

Independence = Compact Form

- Compact form: $V(X_1,\ldots,X_n) = f(g_1(\mathbf{Y}_1),\ldots,g_k(\mathbf{Y}_k))$.



$$o \succ o' \Leftrightarrow f\left(g_1(o[Y_1]),\ldots\right) > f\left(g_1(o'[Y_1]),\ldots\right)$$

Additive Independence

Good news

V is additively independent if
$V(X_1, \ldots, X_n) = V_1(X_1) + \cdots + V_n(X_n)$.

- $V(\text{CAMERA}) =$
  $V_1(\text{resolution}) + V_2(\text{zoom}) + V_3(\text{weight}) + \cdots$

Additive Independence

Good news

V is additively independent if
$V(X_1, \ldots, X_n) = V_1(X_1) + \cdots + V_n(X_n)$.

- $V(\text{CAMERA}) = V_1(\text{resolution}) + V_2(\text{zoom}) + V_3(\text{weight}) + \cdots$

V is additively independent only if
$X_1, \ldots, X_n$ are mutually independent.

Additive Independence is good!

- Easier to elicit – need only think of individual attributes
- Only $O(n)$ parameters required
- Easy to represent
- Easy to compute with

Additive Independence

Not so good news

V is additively independent if
$V(X_1, \ldots, X_n) = V_1(X_1) + \cdots + V_n(X_n)$.

Additive Independence is good!

- Easier to elicit – need only think of individual attributes
- Easy to represent, and easy to compute with

Additive Independence is too good to be true!

Very strong independence assumptions

- Preferences are unconditional
  - If I like my coffee with sugar, I must like my tea with sugar.
- Strength of preference is unconditional
  - If a sun-roof on my new Porsche is worth $1000,
    it's worth the same on any other car.

Generalized Additive Independence (GAI)

$V(X_1, \ldots, X_n) = V_1(\mathbf{Y}_1) + \cdots + V_k(\mathbf{Y}_k)$, where $\mathbf{Y}_i \subseteq \mathbf{X}$.

- $\mathbf{Y}_i$ is called a *factor*
- $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are not necessarily disjoint
- Number of parameters required: $O(k \cdot 2^{\max_i |Y_i|})$

Example: $V(\text{VACATION}) =$
$V_1(\text{location}, \text{season}) + V_2(\text{season}, \text{facilities}) + \cdots$

Generalized Additive Independence (GAI)

$V(X_1, \ldots, X_n) = V_1(\mathbf{Y}_1) + \cdots + V_k(\mathbf{Y}_k)$, where $\mathbf{Y}_i \subseteq \mathbf{X}$.

- $\mathbf{Y}_i$ is called a *factor*
- $\mathbf{Y}_i$ and $\mathbf{Y}_j$ are not necessarily disjoint
- Number of parameters required: $O(k \cdot 2^{\max_i |Y_i|})$

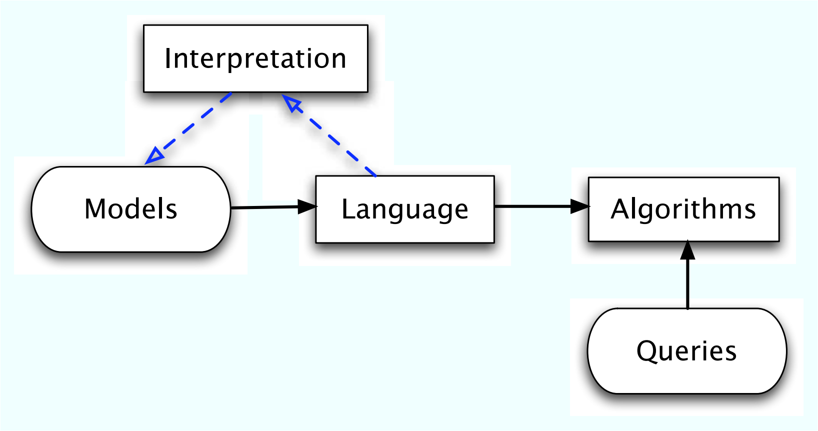
Example: $V(\text{VACATION}) =$
$V_1(\text{location}, \text{season}) + V_2(\text{season}, \text{facilities}) + \cdots$

GAI value functions are very general

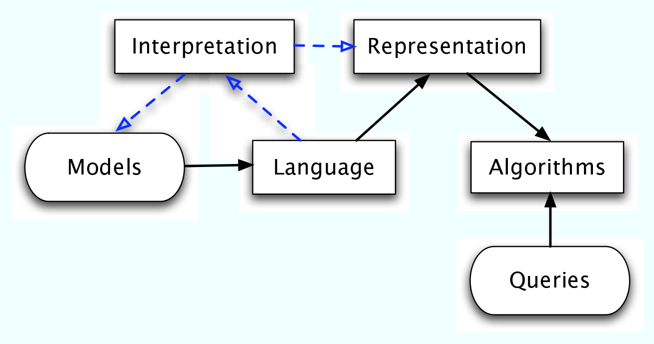
♠ Factors $Y_1, \ldots, Y_k$ do not have to be disjoint!

- One extreme – single factor
- Other extreme – n unary factors $\mathbf{Y}_i = X_i$ (additive independence)
- Interesting case – $O(n)$ factors where $|\mathbf{Y}_i| = O(1)$.

Recalling the Meta-Model



Meta-Model: The Final Element



Interpretation

Representation

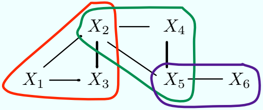
Models

Language

Algorithms

Queries

$$\begin{aligned} V(X_1,\ldots,X_6) &= g_1(X_1,X_2,X_3)+\\ &\quad g_2(X_2,X_4,X_5)+\\ &\quad g_3(X_5,X_6)\end{aligned}$$



Graphical Representation and Algorithms

Queries for which graphical representation is not needed

Compare outcomes Assign utilities and compare.

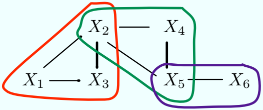
Order items Assign utilities and sort.

Queries for which graphical representation **might help**

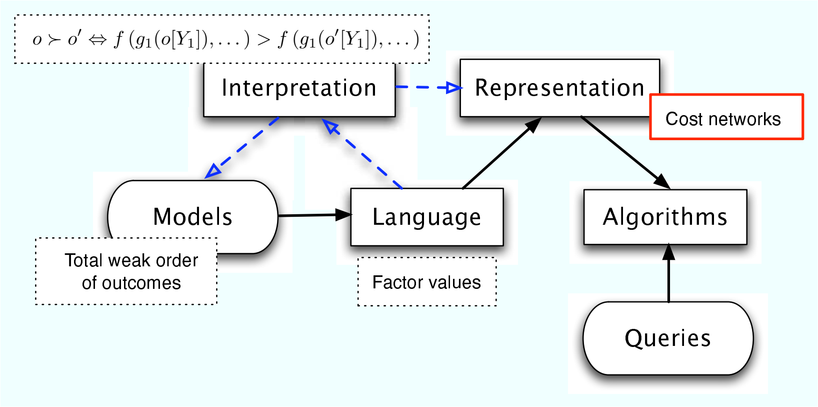
Finding **X** values maximizing *V*

1. Instance of standard constraint optimization (COP)
2. Cost network topology is crucial for efficiency of COP
3. GAI structure $\equiv$ Cost network topology

$$\begin{aligned} V(X_1,\dots,X_6) &= g_1(X_1,X_2,X_3) + \\ &\quad g_2(X_2,X_4,X_5) + \\ &\quad g_3(X_5,X_6) \end{aligned}$$



Graphical Representation of GAI Value Functions


$o \succ o' \Leftrightarrow f\left(g_1(o[Y_1]), \ldots\right) > f\left(g_1(o'[Y_1]), \ldots\right)$
Interpretation
Representation
Cost networks
Models
Language
Algorithms
Total weak order of outcomes
Factor values
Queries

Bibliography

F. Bacchus and A. Grove.
Graphical models for preference and utility.
In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 3–10, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

S. Bistarelli, H. Fargier, U. Montanari, F. Rossi, T. Schiex, and G. Verfaillie.
Semiring-based CSPs and valued CSPs: Frameworks, properties, and comparison.
Constraints, 4(3):275–316, September 1999.

C. Boutilier, F. Bacchus, and R. I. Brafman.
UCP-networks: A directed graphical representation of conditional utilities.
In *Proceedings of Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 56–64, 2001.

R. Dechter.
Constraint Processing.
Morgan Kaufmann, 2003.

P. C. Fishburn.
Utility Theory for Decision Making.
John Wiley & Sons, 1969.

P. C. Fishburn.
The Foundations of Expected Utility.
Reidel, Dordrecht, 1982.

C. Gonzales and P. Perny.
Gai networks for utility elicitation.
In *Proceedings of the International Conference on Knowledge Representation and Reasoning (KR)*, pages 224–234, 2004.

Bibliography

P. E. Green, A. M. Krieger, and Y. Wind.
Thirty years of conjoint analysis: Reflections and prospects.
Interfaces, 31(3):56–73, 2001.

R. L. Keeney and H. Raiffa.
Decision with Multiple Objectives: Preferences and Value Tradeoffs.
Wiley, 1976.

A. Tversky.
A general theory of polynomial conjoint measurement.
Journal of Mathematical Psychology, 4:1–20, 1967.

Outline

1. Introduction:
   1. Why preferences?
   2. The Meta-Model: Models, Languages, Algorithms
2. Preference Models, Languages, and Algorithms
   1. Total orders and Value Functions
   2. Partial orders and Qualitative Languages
   3. Preference Compilation
   4. Gambles and Utility functions
3. From Preference Specification to Preference Elicitation

Starting with the Language

Language choices crucial in practice

- Language: main interface between user and system
- Inappropriate language: forget about lay users
- GAI value functions are *not* for lay users
- Questions:
  - What is a good language?
  - How far can we go with it?

Starting with the Language

Language choices crucial in practice

- Language: main interface between user and system
- Inappropriate language: forget about lay users
- GAI value functions are *not* for lay users
- Questions:
  - What is a good language?
  - How far can we go with it?

What would be an ”ultimate” language?

1. Based on information that’s
   - cognitively easy to reflect upon, and
   - has a common sense interpretation semantics
2. Compactly specifies natural orderings
3. Computationally efficient reasoning
   - complexity = F(language, query)

Qualitative Preference Statements

From natural language to logics

What qualitative statements can we expect users to provide?

- comparison between pairs of complete alternatives
  - “I prefer this car to that car”
- information-revealing critique of certain alternatives
  - “I prefer a car similar to this one but without the sunroof”
- ...
- generalizing preference statements over some attributes
  - “In a minivan, I prefer automatic transmission to manual transmission”
  - $mv \wedge a \succ mv \wedge m$

Qualitative Preference Statements

From natural language to logics

Language = Qualitative preference expressions over **X**

User provides the system with a **preference expression**

$$S = \{s_1, \ldots, s_m\} = \{\langle \varphi_1 \oslash_1 \psi_1 \rangle, \cdots, \langle \varphi_m \oslash_m \psi_m \rangle\}$$

consisting of a set of **preference statements** $s_i = \varphi_i \oslash_i \psi_i$, where

- φ_i, ψ_i are some logical formulas over **X**,
- $\oslash_i \in \{\succ, \succeq, \sim\}$, and
- $\succ$, $\succeq$, and $\sim$ have the standard semantics of strong preference, weak preference, and preferential equivalence, respectively.

Generalizing Preference Statements

Examples

s_1 SUV is at least as good as a minivan

- $X_{type} = SUV \succeq X_{type} = minivan$

s_2 In a minivan, I prefer automatic transmission to manual transmission

- $X_{type} = minivan \wedge X_{trans} = automatic \succ$ $X_{type} = minivan \wedge X_{trans} = manual$

Generalizing Preference Statements

One generalizing statement can encode many comparisons

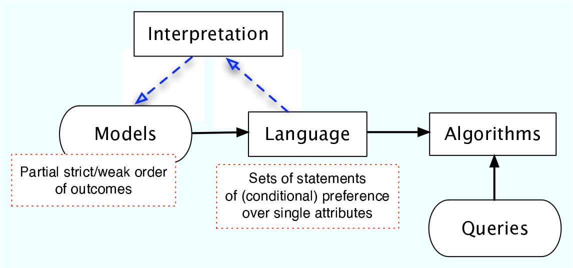
”Minivan with automatic transmission is better than one with manual transmission” implies (?)

- Red minivan with automatic transmission is better than Red minivan with manual transmission
- Red, hybrid minivan with automatic transmission is better than Red hybrid minivan with manual transmission
- . . .

Generalized statements and independence seem closely related

Showcase: Statements of Conditional Preference

Model + Language + Interpretation + Representation + Algorithms


Interpretation
Models
Language
Algorithms
Partial strict/weak order of outcomes
Sets of statements of (conditional) preference over single attributes
Queries

Language

- I prefer an SUV to a minivan
- In a minivan, I prefer automatic transmission to manual transmission

$$S = \{\, \mathbf{y} \wedge x_i \succ \mathbf{y} \wedge x_j \mid X \in \mathbf{X}, \mathbf{Y} \subseteq \mathbf{X} \setminus \{X\}, x_i, x_j \in Dom(X), \mathbf{y} \in Dom(\mathbf{Y}) \,\}$$

Dilemma of Statement Interpretation

I prefer an SUV to a minivan

What information does this statement convey about the model?

Totalitarianism Ignore the unmentioned attributes

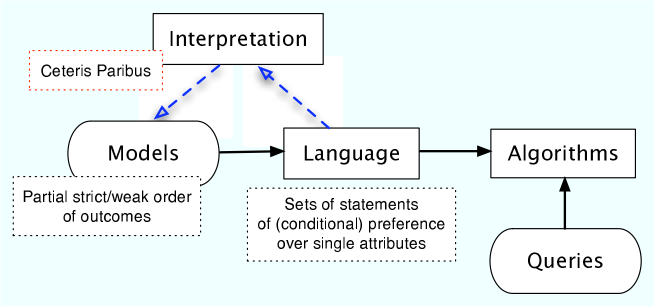
- *Any* SUV is preferred to *any* minivan

Ceteris Paribus Fix the unmentioned attributes

- An SUV is preferred to a minivan, provided that otherwise the two cars are *similar (identical)*

Other? ... Somewhere in between the two extremes?

From Statement to Expression Interpretation


Interpretation
Ceteris Paribus
Models
Language
Algorithms
Partial strict/weak order of outcomes
Sets of statements of (conditional) preference over single attributes
Queries

Given expression $S = \{s_1, \ldots, s_m\}$

- Each s_i induces a strict partial order $\succ_i$ over Ω
- What does $\succ_1, \ldots, \succ_m$ tell us about the model $\succ$?
    - Natural choice: $\succ \; = \; \mathrm{TC}[\cup_i \succ_i]$
    - In general, more than one alternative

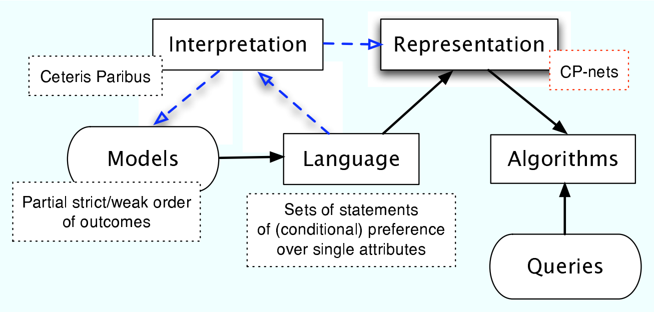
Representation
CP-nets

CP-nets – from expressions *S* to annotated directed graphs

Nodes

Edges

Annotation


Interpretation
Representation
Ceteris Paribus
CP-nets
Models
Language
Algorithms
Partial strict/weak order of outcomes
Sets of statements of (conditional) preference over single attributes
Queries

Representation

CP-nets

CP-nets – from expressions S to **annotated directed graphs**

Nodes Attributes **X**

Edges Direct preferential dependencies induces by S

- Edge $X_j \rightarrow X_i$ iff preference over $Dom(X_i)$ vary with values of X_j

Annotation Each node $X_i \in \mathbf{X}$ is annotated with statements of preference $S_i \subseteq S$ over $Dom(X_i)$

- Note: the language implies $S_i \cap S_j = \emptyset$

Example

Preference expression

s_1 *I prefer red minivans to white minivans.*

s_2 *I prefer white SUVs to red SUVs.*

s_3 *In white cars I prefer a dark interior.*

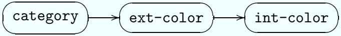
s_4 *In red cars I prefer a bright interior.*

s_5 *I prefer minivans to SUVs.*

Outcome space

	`category`	`ext-color`	`int-color`
t_1	minivan	red	bright
t_2	minivan	red	dark
t_3	minivan	white	bright
t_4	minivan	white	dark
t_5	SUV	red	bright
t_6	SUV	red	dark
t_7	SUV	white	bright
t_8	SUV	white	dark

CP-net

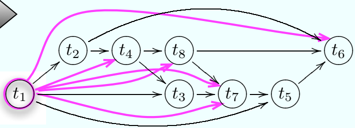


$C_{mv} \succ C_{suv}$

C_{mv}	$E_r \succ E_w$
C_{suv}	$E_w \succ E_r$

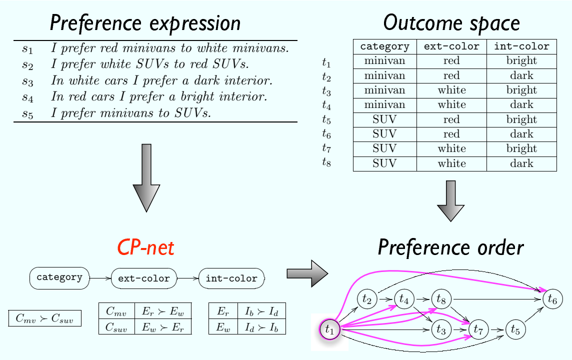
E_r	$I_b \succ I_d$
E_w	$I_d \succ I_b$

Preference order



Example

Conditional preferential independence



Preference expression

- s_1 *I prefer red minivans to white minivans.*
- s_2 *I prefer white SUVs to red SUVs.*
- s_3 *In white cars I prefer a dark interior.*
- s_4 *In red cars I prefer a bright interior.*
- s_5 *I prefer minivans to SUVs.*

Outcome space

	category	ext-color	int-color
t_1	minivan	red	bright
t_2	minivan	red	dark
t_3	minivan	white	bright
t_4	minivan	white	dark
t_5	SUV	red	bright
t_6	SUV	red	dark
t_7	SUV	white	bright
t_8	SUV	white	dark

CP-net

$C_{mv} \succ C_{suv}$

C_{mv}	$E_r \succ E_w$
C_{suv}	$E_w \succ E_r$

E_r	$I_b \succ I_d$
E_w	$I_d \succ I_b$

Preference order

Principle: *Assume independence wherever possible!*

- Here: assumes preference over **int-color** is independent of **category** given **ext-color**

What is the Graphical Representation Good For?

CP-nets

Syntactic sugar, useful tool, or both?

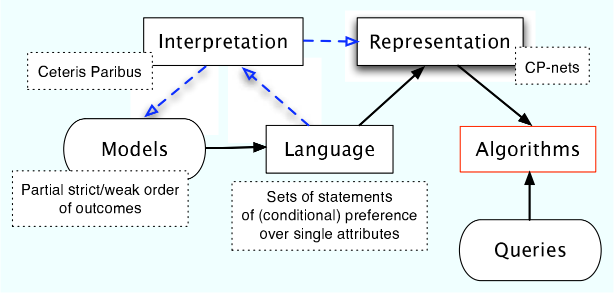
1. Convenient “map of independence”
2. Classifies preference expressions based on induced graphical structure
   - Other classifications possible
   - This one is useful!

Fact: Plays an important role in computational analysis

- Helps identifying tractable classes
- Plays a role in efficient algorithms and informed heuristics

Complexity and Algorithms for Queries on CP-nets

... and the role of graphical representation



Interpretation
Representation
Ceteris Paribus
CP-nets
Models
Language
Algorithms
Partial strict/weak order of outcomes
Sets of statements of (conditional) preference over single attributes
Queries

Various queries

Verification Does S convey an ordering?

Optimization Find $o \in \Omega$, such that $\forall o' \in \Omega : o' \not\succ o$.

Comparison Given $o, o' \in \Omega$, does $S \models o \succ o'$?

Sorting Given $\Omega' \subseteq \Omega$, order Ω' consistently with S.

Complexity and Algorithms for Queries on CP-nets

... and the role of graphical representation

Various queries

Verification Does S convey an ordering?

- “YES” for acyclic CP-nets (no computation!)
- Tractable for *certain* classes of cyclic CP-nets

Optimization Find $o \in \Omega$, such that $\forall o' \in \Omega : o' \not\succ o$.

- Linear time for acyclic CP-nets.
- Tractable for *certain* classes of cyclic CP-nets

Comparison Given $o, o' \in \Omega$, does $S \models o \succ o'$?

Sorting Given $\Omega' \subseteq \Omega$, order Ω' consistently with S.

Pairwise Comparison (in CP-nets)

Given $o, o' \in \Omega$, does $S \models o \succ o'$?

Boolean variables

Graph topology	**Comparison**
Directed Tree	$O(n^2)$
Polytree (indegree $\leq k$)	$O(2^{2k} n^{2k+3})$
Polytree	NP-complete
Singly Connected (indegree $\leq k$)	NP-complete
DAG	NP-complete
General case	PSPACE-complete

Multi-valued variables

Catastrophe ...

Complexity and Algorithms for Queries on CP-nets

... and the role of graphical representation

Various queries

Verification Does S convey an ordering?

- "YES" for acyclic CP-nets (no computation!)
- Tractable for *certain* classes of cyclic CP-nets

Optimization Find $o \in \Omega$, such that $\forall o' \in \Omega : o' \not\succ o$.

- Linear time for acyclic CP-nets.
- Tractable for *certain* classes of cyclic CP-nets

Comparison Given $o, o' \in \Omega$, does $S \models o \succ o'$?

- Bad ... mostly NP-hard
- Still, some restricted tractable classes exist

Sorting Given $\Omega' \subseteq \Omega$, order Ω' consistently with S.

- Bad ??

Ordering vs. Comparison

CP-nets

Hypothesis: Ordering is as hard as comparison

Pairwise comparison between objects is a basic operation of any sorting procedure

Ordering vs. Comparison

CP-nets

Hypothesis: Ordering is as hard as comparison

Pairwise comparison between objects is a basic operation of any sorting procedure

Observation

To order a pair of alternatives $o, o' \in \Omega$ consistently with S, it suffices to know only that either $S \not\models o \succ o'$ or $S \not\models o' \succ o$

- Note: In partial order models, knowing $S \not\models o' \succ o$ is weaker than knowing $S \models o \succ o'$
- Helps?

Ordering vs. Comparison

CP-nets

Hypothesis: Ordering is as hard as comparison

Pairwise comparison between objects is a basic operation of any sorting procedure

Observation

To order a pair of alternatives $o, o' \in \Omega$ consistently with S, it suffices to know only that either $S \not\models o \succ o'$ or $S \not\models o' \succ o$

Fact: For acyclic CP-nets, the hypothesis is WRONG!

1. Deciding $(S \not\models o \succ o') \vee (S \not\models o' \succ o)$ — in time $O(|\mathbf{X}|)$
2. This decision procedure can be used to sort *any* $\Omega' \subseteq \Omega$ in time $O(|\mathbf{X}| \cdot |\Omega'| \log |\Omega'|)$

Pairwise Ordering vs. Pairwise Comparison

Boolean variables

Graph topology	**Comparison**
Directed Tree	$O(n^2)$
Polytree (indegree $\leq k$)	$O(2^{2k} n^{2k+3})$
Polytree	NP-complete
Singly Connected (indegree $\leq k$)	NP-complete
DAG	NP-complete
General case	PSPACE-complete

Multi-valued variables

Catastrophe ...

Pairwise Ordering vs. Pairwise Comparison

Boolean variables

Graph topology	Ordering
Directed Tree	$O(n)$
Polytree (indegree $\leq k$)	$O(n)$
Polytree	$O(n)$
Singly Connected (indegree $\leq k$)	$O(n)$
DAG	$O(n)$
General case	NP-hard

Multi-valued variables

Same complexity as for boolean variable!

Bibliography

S. Benferhat, D. Dubois, and H. Prade.
Towards a possibilistic logic handling of preferences.
Applied Intelligence, pages 303–317, 2001.

C. Boutilier.
Toward a logic for qualitative decision theory.
In *Proceedings of the Third Conference on Knowledge Representation (KR–94)*, pages 75–86, Bonn, 1994.

C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole.
CP-nets: A tool for representing and reasoning about conditional *ceteris paribus* preference statements.
Journal of Artificial Intelligence Research, 21:135–191, 2004.

C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole.
Preference-based constrained optimization with CP-nets.
Computational Intelligence (Special Issue on Preferences in AI and CP), 20(2):137–157, 2004.

C. Boutilier, R. Brafman, H. Hoos, and D. Poole.
Reasoning with conditional *ceteris paribus* preference statements.
In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 71–80. Morgan Kaufmann Publishers, 1999.

R. Brafman, C. Domshlak, and S. E. Shimony.
On graphical modeling of preference and importance.
Journal of Artificial Intelligence Research, 25:389–424, 2006.

R. I. Brafman and Y. Dimopoulos.
Extended semantics and optimization algorithms for cp-networks.
Computational Intelligence (Special Issue on Preferences in AI and CP), 20(2):218–245, 2004.

Bibliography

G. Brewka.
Reasoning about priorities in default logic.
In *Proceedings of Sixth National Conference on Artificial Intelligence*, pages 940–945. AAAI Press, 1994.

G. Brewka.
Logic programming with ordered disjunction.
In *Proceedings of Eighteenth National Conference on Artificial Intelligence*, pages 100–105, Edmonton, Canada, 2002. AAAI Press.

G. Brewka, I. Niemelä, and M. Truszczynski.
Answer set optimization.
In *Proceedings of of the Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.

J. Chomicki.
Preference formulas in relational queries.
ACM Transactions on Database Systems, 28(4):427–466, 2003.

J. Delgrande and T. Schaub.
Expressing preferences in default logic.
Artificial Intelligence, 123(1-2):41–87, 2000.

C. Domshlak, S. Prestwich, F. Rossi, K. B. Venable, and T. Walsh.
Hard and soft constraints for reasoning about qualitative conditional preferences.
Journal of Heuristics, 12(4-5):263–285, 2006.

J. Doyle and R. H. Thomason.
Background to qualitative decision theory.
AI Magazine, 20(2):55–68, 1999.

Bibliography

J. Doyle and M. Wellman.
Representing preferences as ceteris paribus comparatives.
In *Proceedings of the AAAI Spring Symposium on Decision-Theoretic Planning*, pages 69–75, March 1994.

S. O. Hansson.
The Structure of Values and Norms.
Cambridge University Press, 2001.

U. Junker.
Preference programming: Advanced problem solving for configuration.
Artificial Intelligence for Engineering, Design, and Manufacturing, 17, 2003.

J. Lang.
Logical preference representation and combinatorial vote.
Annals of Mathematics and Artificial Intelligence, 42(1-3):37–71, 2004.

Y. Shoham.
A semantics approach to non-monotonic logics.
In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 388–392, 1987.

S. W. Tan and J. Pearl.
Qualitative decision theory.
In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 928–933, Seattle, 1994. AAAI Press.

M. Wellman.
Fundamental concepts of qualitative probabilistic networks.
Artificial Intelligence, 44:257–304, 1990.

Bibliography

M. Wellman and J. Doyle.
Preferential semantics for goals.
In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 698–703, July 1991.

N. Wilson.
Consistency and constrained optimisation for conditional preferences.
In *Proceedings of the Sixteenth European Conference on Artificial Intelligence*, pages 888–894, Valencia, 2004.

N. Wilson.
Extending CP-nets with stronger conditional preference statements.
In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 735–741, San Jose, CL, 2004.

Outline

1. Introduction:
   1. Why preferences?
   2. The Meta-Model: Models, Languages, Algorithms

2. Preference Models, Languages, and Algorithms
   1. Total orders and Value Functions
   2. Partial orders and Qualitative Languages
   3. Preference Compilation
   4. Gambles and Utility functions

3. From Preference Specification to Preference Elicitation

Language and Reasoning

What language should we select?

Expressions in preference logic

- + Flexible and cognitively easy to reflect upon
- - Doesn’t have a (single) common sense interpretation semantics
- - Generally hard comparison and ordering of outcomes OR specifically restricted language

Value functions

- + Has a common sense interpretation semantics
- + Tractable comparison and ordering of outcomes
- - Cognitively hard to reflect upon ...

Language and Reasoning

What language should we select?

Expressions in preference logic

- + Flexible and cognitively easy to reflect upon
- - Doesn’t have a (single) common sense interpretation semantics
- - Generally hard comparison and ordering of outcomes OR specifically restricted language

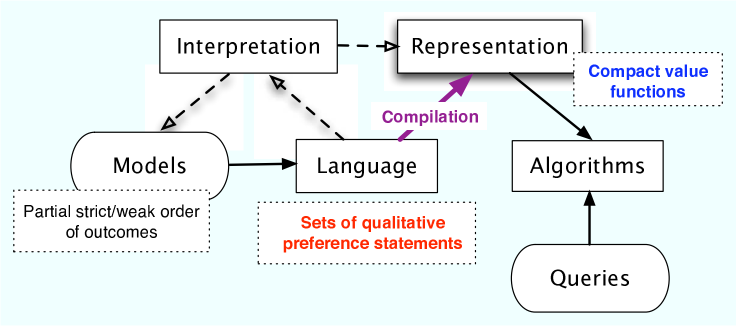
Value functions

- + Has a common sense interpretation semantics
- + Tractable comparison and ordering of outcomes
- - Cognitively hard to reflect upon ...

Can we benefit of both worlds?

Representation to the Rescue

Language = Qualitative Statements, Representation = Compact Value Functions



Preference Compilation

Given a preference expression $S = \{s_1, \ldots, s_m\}$ in terms of **X**, generate a value function $V : \mathcal{X} \mapsto \mathbb{R}$ such that

$$S \models o \succ o' \quad \Rightarrow \quad V(o) > V(o')$$

Structure-based Value-Function Compilation

Structure-based Compilation Methodology

1. **Restrict the language** to a certain class of expressions
   - Acyclic CP-nets OR Acyclic CP-nets + $\{o \succ o'\}$ OR ...
2. **Fix semantics** of these expressions
   - Typically involves various independence assumptions
3. Provide a **representation theorem**

   Given a statement S in the chosen class,
   if there exists a value function V that models S, then

   - there exists a compact value function V_c that models S
4. Provide a **compilation theorem**

   Given a statement S in the chosen class,
   if there exists a value function V that models S, then

   - V_c can be efficiently generated from S.

Preference Compilation Map

CP-nets

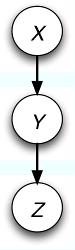
Language	Acyclic CP-nets
Compactness	In-degree $O(1)$
Efficiency	Markov blanket $O(1)$
Sound?	YES
Complete?	YES

$$\boxed{V(X,Y,Z) = V_X(X) + V_Y(Y,X) + V_Z(Z,Y)}$$

$x_1 \succ x_2$

$x_1 : y_1 \succ y_2$
$x_2 : y_2 \succ y_1$

$y_1 : z_1 \succ z_2$



V_X

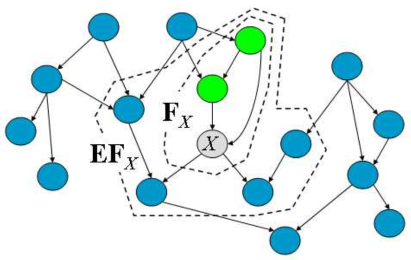
$x_1 \rightarrow 20$
$x_2 \rightarrow 5$

V_Y

$x_1, y_1 \rightarrow 20$
$x_1, y_2 \rightarrow 17$
$x_2, y_1 \rightarrow 17$
$x_2, y_2 \rightarrow 20$

V_Z

$y_1, z_1 \rightarrow 6$
$y_1, z_2 \rightarrow 5$
$y_2, z_1 \rightarrow 6$
$y_2, z_2 \rightarrow 6$



Preference Compilation Map

CP-nets

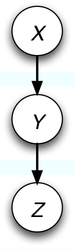
Language	Acyclic CP-nets	Cyclic CP-nets
Compactness	In-degree $O(1)$	In-degree $O(1)$
Efficiency	Markov blanket $O(1)$	Markov blanket $O(1)$
Sound?	YES	YES
Complete?	YES	NO

$$\boxed{V(X,Y,Z) = V_X(X) + V_Y(Y,X) + V_Z(Z,Y)}$$

$\boxed{x_1 \succ x_2}$

$\boxed{\begin{array}{l} x_1 : y_1 \succ y_2 \\ x_2 : y_2 \succ y_1 \end{array}}$

$\boxed{y_1 : z_1 \succ z_2}$



V_X

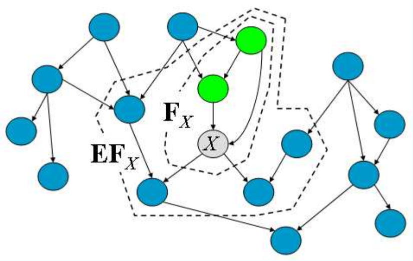
x_1	$\rightarrow 20$
x_2	$\rightarrow 5$

V_Y

x_1, y_1	$\rightarrow 20$
x_1, y_2	$\rightarrow 17$
x_2, y_1	$\rightarrow 17$
x_2, y_2	$\rightarrow 20$

V_Z

y_1, z_1	$\rightarrow 6$
y_1, z_2	$\rightarrow 5$
y_2, z_1	$\rightarrow 6$
y_2, z_2	$\rightarrow 6$



Preference Compilation Map

CP-nets

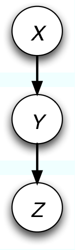
Language	Acyclic CP-nets	Cyclic CP-nets	Acyclic CP-nets + $\{o \succ o'\}$
Compactness	In-degree $O(1)$	In-degree $O(1)$	In-degree $O(1)$
Efficiency	Markov blanket $O(1)$	Markov blanket $O(1)$	Markov blanket $O(1)$
Sound?	YES	YES	YES
Complete?	YES	NO	NO

$$\boxed{V(X,Y,Z) = V_X(X) + V_Y(Y,X) + V_Z(Z,Y)}$$

$x_1 \succ x_2$

$x_1 : y_1 \succ y_2$
$x_2 : y_2 \succ y_1$

$y_1 : z_1 \succ z_2$



V_X

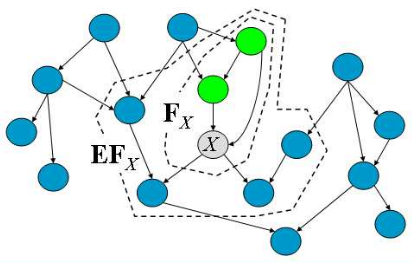
x_1	$\rightarrow 20$
x_2	$\rightarrow 5$

V_Y

x_1, y_1	$\rightarrow 20$
x_1, y_2	$\rightarrow 17$
x_2, y_1	$\rightarrow 17$
x_2, y_2	$\rightarrow 20$

V_Z

y_1, z_1	$\rightarrow 6$
y_1, z_2	$\rightarrow 5$
y_2, z_1	$\rightarrow 6$
y_2, z_2	$\rightarrow 6$



How is it done?

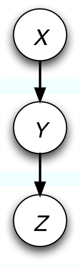
1. Given a CP-net *N*, construct a system of linear constraints L_N, variables of which correspond to the factor values (= entries of the CP-tables)
2. Pick *any* solution for L_N

$$\boxed{V(X,Y,Z) = V_X(X) + V_Y(Y,X) + V_Z(Z,Y)}$$

$x_1 \succ x_2$

$x_1 : y_1 \succ y_2$
$x_2 : y_2 \succ y_1$

$y_1 : z_1 \succ z_2$



V_X

$x_1 \rightarrow 20$
$x_2 \rightarrow 5$

V_Y

$x_1, y_1 \rightarrow 20$
$x_1, y_2 \rightarrow 17$
$x_2, y_1 \rightarrow 17$
$x_2, y_2 \rightarrow 20$

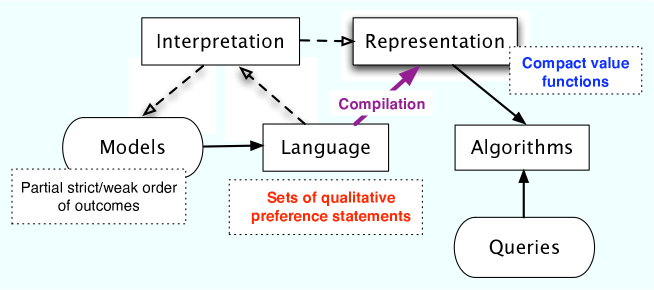
V_Z

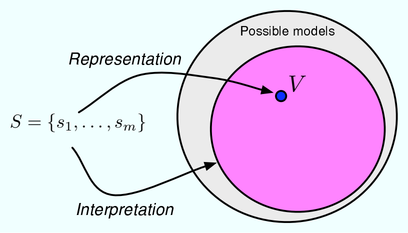
$y_1, z_1 \rightarrow 6$
$y_1, z_2 \rightarrow 5$
$y_2, z_1 \rightarrow 6$
$y_2, z_2 \rightarrow 6$

L_N

$$V_X(x_1) - V_X(x_2) > V_Y(y_1, x_2) - V_Y(y_1, x_1)$$
$$V_X(x_1) - V_X(x_2) > V_Y(y_2, x_2) - V_Y(y_2, x_1)$$
$$\ldots$$

Query Oriented Representation





Structure ...

The Pitfalls of Structure-based Compilation Methodology

1. Language is usually restrictive
2. Greatly influenced by the choice of attributes **X**
3. System makes rigid assumptions w.r.t. statement interpretation.
   - These assumptions make it harder to satisfy a sufficiently heterogeneous set of statements

Structureless Value-Function Compilation

Fundamental Question

Can we have value-function compilation in which

- The language is as general as possible
- The semantics makes as few commitments as possible, while remaining reasonable
- The target representation is efficiently generated and used

Recall that ...

Attribution **X** is just one (out of many) ways to describe the outcomes, and thus it does not necessarily corresponds to the criteria that affect user preferences over the actual physical outcomes.

Escaping the requirement for structure

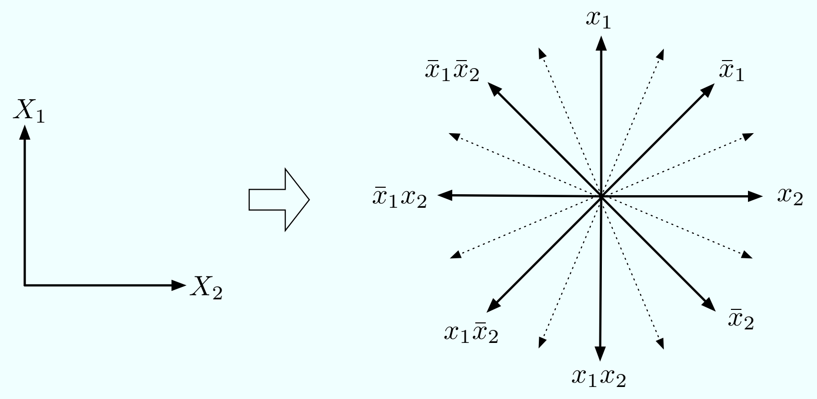
Since no independence information in the original space $\mathcal{X}$ should be expected, may be we should work in a different space in which no such information is required?

From Attributes to Factors

Assume boolean attributes **X** ...

$\Phi : \mathcal{X} \mapsto \mathcal{F} = \mathbb{R}^{4n}$

$\mathfrak{f}_i \overset{1-1}{\longleftrightarrow} \text{val}(\mathfrak{f}_i) \subseteq \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\}$

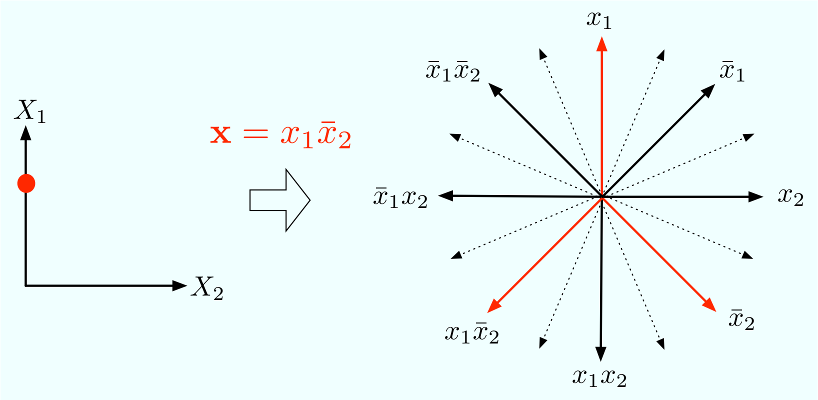


From Attributes to Factors

Assume boolean attributes **X** ...

$$\Phi : \mathcal{X} \mapsto \mathcal{F} = \mathbb{R}^{4n}$$

$$\Phi(\mathbf{x})[i] = \begin{cases} 1, & \mathrm{val}(\mathfrak{f}_i) \subseteq \mathbf{x} \\ 0, & \text{otherwise} \end{cases}$$



What is the Semantics of the Abstraction $\mathcal{F}$?

Basic Idea

Semantics

Any preference-related criterion expressible in terms of **X** corresponds to a single feature in **F**.

Value Functions in $\mathcal{F}$

Additive Decomposibility

Any preference ordering $\succeq$ over $\mathcal{X}$ is **additively decomposable** in $\mathcal{F}$. That is, for any $\succeq$ over $\mathcal{X}$, there exists a linear function

$$V\left(\Phi(\mathbf{x})\right) = \sum_{i=1}^{4^n} w_i \; \Phi(\mathbf{x})[i]$$

satisfying

$$\mathbf{x} \succeq \mathbf{x}' \quad \Leftrightarrow \quad V\left(\Phi(\mathbf{x})\right) \geq V\left(\Phi(\mathbf{x}')\right)$$

Value Functions in $\mathcal{F}$

Additive Decomposibility

Any preference ordering $\succeq$ over $\mathcal{X}$ is **additively decomposable** in $\mathcal{F}$. That is, for any $\succeq$ over $\mathcal{X}$, there exists a linear function

$$V\left(\Phi(\mathbf{x})\right) = \sum_{i=1}^{4^n} w_i \; \Phi(\mathbf{x})[i]$$

satisfying

$$\mathbf{x} \succeq \mathbf{x}' \quad \Leftrightarrow \quad V\left(\Phi(\mathbf{x})\right) \geq V\left(\Phi(\mathbf{x}')\right)$$

But is it of any practical use??

- Postpone the discussion of complexity
- Focus of preference expression interpretation.

Interpretation of Preference Statements

Statements in Expression $S = \{s_1, \ldots, s_m\}$

Suppose you are rich :)

1. **Comparative**
   - Red color is better for sport cars than white color
2. Classificatory
   - Brown color for sport cars is the worst
3. High-order
   - For sport cars, I prefer white color to brown color more than I prefer red color to white color

Statement Interpretation in $\mathcal{F}$

Marginal Values of Preference-Related Criteria

Observe that each coefficient w_i in

$$V\left(\Phi(\mathbf{x})\right) = \sum_{i=1}^{4^n} w_i \; \Phi(\mathbf{x})[i]$$

can be seen as capturing the “marginal value” of the criterion $\mathfrak{f}_i$ (and this “marginal value” only).

Statement Interpretation in $\mathcal{F}$

Framework

- $\varphi \succ \psi$
- Variable in φ: $\mathbf{X}_\varphi \subseteq \mathbf{X}$
- Models of φ: $M(\varphi) \subseteq Dom(\mathbf{X}_\varphi)$

Example

- $(X_1 \vee X_2) \succ (\neg X_3)$
- $\mathbf{X}_\varphi = \{X_1, X_2\}, \mathbf{X}_\psi = \{X_3\}$
- $M(\varphi) = \{x_1 x_2, \overline{x_1} x_2, x_1 \overline{x_2}\}$, $M(\psi) = \{\overline{x_3}\}$

Statement Interpretation in $\mathcal{F}$

Framework

- $\varphi \succ \psi$
- Variable in φ: $\mathbf{X}_\varphi \subseteq \mathbf{X}$
- Models of φ: $M(\varphi) \subseteq Dom(\mathbf{X}_\varphi)$

Example

- $(X_1 \vee X_2) \succ (\neg X_3)$
- $\mathbf{X}_\varphi = \{X_1, X_2\}, \mathbf{X}_\psi = \{X_3\}$
- $M(\varphi) = \{x_1x_2, \overline{x_1}x_2, x_1\overline{x_2}\}$, $M(\psi) = \{\overline{x_3}\}$

$$\forall m \in M(\varphi), \forall m' \in M(\psi) : \sum_{\mathfrak{f}_i : \mathrm{val}(\mathfrak{f}_i) \in 2^m} w_i > \sum_{\mathfrak{f}_j : \mathrm{val}(\mathfrak{f}_j) \in 2^{m'}} w_j$$

$$w_{x_1} + w_{x_2} + w_{x_1x_2} > w_{\overline{x_3}}$$
$$w_{x_1} + w_{\overline{x_2}} + w_{x_1\overline{x_2}} > w_{\overline{x_3}}$$
$$w_{\overline{x_1}} + w_{x_2} + w_{\overline{x_1}x_2} > w_{\overline{x_3}}$$

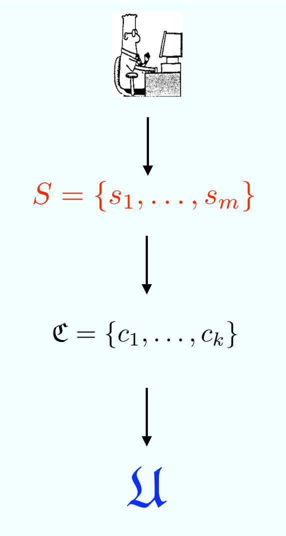
From Statements to Value Function

Good news

$$\forall m \in M(\varphi), \forall m' \in M(\psi) :$$

$$\sum_{\mathfrak{f}_i : \mathrm{val}(\mathfrak{f}_i) \in 2^m} w_i \; > \; \sum_{\mathfrak{f}_j : \mathrm{val}(\mathfrak{f}_j) \in 2^{m'}} w_j$$

1. All constraints in $\mathfrak{C}$ are linear
2. Any solution of $\mathfrak{C}$ gives us a value function V as required
3. $\mathfrak{C}$ corresponds to a very least-committing interpretation of the expression S



$S = \{s_1, \ldots, s_m\}$

$\mathfrak{C} = \{c_1, \ldots, c_k\}$

$\mathfrak{U}$

Bad News – Complexity of $\mathfrak{C}$

$$\varphi \succ \psi \implies \forall m \in M(\varphi), \forall m' \in M(\psi) : \sum_{\mathfrak{f}_i : \mathrm{val}(\mathfrak{f}_i) \in 2^m} w_i > \sum_{\mathfrak{f}_j : \mathrm{val}(\mathfrak{f}_j) \in 2^{m'}} w_j$$

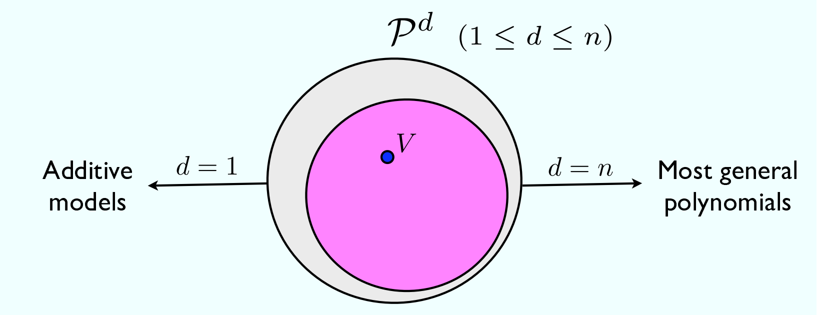
Complexity is Manyfold

1. All constraints in $\mathfrak{C}$ are linear ... in $\mathbb{R}^{4n}$
2. The summations in *each* constraint for a statement $\varphi \succ \psi$ are exponential in $\mathbf{X}_\varphi$ and $\mathbf{X}_\psi$
3. The number of constraints generated for a statement $\varphi \succ \psi$ can be exponential in $\mathbf{X}_\varphi$ and $\mathbf{X}_\psi$ as well
4. Not only generating V, but even storing and evaluating it explicitly might be infeasible.

Complexity Can Be Overcome

Both identifying a valid value function and using it can be done in time linear in $|\mathbf{X}|$ and polynomial in $|S|$

- The computational machinery is based on certain tools from convex optimization and statistical learning
  - Quadratic programming as in Support Vector Machines
  - Mercer kernel functions



Complexity Can Be Overcome

Both identifying a valid value function and using it can be done in time linear in $|\mathbf{X}|$ and polynomial in $|S|$

- The computational machinery is based on certain tools from convex optimization and statistical learning
  - Quadratic programming as in Support Vector Machines
  - Mercer kernel functions
- Selected value function has interesting semantics
- Ability to deal with inconsistent information
- Experimental results show both empirical efficiency and effectiveness

Bibliography

F. Bacchus and A. Grove.
Utility independence in qualitative decision theory.
In *Proceedings of the Fifth Conference on Knowledge Representation (KR–96)*, pages 542–552, Cambridge, 1996. Morgan-Kauffman.

J. Blythe.
Visual exploration and incremental utility elicitation.
In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 526–532, 2002.

R. Brafman, C. Domshlak, and T. Kogan.
Graphically structured value-function compilation.
Artificial Intelligence, 2007.
to appear.

W. W. Cohen, R. E. Schapire, and Y. Singer.
Learning to order things.
Journal of Artificial Intelligence Research, 10:243–270, May 1999.

C. Domshlak and T. Joachims.
Efficient and non-parametric reasoning over user preferences.
User Modeling and User-Adapted Interaction, 17(1-2):41–69, 2007.
Special issue on Statistical and Probabilistic Methods for User Modeling.

R. Herbrich, T. Graepel, and K. Obermayer.
Large margin rank boundaries for ordinal regression.
In *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, Cambridge, MA, 2000.

R. L. Keeney and H. Raiffa.
Decision with Multiple Objectives: Preferences and Value Tradeoffs.
Wiley, 1976.

Bibliography

D. H. Krantz, R. D. Luce, P. Suppes, and A. Tversky.
Foundations of Measurement.
New York: Academic, 1971.

P. La Mura.
Decision-theoretic entropy.
In *Proceedings of the Ninth Conference on Theoretical Aspects of Rationality and Knowledge*, pages 35–44, Bloomington, IN, 2003.

G. Linden, S. Hanks, and N. Lesh.
Interactive assessment of user preference models: The automated travel assistant.
In *Proceedings of the Sixth International Conference on User Modeling*, pages 67–78, 1997.

M. McGeachie and J. Doyle.
Utility functions for ceteris paribus preferences.
Computational Intelligence, 20(2):158–217, 2004.
(Special Issue on Preferences in AI).

Outline

1. Introduction:
   1. Why preferences?
   2. The Meta-Model: Models, Languages, Algorithms

2. Preference Models, Languages, and Algorithms
   1. Total orders and Value Functions
   2. Partial orders and Qualitative Languages
   3. Preference Compilation
   4. Gambles and Utility functions

3. From Preference Specification to Preference Elicitation

Uncertainty

So far: What You Choose is What you Get

All choices were over *(certain) outcomes*

Life isn’t (Always) That Simple

Often, the outcome of our choices is uncertain:

- How long will the new TV function properly?
- We’ll the flight we purchased arrive on-time? arrive.
- When we tell a robot to move in some direction:
  - We don’t know the precise direction it will move in
  - We don’t know how much energy it will consume

Modeling Preferences over Uncertain Outcomes

1. What are we selecting from?

- We choose something (e.g., actions) that leads to some set $O \subset \Omega$ of possible results.
- We are uncertain as to which of these results will transpire.

Modeling Preferences over Uncertain Outcomes

1. What are we selecting from?

- We choose something (e.g., actions) that leads to some set $O \subset \Omega$ of possible results.
- We are uncertain as to which of these results will transpire.

Example 1:

- Item to select: *route to work* (101,280,Foothill Expressway, El-Camino)
- For each route, there are (continuously) many real outcomes that describe: travel-time, gas cost, scenery, etc.

Example 2:

- Item to select: *vacation package*
- Each vacation package can lead to many ”real” vacations that vary in temperature, food quality, facilities, etc.

Modeling Preferences over Uncertain Outcomes

1. What are we selecting from?

- We choose something (e.g., actions) that leads to some set $O \subset \Omega$ of possible results.
- We are uncertain as to which of these results will transpire.

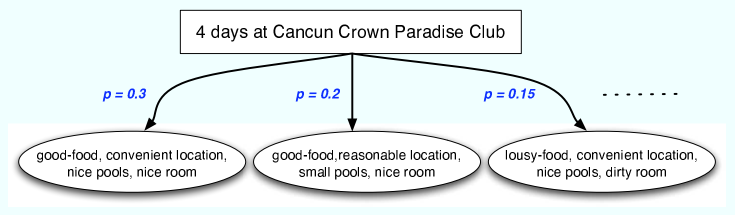
2. How do we capture this uncertainty?

- We model our uncertainty about the precise result using a **probability distribution** over Ω. (Other choices possible.)
- A probability distribution over Ω is called a lottery or a gamble.

Modeling Preferences over Uncertain Outcomes

2. How do we capture this uncertainty?

- We model our uncertainty about the precise result using a **probability distribution** over Ω. (Other choices possible.)
- A probability distribution over Ω is called a lottery or a gamble.

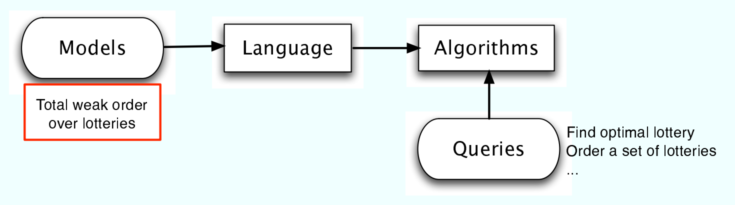

4 days at Cancun Crown Paradise Club
p = 0.3
p = 0.2
p = 0.15
- - - - - - -
good-food, convenient location, nice pools, nice room
good-food,reasonable location, small pools, nice room
lousy-food, convenient location, nice pools, dirty room

Our model = Weak order over lotteries.

Model = Total Weak Order over Lotteries

- Ω – Set of possible *concrete* outcomes
- $\mathcal{L} = \Pi(\Omega)$ – Set of **possible lotteries** over Ω
- $L \subseteq \mathcal{L}$ – Set of **available lotteries** over Ω (e.g., possible actions)
- If $l \in \mathcal{L}$ and $o \in \Omega$, we use $l(o)$ to denote the probability that lottery l will result in outcome o.

Model = Total weak order over L



Specifying Preferences over Lotteries

Difficulties:

Same difficulties as specifying a total-order over outcomes, but compounded:

1. The set of lotteries is potentially uncountably infinite
2. Comparing lotteries is much harder than comparing outcomes

Can we do something?

Structure to the Rescue

The von-Neumann Morgenstern Axioms

Language – Main Result

- Preferences over *lotteries* with certain structure can be described by a utility function over *outcomes*.
- This structure can be captured by means of a number of intuitive properties.

Preliminary Definitions and Assumptions

Assumption 1

$L = \mathcal{L}$

Definition: Complex Lottery

- Let $l_1, \ldots, l_k$ be lotteries.
- Let $a_1, \ldots, a_k$ be positive reals such that $\sum_{i=1}^{k} a_i = 1$
- $l = a_1 l_1 + a_2 l_2 + \ldots + a_k l_k$ is lottery whose ”outcomes” are lotteries themselves.
- l is called a *complex* (as opposed to *simple*) lottery

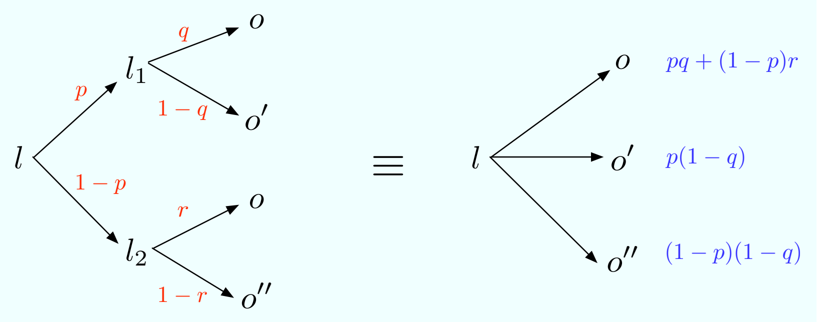
Assumption 2

Every complex lottery is equivalent to a simple lottery

Preliminary Definitions and Assumptions

Assumption 2

Every complex lottery is equivalent to a simple lottery



The von-Neumann Morgenstern Axioms

Axiom 1: $\succeq$ is a Total Weak Order.

For every $l, l' \in \mathcal{L}$ at least one of $l \succeq l'$ or $l' \succeq l$ holds.

Axiom 2: Independence/Substitution

For every lottery p, q, r and every $a \in [0,1]$ if $p \succeq q$ then

$$ap + (1-a)r \succeq aq + (1-a)r$$

Axiom 3: Continuity

If p, q, r are lotteries s.t. $p \succeq q \succeq r$ then $\exists a, b \in [0,1]$ such that

$$ap + (1-a)r \succeq q \succeq bp + (1-b)r$$

The von-Neumann Morgenstern Theorem

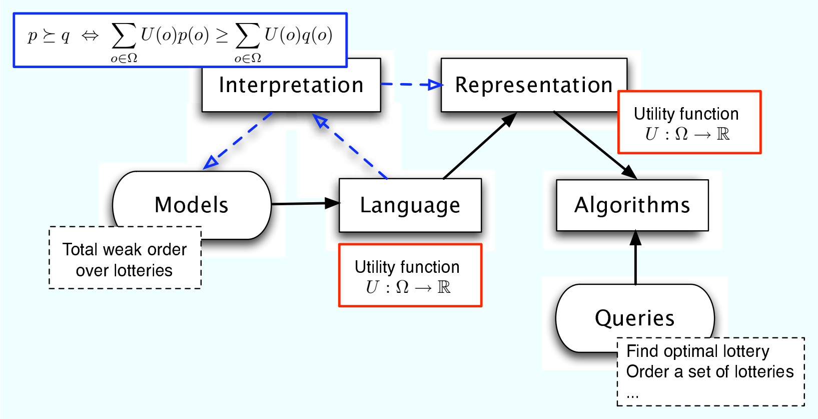
A binary relation over $\mathcal{L}$ satisfies Axioms 1-3 IFF there exists a function $U : \Omega \rightarrow \mathbb{R}$ such that

$$p \succeq q \Leftrightarrow \sum_{o \in \Omega} U(o)p(o) \geq \sum_{o \in \Omega} U(o)q(o).$$

Moreover, U is unique upto affine (= linear) transformations.

Putting Things Together

The von-Neumann Morgenstern Theorem



$$p \succeq q \;\Leftrightarrow\; \sum_{o\in\Omega} U(o)p(o) \geq \sum_{o\in\Omega} U(o)q(o)$$

Interpretation

Representation

Utility function
$U : \Omega \rightarrow \mathbb{R}$

Models

Total weak order
over lotteries

Language

Utility function
$U : \Omega \rightarrow \mathbb{R}$

Algorithms

Queries

Find optimal lottery
Order a set of lotteries
...

Eliciting a Utility Function

1. Order the outcomes in O from best to worst
2. Assign values to best and worst outcome: $U(o_{best}) := 1$ and $U(o_{worst}) := 0$
3. For each outcome $o \in \Omega$:

   a. Ask for $a \in [0, 1]$ such that $o \sim ao_{best} + (1 - a)o_{worst}$

      - What lottery over $\{o_{best}, o_{worst}\}$ is preferentially equivalent to o?

   b. Assign $U(o) := a$

Eliciting a Utility Function

1. Order the outcomes in O from best to worst
2. $U(o_{best}) := 1$ and $U(o_{worst}) := 0$
3. For each outcome $o \in \Omega$:
   a. Ask for $a \in [0,1]$ such that $o \sim ao_{best} + (1-a)o_{worst}$
   b. Assign $U(o) := a$

Example

1. (unspicy, healthy) $\succeq$ (spicy,junk-food) $\succeq$ (spicy,healthy) $\succeq$ (unspicy, junk-food)
2. U(unspicy,healty) := 1; U(unspicy, junk-food) := 0;
   a. Ask for p and q such that

$$(spicy, healthy) \sim p(unspicy, healthy) + (1-p)(unspicy, junk\text{-}food)$$
$$(spicy, junk - food) \sim q(unspicy, healthy) + (1-q)(unspicy, junk - food)$$

   b. U(spicy,healthy) := p; U(spicy,junk-food) := q

Research Issues: Representation and Independence

Representation

Suppose $\Omega = \mathcal{X}$ for some attribute set **X**.
Under what assumptions does U have a simple form?
Simpler form: *sum or product of smaller factors*

Independence

What is the relationship between various utility independence properties and the form of U?

Elicitation

- How can we identify independence properties?
- If U satisfies various independence properties/structure, how can we formulate simple questions that allow us to construct U quickly?
- What information do we need to make a concrete decision?

Bibliography

F. Bacchus and A. Grove.
Utility independence in qualitative decision theory.
In *Proceedings of the Fifth Conference on Knowledge Representation (KR–96)*, pages 542–552, Cambridge, 1996. Morgan-Kauffman.

C. Goutis.
A graphical model for solving a decision analysis problem.
IEEE Trans. Systems, Man and Cybernetics, 26(8):1181–1193, 1995.

R. A. Howard and J. E. Matheson.
Influence diagrams.
The Principles and Applications of Decision Analysis, 2:719–762, 1984.

R. C. Jeffrey.
The Logic of Decision.
University of Chicago Press, 1983.

P. Korhonen, A. Lewandowski, and J. Wallenius (eds.).
Multiple Criteria Decision Support.
Springer-Verlag, Berlin, 1991.

P. La Mura and Y. Shoham.
Expected utility networks.
In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 367–373, Stockholm, Sweden, 1999. Morgan Kaufmann Publishers.

L. Savage.
The Foundations of Statistics.
Dover, 2 edition, 1972.

Bibliography

R. D. Shachter.

Evaluating influence diagrams.
In G.Shafer and J.Pearl, editors, *Reading in Uncertaint Reasoning*, pages 79–90. Morgan Kaufmann, 1990.

Y. Shoham.

Conditional utility, utility independence, and utility networks.
In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 429–436, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

J. von Neumann and O. Morgenstern.

Theory of Games and Economic Behavior.
Princeton University Press, 2 edition, 1947.

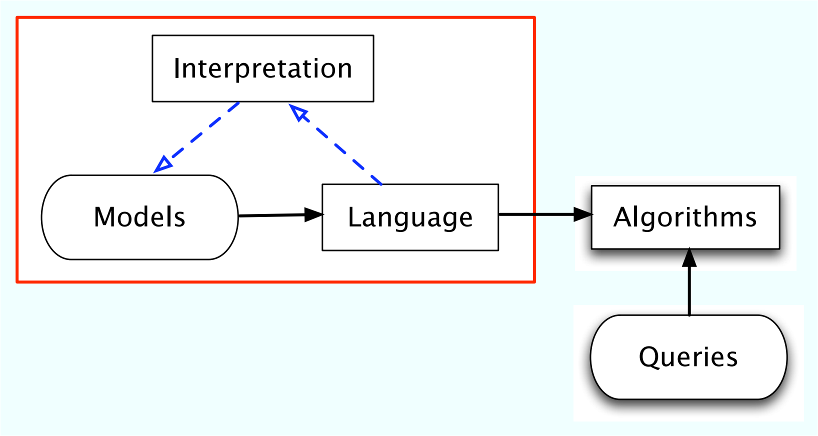
Outline

1. Introduction:
    1. Why preferences?
    2. The Meta-Model: Models, Languages, Algorithms

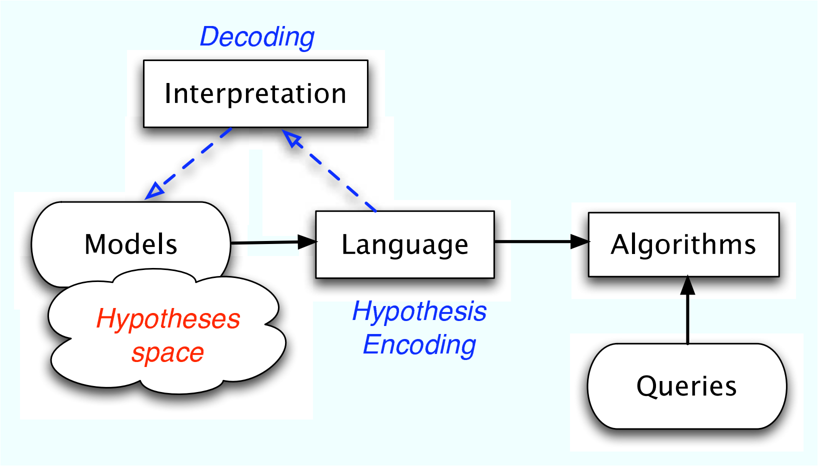
2. Preference Models, Languages, and Algorithms
    1. Total orders and Value Functions
    2. Partial orders and Qualitative Languages
    3. Preference Compilation
    4. Gambles and Utility functions

3. From Preference Specification to Preference Elicitation

A Closer Look at Preference Specification

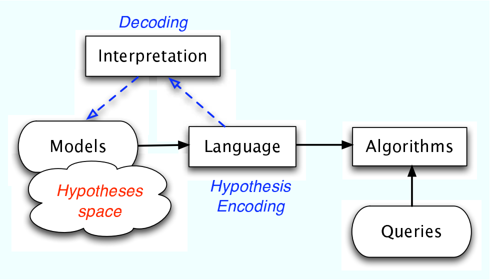


A Closer Look at Preference Specification



Hypotheses Space

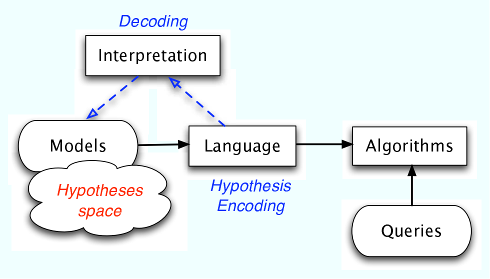
Generalizing perspective



The space of possible preference models constitute an hypotheses space (HS) of the system

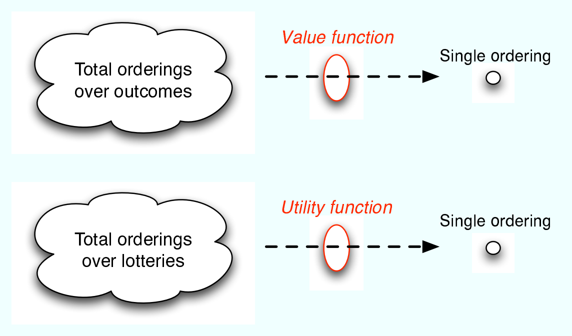
- Space of total/partial orders
- Space of value functions
- Space of utility functions

Information Encoding and Decoding



Encoding	User provides information aiming at reducing HS towards her own model
Decoding	System aims at “understanding” the user as well as possible

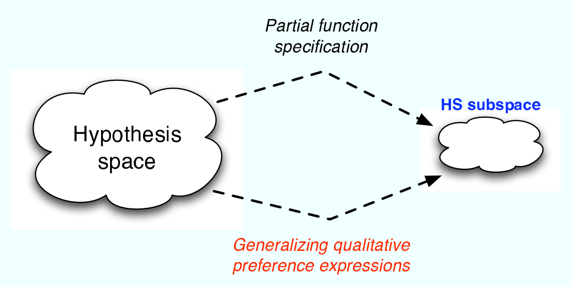
Easy Cases



Complete Value/Utility Specification

- Decoding is redundant ⇒ specified function restricts HS to a single model
- No ambiguity

Complicated Cases


Partial function specification
Hypothesis space
HS subspace
Generalizing qualitative preference expressions

Partial Specification

- User's information leaves us with a subspace of HS
- Hmm ... how should we proceed next?

Reasoning about Partial Preference Specification

What should we do when left with an HS subspace?

Assume Probability Distribution over HS

1. Maximum likelihood inference
    - Start with a prior probability distribution over space of models
    - Update distribution given user statements
    - Find the most likely model
    - Answer queries using this model

Reasoning about Partial Preference Specification

What should we do when left with an HS subspace?

Assume Probability Distribution over HS

1. Maximum likelihood inference
   - Start with a prior probability distribution over space of models
   - Update distribution given user statements
   - Find the most likely model
   - Answer queries using this model

2. Bayesian inference
   - Start with a prior probability distribution over space of models
   - Update distribution given user statements
   - Answer queries by considering all models, weighted by their probability

Max-Likelihood Inference

Assume Probability Distribution over HS

CP-nets

Peaked probability distribution over partial orderings

$$p(\succ) \sim \begin{cases} 1, & \succ \text{ assumes all and only all the information in } N \\ 0, & \text{otherwise} \end{cases}$$

Max-Likelihood Inference

Assume Probability Distribution over HS

CP-nets

Peaked probability distribution over partial orderings

$$p(\succ) \sim \begin{cases} 1, & \succ \text{ assumes all and only all the information in } N \\ 0, & \text{otherwise} \end{cases}$$

Structured Value-function Compilation

Probability distribution over polynomial value functions

$$p(V) \sim \begin{cases} 1, & p'(V) \\ 0, & V \text{ violates structural assumptions} \end{cases}$$

Max-Likelihood Inference

Assume Probability Distribution over HS

CP-nets

Peaked probability distribution over partial orderings

$$p(\succ) \sim \begin{cases} 1, & \succ \text{ assumes all and only all the information in } N \\ 0, & \text{otherwise} \end{cases}$$

Structured Value-function Compilation

Probability distribution over polynomial value functions

$$p(V) \sim \begin{cases} 1, & p'(V) \\ 0, & V \text{ violates structural assumptions} \end{cases}$$

Structure-less Value-function Compilation

Probability distribution over polynomial value functions

$$p(V) \sim -e^{||w_V||^2}$$

Bayesian Reasoning

Assume Probability Distribution over HS

Expected Expected Utility

Probability distribution over utility functions

$$p \succeq q \;\Leftrightarrow\; \sum_{o \in \Omega} U(o)p(o) \geq \sum_{o \in \Omega} U(o)q(o).$$

is replaced with

$$p \succeq q \;\Leftrightarrow\; \sum_{U} p(U) \sum_{o \in \Omega} U(o)p(o) \geq \sum_{U} p(U) \sum_{o \in \Omega} U(o)q(o).$$

Reasoning about Partial Preference Specification

What should we do when left with an HS subspace?

Assume Probability Distribution over HS

1. Max-likelihood inference
2. Bayesian inference

No Reasonable Probability Distribution over HS

1. Act to minimize maximal regret
2. *Other suggestions?*

Minimizing Maximal Regret

No Reasonable Probability Distribution over HS

Concept of Regret

How bad can my decision be in comparison to the best decision

Pairwise Regret

- If the user's *true* utility function is u but I select u'
- Then I'll get the best item, o', according to u' instead of the best item, o, according to u
- The *user's regret* would be: $u(o) - u(o')$

Minimizing Maximal Regret

No Reasonable Probability Distribution over HS

Maximal Regret

- Given a set $\mathcal{U}$ of candidate utility functions
- If I select $u' \in \mathcal{U}$ as the user's utility function, then the user's maximal regret will be:

$$\text{Regret}(u'|\mathcal{U}) = \max_{u \in \mathcal{U}} [u(o^*_u) - u(o^*_{u'})]$$

where o^*_u is the best outcome according to u

Minimizing Max Regret

Given a set of candidate utility function $\mathcal{U}$, select the utility function u such that $Regret(u|\mathcal{U})$ is minimal

From Preference Specification to Preference Elicitation

So far: Preference Specification

Offline, *user-selected* pieces of information about her preferences

Pros	- User should know better what matters to him
Cons	- “Should know” does not mean “comprehend”, surely does not mean “will express” - User knows worse the feasibility of different outcomes (e.g., the catalog of Amazon.com)

From Preference Specification to Preference Elicitation

So far: Preference Specification

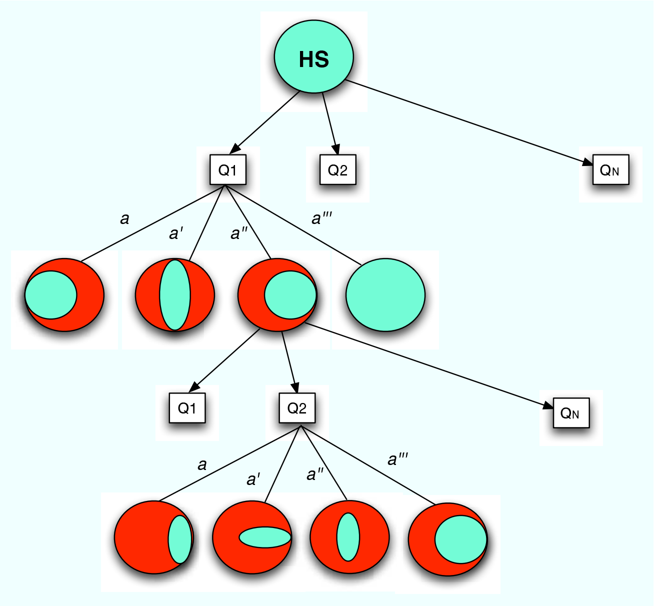
Offline, *user-selected* pieces of information about her preferences

Pros	- User should know better what matters to him
Cons	- "Should know" does not mean "comprehend", surely does not mean "will express" - User knows worse the feasibility of different outcomes (e.g., the catalog of Amazon.com)

Alternative: Preference Elicitation

1. Online, *system-selected questions* about user preferences
2. User's *answers* constitute the elicited pieces of information about her preferences
3. Questions can be asked (and thus *selected*) sequentially

Sequential HS Reduction



Example: K-Items Queries

Task: Given a set of outcomes, home-in on the most-preferred one

Interface/Protocol

- While user is not tired, loop
  1. System presents the user with a list of K alternative outcomes
  2. User selects the most preferred outcome from the list
- Select a non-dominated outcome

Example: K-Items Queries

Task: Given a set of outcomes, home-in on the most-preferred one

Interface/Protocol

- While user is not tired, loop
  1. System presents the user with a list of K alternative outcomes
  2. User selects the most preferred outcome from the list
- Select a non-dominated outcome

HS Reduction: Simple, yet inefficient

HS	Total strict orderings
Queries	Different sets of K outcomes
Answers	K alternative answers per query
Effect on HS	Elimination of orderings inconsistent with K pairwise relations implied by the answer
Issues	Slow progress, Vague principles for query selection

Example: K-Items Queries

Task: Given a set of outcomes, home-in on the most-preferred one

Interface/Protocol

- While user is not tired, loop
  1. System presents the user with a list of K alternative outcomes
  2. User selects the most preferred outcome from the list
- Select a non-dominated outcome

HS Reduction: Structured Value-Function Compilation

HS	Certain class of value functions over attributes **X**
Queries	Different sets of K outcomes
Answers	K alternative answers per query
Effect on HS	Elimination of value functions inconsistent with K pairwise relations implied by the answer
Issues	Progress is faster due to generalization

Example: *K*-Items Queries

Task: Given a set of outcomes, home-in on the most-preferred one

Interface/Protocol

- While user is not tired, loop
  1. System presents the user with a list of *K* alternative outcomes
  2. User selects the most preferred outcome from the list
- Select a non-dominated outcome

Research Questions

1. How should we measure *query informativeness*?
2. When can we efficiently compute the informativeness of a query?
3. When can we efficiently select the *most informative query*?
4. Use “most informative” query, or a top-*K* set of most likely candidates for the optimal outcome? (User gets tired ...)

Example: Decision-oriented Utility Elicitation

Task: Given a set of lotteries, home-in on a most-preferred one

Interface/Protocol

- Assume
  - Probability distribution $p(U)$ over utility functions
  - Fixed set of possible queries
    Example: Ask for $p \in [0,1]$ such that $o \sim po' + (1-p)o''$
- While user is not tired, loop
  1. Ask query with the highest myopic/sequential value of information
  2. Given user's answer, update $p(U)$
- Select the lottery with the highest expected expected utility

Bibliography

C. Boutilier.
A POMDP formulation of preference elicitation problems.
In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, pages 239–246, 2002.

C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans.
Regret-based utility elicitation in constraint-based decision problems.
In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005.

U. Chajewska, L. Getoor, J. Norman, and Y. Shahar.
Utility elicitation as a classification problem.
In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 79–88, San Francisco, CA, 1998. Morgan Kaufmann Publishers.

U. Chajewska, D. Koller, and R. Parr.
Making rational decisions using adaptive utility elicitation.
In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 363–369, 2000.

B. Faltings, M. Torrens, and P. Pu.
Solution generation with qualitative models of preferences.
International Journal of Computational Intelligence and Applications, 7(2):246–264, 2004.

V. Ha and P. Haddawy.
Problem-focused incremental elicitation of multi-attribute utility models.
In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 215–222, Providence, Rhode Island, 1997. Morgan Kaufmann.

V. Ha and P. Haddawy.
A hybrid approach to reasoning with partially elicited preference models.
In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, Stockholm, Sweden, July 1999. Morgan Kaufmann.

Bibliography

J. Payne, J. Bettman, and E. Johnson.
The Adaptive Decision Maker.
Cambridge University Press, 1993.

P. Pu and B. Faltings.
Decision tradeoff using example critiquing and constraint programming.
Constraints, 9(4):289–310, 2004.

B. Smith and L. McGinty.
The power of suggestion.
In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 127–132, 2003.

M. Torrens, B. Faltings, and P. Pu.
SmartClients: Constraint satisfaction as a paradigm for scaleable intelligent information systems.
Constraints, 7:49–69, 2002.

A. Tversky.
Elimination by aspects: A theory of choice.
Psychological Review, 79:281–299, 1972.

Summary

1. Introduction:
    1. Why preferences?
    2. The Meta-Model: Models, Languages, Algorithms
2. Preference Models, Languages, and Algorithms
    1. Total orders and Value Functions
    2. Partial orders and Qualitative Languages
    3. Preference Compilation
    4. Gambles and Utility functions
3. From Preference Specification to Preference Elicitation