DD-PREF: A Language for Expressing Preferences Over Sets

Problem Statement:

Focus: Modeling preferences over sets of items.
Why set-based methods?
Ranking items independently cannot capture inter-object interactions.

Example: selecting items for a meal:

Sub-additive utility (redundancy, incompatibility)
Super-additive utility (complementarity)

Need:
1. A language to express preferred (and non-preferred) relationships between items in a set
2. A method to select sets that satisfy the preferences
3. A method to infer preferences from user selections

Blockworld: Randomly generated blocks with four features: size, color, number of sides, and location (bin number).

- Task 1: Create a mosaic.
  \[ P = \{ 0.25, 0.3, 0.5 \} \]
  \[ Q = \{ 0.25, 0.5, 0.5 \} \]
  \[ W = \{ 0.5, 0.5, 0.5 \} \]
  \[ P_{1} = \{ 0.21, 0.21, 0.58 \} \]
  \[ P_{2} = \{ 0.21, 0.21, 0.58 \} \]
- Task 2: Build a uniform tower.
  \[ P_{1} = \{ 0.25, 0.25, 0.5 \} \]
  \[ P_{2} = \{ 0.25, 0.25, 0.5 \} \]
  \[ W = \{ 0.5, 0.5, 0.5 \} \]
- Task 3: Select blocks for a child.
  \[ Q = \{ 0.25, 0.25, 0.5 \} \]
  \[ P_{1} = \{ 0.21, 0.21, 0.58 \} \]
  \[ P_{2} = \{ 0.21, 0.21, 0.58 \} \]

Mars Rover images: Collected during a field test on Earth and represented by six features: percent of the image classified as sky, rock, rock layers, light soil, dark soil, and shadow.

Users select top five of 25; the system then infers their preferences and applies them to a larger set of 100 to select their optimal top 20.

One user’s top five images and the resulting derived preferences:

- Task 1: Create a mosaic.
  \[ P_{1} = \{ 0.25, 0.3, 0.5 \} \]
  \[ Q = \{ 0.25, 0.5, 0.5 \} \]
  \[ W = \{ 0.5, 0.5, 0.5 \} \]
  \[ P_{1} = \{ 0.21, 0.21, 0.58 \} \]
  \[ P_{2} = \{ 0.21, 0.21, 0.58 \} \]

Data Sets

Objective function (minimize):
\[ \sum_{i=1}^{n} w_{i} \sum_{j=1}^{m} f_{ij} \times d_{ij} \]
for subset \( S \), preferences \( P \), and diversity weight \( \alpha \)

Algorithm: Identifying the Best Subset (Wrapper-Greedy)

Given preferences \( P \), a universe \( \mathcal{U} \) of objects, a “seed” object \( v \), and a diversity weight \( \alpha \), select \( k \) objects as a set.

Basic-Greedy(P, \( \mathcal{U} \), \( v \), \( \alpha \), \( k \))
1. Initialize candidate set \( S \) with seed object \( v \).
2. For \( j \) from 1 to \( k \):
   \( a) \) Select the object \( x \in \mathcal{U} - S \) that maximises
   \[ \sum_{i=1}^{n} w_{i} \sum_{j=1}^{m} f_{ij} \times d_{ij} \]
   \( b) \) Set \( S = S \cup \{ x \} \).
3. Return \( S \).

Wrapper-Greedy: Iterate over all possible seed objects and select the best result (by objective function value).

Experimental Questions

1. Can we express qualitatively different preferences?
2. Given preferences and a set of items, can we efficiently select a subset to satisfy the preferences?
3. Given user selections, can we capture (learn) their implicit preferences?

Results

<table>
<thead>
<tr>
<th>Preferences</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximal</td>
<td>0.21</td>
<td>0.21</td>
<td>0.21</td>
</tr>
<tr>
<td>Exhaustive</td>
<td>0.21</td>
<td>0.21</td>
<td>0.21</td>
</tr>
<tr>
<td>Rank</td>
<td>0.21</td>
<td>0.21</td>
<td>0.21</td>
</tr>
</tbody>
</table>

Conclusions

1. User preferences are necessary for encoding different task goals and individual desires.
2. Feature-based preference statements can capture relevant preferences.
3. Both diversity and depth are important for finding the best subset.

Future Work
- Investigate the use of OP-Nets to code dependencies between features.
- Apply preferences to a large music data base, to generate DJ playlists.
- Learn preferences automatically from observing user behavior.