Effective Query Selection during Preference Elicitation

Minruo Li[1], Michael Fleming[1], Scott Buffett[2]

1. Faculty of Computer Science, University of New Brunswick

2. National Research Council of Canada

Objective

One of the problems in conducting automated negotiation is that of maximizing the utility of the user being represented. However, the agent conducting the negotiation typically will not have prior knowledge of the user's true utility for every outcome. This project tries to find an efficient and accurate way for the agent to estimate the user's utility for each outcome and use this estimated utility when making decisions during the negotiation process.

Background – COP-nets

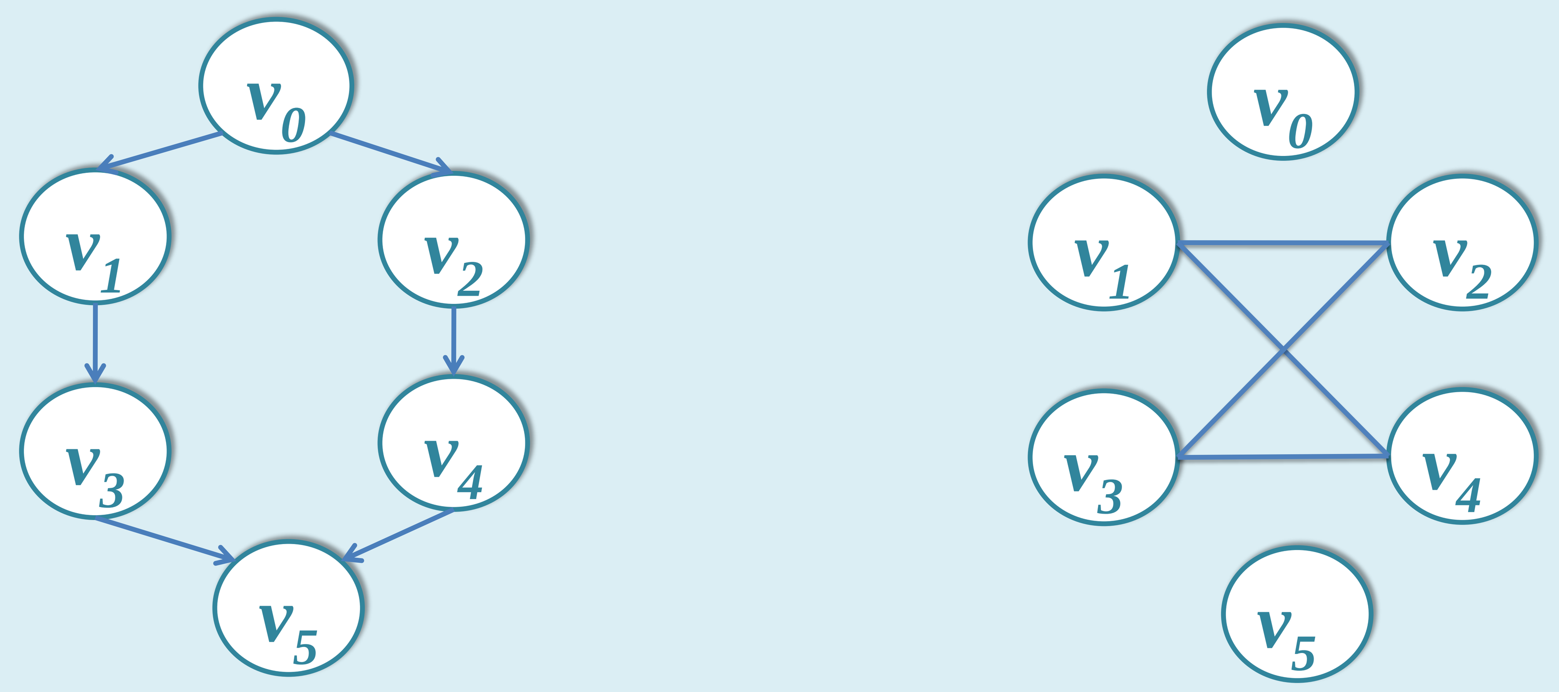
A Conditional Outcome Preference Network (COP-net) is a graphical model used to represent a user's preferences over a set of outcomes [1].

- A COP-net is a directed acyclic graph consisting of a set of nodes and a set of directed edges.
- Each node denotes an outcome and each directed edge represents a preference.
- COP-nets are transitively reduced graphs.
- A COP-net has a small number of nodes with a prior labeling of known utilities for the user, which are known as true utilities and will then be used, along with the known preferences, to estimate the utilities of the rest of the nodes for the user.

Query Selection with COP-nets

Our goal is to find some currently *unknown* preferences that would be the most useful to the agent. This can be modeled with the COP-net as well. The procedure for finding all unknown preferences using a COP-net is summarized as follows:

- Find the transitive closure C_{TC} of a COP-net
- "Undirect" the graph, by converting arcs to undirected edges
- Find the complement C'_{TC}, which is referred to as the query graph
- Each edge in the resulting graph represents an unknown preference or a query



Methodology

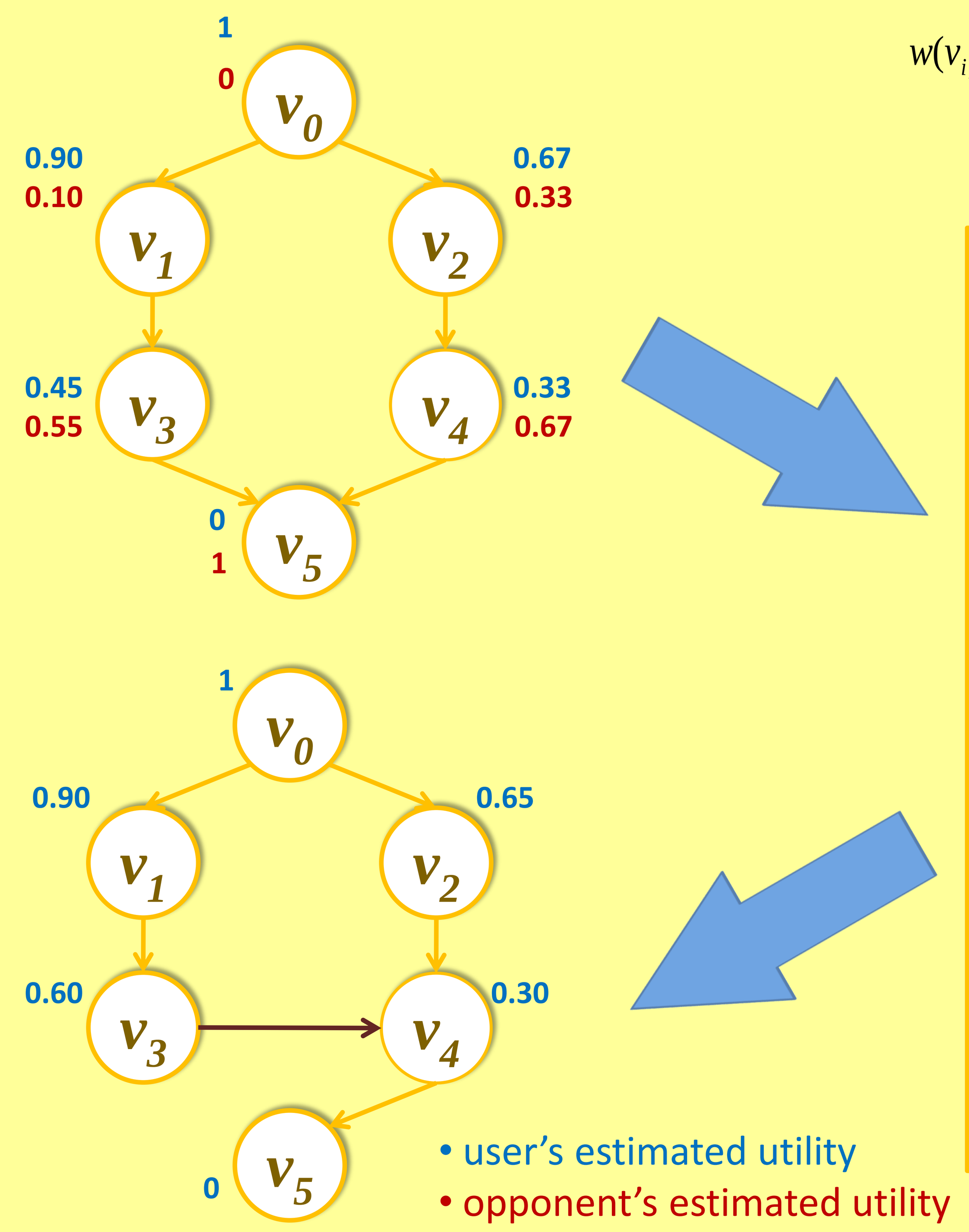
1. Construct the COP-net
2. Estimate user's utility of each outcome using an existing method (longest path method)
3. Generate query graph and find all possible queries (unknown preferences)
4. **Weight each query**
5. Ask one or more queries with highest weight
6. Re-construct the COP-net and re-estimate user's utility of each outcome
7. **Simulate the negotiation process**

Weighting Scheme

We would like to ask queries that reveal a high number of unknown preferences and that involve outcomes that are more likely to be interesting for both the user and the opponent. In one proposed method, the steps for calculating the weight for each edge in the query graph are described as follows:

- Let o_i be the outcome represented by v_i in the query graph
- Let (v_i, v_j) be an edge in the query graph
- Let $P_{i>j}$ be the set of preferences that would be learned if the user specifies $o_i > o_j$, including $o_i > o_j$
- Let $E_{i>j}$ represent the set of edges that would be removed from the query graph as a result
- Let $u(o_i)$ and $u_{opp}(o_i)$ be the agent's estimates of the user's and the opponent's utilities for o_i
- Compute weight of (v_i, v_j) by the formula:

$$w(v_{i,} v_j) = \min \left\{ \begin{array}{l} \sum_{(v_k, v_l) \in E_{i>j}} (u(v_k) \times u_{opp}(v_k) \times u(v_l) \times u_{opp}(v_l)), \\ \sum_{(v_k, v_l) \in E_{j>i}} (u(v_k) \times u_{opp}(v_k) \times u(v_l) \times u_{opp}(v_l)), \end{array} \right\}$$



Weight(v_1, v_2) = min{0.90*0.10*0.67*0.33
+ 0.90*0.10*0.33*0.67,
0.67*0.33*0.90*0.10
+ 0.67*0.33*0.45*0.55}
= 0.0398

Weight(v_1, v_4) = min{0.90*0.10*0.33*0.67,
0.33*0.67*0.90*0.10
+ 0.33*0.67*0.45*0.55}
= 0.0199

Weight(v_3, v_4) = min{0.45*0.55*0.33*0.67
+ 0.90*0.10*0.33*0.67,
0.33*0.67*0.45*0.55
+ 0.67*0.33*0.45*0.55}
= **0.0746**

Weight(v_2, v_3) = min{0.67*0.33*0.45*0.55,
0.45*0.55*0.67*0.33
+ 0.45*0.55*0.33*0.67}
= 0.0547

Testing – Negotiation process

Both the user and the opponent will give an offer that maximizes their own utilities and accept an offer only when their utility for the offer reaches some acceptance point.

Outcomes	User's True Utility	User's Estimated Utility		Opponent's True Utility
		Without asking queries	After asking queries	
v_0	1	1	1	0
v_1	0.85	0.90	0.80	0.20
v_2	0.50	0.60	0.55	0.40
v_3	0.30	0.50	0.25	0.70
v_4	0.20	0.40	0.15	0.85
v_5	0	0	0	1

(Accepting point = 0.40)

Negotiation without asking queries:
- The user offers v_0
- The opponent rejects and offers v_5
- The user rejects and offers v_1
- The opponents rejects and offers v_4
- The user accepts v_4

Final utility achieved: 0.20

Negotiation with asking queries:
- The user offers v_0
- The opponent rejects and offers v_5
- The user rejects and offers v_1
- The opponents rejects and offers v_4
- The user rejects and offers v_2
- The opponent accepts v_2

Final utility achieved: 0.50

The negotiation process will be conducted with different weighting schemes. At the end of each negotiation process, we will measure the user's true utility of the accepted offer and compare the results obtained from the use of several different query weighting methods.

References

[1] S. Chen, S. Buffett, and M. W. Fleming. Reasoning with Conditional Preferences across Attributes. 2007. The 20th Canadian Conference on Artificial Intelligence May 28, 2007.

Faculty of Computer Science UNB

National Research Council Canada Conseil national de recherches Canada