Conceptual spaces for matching and representing preferences

Anton Benz
Alexandra Strekalova
ZAS Berlin

Tandem Workshop on Optimality in Language and Geometric Approaches to Cognition

13 December 2010

The KomParse project

- **Overall goal**: Develop Non-Player Characters (NPCs) with natural language dialogue capabilities.
- **Our scenario**: furniture sales agent

- Funded by Investitionsbank Berlin (IBB) by the ProFIT Programme.
- Partners: German Research Centre for Artificial Intelligence (DFKI) and Centre for General Linguistics (ZAS).
- In cooperation with *Random Labs* and *Metaversum GmbH*.

That's how it looks

Modeling a dialogue situation and an NPC response

- User gives some preferences about a furniture object that he would like to have.
- NPC has to respond by:
  - showing an object that fulfills these preferences, if he can find one.
  - suggesting alternative object properties, if the database does not contain such an object.

User: I would like to have a *purple leather* sofa.
Agent: I'm afraid we don't have a purple leather sofa, but I can show you a *purple fabric* sofa or *black leather* one.

Preference modelling: Deontic Logic

I would like to have a **purple leather sofa**.

- Modal logic:

$$\mathbf{D}\ \exists x(have(I, x) \wedge sofa(x) \wedge purple(x) \wedge leather(x))$$

Ross's paradox:

- I want that the letter is mailed.
- I want that the letter is mailed or burned.

$$\mathbf{D}\varphi \Rightarrow \mathbf{D}(\varphi \vee \psi).$$

Alternative: Multi–Attribute Utility Analysis

I would like to have a **purple leather sofa**.

- Representation as Constraints:
  - $C_1 = <$color, purple$>$ soft
  - $C_2 = <$material, leather$>$ soft
  - $C_2 = <$ObjType, sofa$>$ hard
- Decompose utility function of customer:
  - F: global utility function over objects of given type;
  - $F_{\textbf{colour}}$: preference over colours;
  - $F_{\textbf{material}}$: preference over materials;

$$F(o) = F_{\textbf{colour}}(o) + F_{\textbf{material}}(o),\ o \in \text{ObjType}.$$

Represent Preferences in Cost Network

I would like to have a **purple leather sofa**.

- (X, D, C, F): Cost network
- X $=$ {object, color, material, style}
- D_{color} $=$ {Auburn, Chocolate, Mahogany,...}
- $D_{material}$ $=$ {fabric, leather, plastic,...}
- o: objects $=$ instantiation of variables
- C $=$ <ObjType, sofa>: hard constraint
- $F_{\mathbf{global}} = \alpha_{\mathbf{colour}} F_{\mathbf{colour}} + \alpha_{\mathbf{material}} F_{\mathbf{material}}$

Constraint optimization

- Task: To find an optimal suggestion by minimizing the global cost function:

$$min_o F(o) = min_o \sum_{i=1}^{n} \alpha_i F_i(o). \quad (1)$$

- Problem: Values for the weights α_i and functions F_i are unknown!
- Expressed preferences only set the goal.
  - Functions F_i can be constrained only very broadly;
  - Weights $\alpha_i > 0$ can have arbitrary values.
- Approach: Use natural similarity measure on the domains (*Conceptual Spaces*) to constrain F_i.

Conceptual spaces

I would like to have a **purple leather sofa**.

- **Purple leather sofa** defines a point in a conceptual space
- This conceptual space is a product of color and material spaces
  - Color space is defined by HSV color model (hue, saturation, value)
  - Material space is defined by material properties (organic, robust, rough, ...)
- Problem: the color space is too fine grained
- Solution: define equivalence classes of properties which have a similar distance from the desired goal property.

Equivalence classes

- Divide all property values into n equivalence classes according to the distance to the desired property value.
- How can we assign equivalence classes?

Equivalence classes

- Divide all property values into *n* equivalence classes according to the distance to the desired property value.
- How can we assign equivalence classes?

Color = (hue, saturation, value)

- Compute distance between the desired color and the color of the current object.
- Compare the value with a threshold and assign an equivalence class.

Equivalence classes

- Divide all property values into n equivalence classes according to the distance to the desired property value.
- How can we assign equivalence classes?

Color = (hue, saturation, value)

- Compute distance between the desired color and the color of the current object.
- Compare the value with a threshold and assign an equivalence class.

Material = (organic, robust, rough, ...)

- Count the number of overlapping boolean values for material properties for the desired material and the material of the current object.
- Compare the number with a threshold and assign an equivalence class.

Theorem

Let $(E_i)_{i=1}^n$ be a sequence of sets of natural numbers, and $E = \prod_1^n E_i$. Let $e = (e_i)_{i=1}^n \in E$. Then the following conditions are equivalent:

1. *There are weights α_i and functions $F_i : E_i \to \mathbb{R}_0^+$, $i = 1 \ldots, n$, such that*
   i. $\forall i : \alpha_i > 0$,
   ii. $\forall n, m \in E_i : n < m \to F_i(n) < F_i(m)$,
   iii. *and*

$$F(e) = \min_{e=(e_i)_{i=1,\ldots,n}} \sum_{i=1}^{n} \alpha_i F_i(e_i)$$

2. *e is an element of the set*

$$K = \{e \in E | \forall e' \in E : \exists i \; e'_i < e_i \to \exists j : e_j < e'_j\}.$$

Candidate set

- Determine the candidate set K (vectors of equivalence classes), such that for each $e \in K$ there are weights $(\alpha_i)_{i=1,\ldots,n}$ and functions $(F_i)_{i=1,\ldots,n}$ and for which holds:

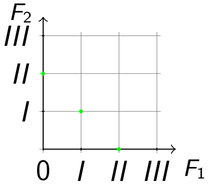
$$F(e) = \min_{e=(e_i)_{i=1,\ldots,n}} \sum_{i=1}^{n} \alpha_i F_i(e_i) \qquad (2)$$

Candidate set

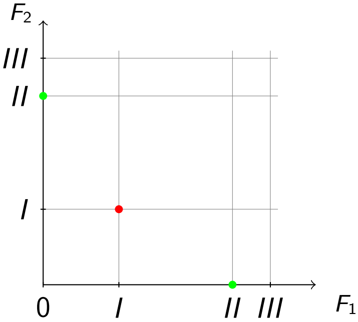
- Determine the candidate set K (vectors of equivalence classes), such that for each $e \in K$ there are weights $(\alpha_i)_{i=1,\ldots,n}$ and functions $(F_i)_{i=1,\ldots,n}$ and for which holds:

$$F(e) = \min_{e=(e_i)_{i=1,\ldots,n}} \sum_{i=1}^{n} \alpha_i F_i(e_i) \qquad (2)$$

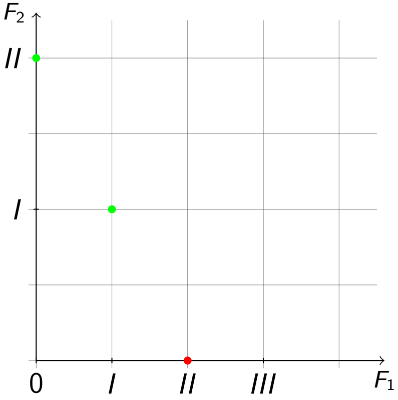
- Geometric representation



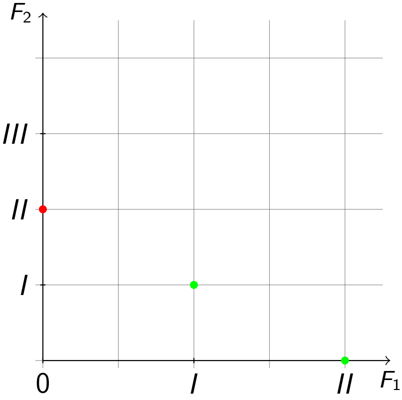
Constraining F_i functions



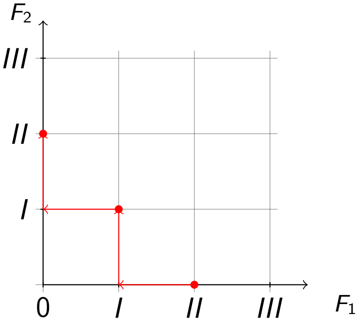
Weights: $\alpha_2 = 2\alpha_1$



Weights: $\alpha_1 = 2\alpha_2$



Candidate set



Example scenario

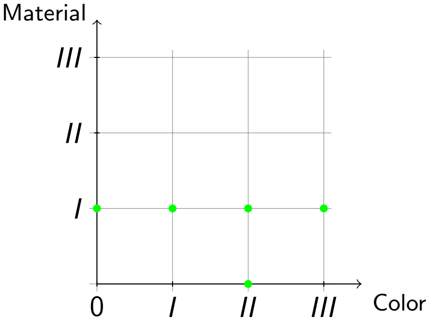
- **User**: I would like to have a *purple leather sofa*.

$$\begin{bmatrix} \textsc{color} & \text{purple} \\ \textsc{material} & \text{leather} \end{bmatrix}$$

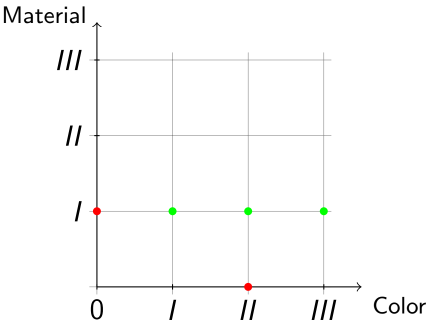
Mapping ontology objects on equivalence classes

Object	Properties	Equivalence classes
Sofa_Alatea	[COLOR red, MATERIAL fabric]	[COLOR II, MATERIAL I]
Sofa_Anni	[COLOR blue, MATERIAL fabric]	[COLOR I, MATERIAL I]
Sofa_Consuelo	[COLOR yellow, MATERIAL fabric]	[COLOR III, MATERIAL I]
Sofa_Grace	[COLOR blue, MATERIAL fabric]	[COLOR I, MATERIAL I]
Sofa_Nadia	[COLOR black, MATERIAL leather]	[COLOR II, MATERIAL 0]
Sofa_Isadora	[COLOR purple, MATERIAL fabric]	[COLOR 0, MATERIAL I]

Search of n-best candidates



Search of n-best candidates



Output: response generation

Equivalence classes	Object	Property classes
COLOR 0 MATERIAL I	Sofa_Isadora	COLOR purple MATERIAL fabric
COLOR II MATERIAL 0	Sofa_Nadia	COLOR black MATERIAL leather

User: I would like to have a *purple leather* sofa.
Agent: I'm afraid we don't have a purple leather sofa, but I can show you a *purple fabric* sofa or *black leather* one.

Summary

- Task:
  - Find alternative values for preferences expressed by user
  - Generate an adequate answer
- Approach:
  - Represent preferences as hard and soft constraints
  - Minimize functions for value parameters in a cost network
  - Exploit the geometric structure of property spaces
  - Generate an answer which offers the closest satisfiable property combination