Within Task Preference Elicitation in Net Benefit Planning

Alan Lindsay, Bart Craenen, Ronald P. A. Petrick
Automated Planning Lab,
Heriot-Watt University, Edinburgh, Scotland, UK
{alan.lindsay,b.craenen,r.petrick}@hw.ac.uk

Abstract

In order that an agent can be an effective collaborator it is important that the agent is able to adapt its behaviour for the preferences of a particular user. User preference elicitation has been considered as a process that happens prior to plan execution and typically prior to the planning process. However, when entering an interaction with a new human user it will not always be possible or desirable for an elicitation episode to take place. Moreover, the cost of any elicitation (e.g., annoyance) must be weighed against its benefit in distinguishing between alternative plans. We therefore pose the problem of *within task preference elicitation*, which explicitly represents the agent's knowledge about the user's preference model and how the agent's knowledge can develop as the interaction progresses. Our approach parameterises a utility model for a net benefit planning task with a set of (observable) user attributes. This set of user attributes are represented as unknown values in a partially observable planning model and can be accessed through guarded sensing actions (e.g., through asking a question when it becomes relevant), allowing the planner to reason with the possible alternative user utility models. In this work we define the within task preference elicitation problem and present our framework for solving these problems. We present results examining its use in modified benchmark scenarios, including a new planning domain based on a tour guide scenario.

Introduction

Autonomous and intelligent social robots are becoming more common and are being used in an increasing number of roles, including physically assistive robots (Canal, Alenyà, and Torras 2017) and interactive collaborative robots (Kragic et al. 2018). In order that an agent can be an effective collaborator it is important that the agent is able to adapt its behaviour for the preferences of a particular user. In certain circumstances it is possible to observe each individual user over long periods of time in order that their preferences can be learned, e.g., (Woodworth et al. 2018). However, in many roles, such as a tour guide, or an office gopher, the agent may only interact with each individual a small number of times. In these cases it is still desirable that the agent can customise its behaviour, but the user's preferences must be elicited by the agent as part of the interaction.

The elicitation of user preferences has been considered as a process that occurs prior to plan execution and typically prior to the planning process. The selection of elicitation questions is typically targeted towards optimising the accuracy of the resulting preference model, e.g., using a minimax regret decision criterion (Boutilier et al. 2006). As these approaches can result in large sets of questions, research has also been done to look at incorporating user annoyance into the model of elicitation (Gucsi et al. 2020). Starting from an existing preference model, various approaches incorporate the user's preferences within the selection of an appropriate solution. For example, user preferences can be captured as utility functions in net benefit planning problems (Smith 2004), soft trajectory constraints (Gerevini and Long 2005; Baier, Bacchus, and McIlraith 2009), and as partial orderings over solutions (Boutilier et al. 2004).

In many situations it will not be appropriate for an isolated elicitation process to proceed the execution of the task. Moreover, within a human-agent interaction we see preference elicitation as a natural part of engagement. For instance, consider a tourist on a guided walking tour of a city. After reaching a place where they can see they are almost back to the starting point, the tour guide says "Let's go up that hill," pointing to a large hill. "We can get a good view of the city from there." However, on seeing the tired expression on the tourist's face, the guide adds "Or we can stop at that cafe over there and take a break." In this case it would be desirable for the agent to be able to use information elicited at execution time to influence the remainder of the execution.

In this paper we propose the *within task preference elicitation* (ITAPE) planning problem, which explicitly represents the agent's knowledge about the user's preference model, and how the agent's knowledge can develop as the interaction progresses. Our approach parameterises a utility model for a net benefit planning task with a set of (observable) user attributes. This set of user attributes are represented as unknown values in a partially observable planning model and can be accessed through guarded sensing actions (e.g., through asking a question when it becomes relevant), allowing the planner to reason with the possible alternative user utility models. In this work we define the ITAPE problem and present our framework for solving these problem. We present results examining its use in modified benchmark scenarios, including a new tour guide inspired planning domain.

The paper is organised as follows. We begin by presenting the planning background, a motivating scenario and the related work. We define the ITAPE planning problem and some key properties and then present our representation for the multi-user preference model. We present our framework that we use to plan in these domains. Finally, we present an evaluation of our approach and our conclusions.

Background

In this work we bring together partially observable planning with net benefit planning. In this section we provide the background for these problems. A classical planning problem can be defined as follows.

Definition 1. *A Classical Planning Problem is a planning problem, $P = \langle F, A, I, G\rangle$, with fluents, F, actions, A, initial state, I, and goals, G. A solution (a plan) is a sequence of actions, $\pi = a_0, \dots, a_n$, that transform the initial state, I, to a state, s_n, that satisfies the goals, $G \subseteq s_n$.*

An action is defined by a precondition and an effect and is applicable in a state if its precondition is satisfied by the state. The set S of states of a planning problem is the set of states that can be reached by applying any sequence of applicable actions to the initial state. The aim in classical planning is typically to find short plans.

Similar to (Menkes Van Den Briel, Do, and Kambhampati 2004), we extend the classical planning problem to a definition of a net benefit planning problem.

Definition 2. *A Net Benefit Planning Problem extends a classical planning problem, $P = \langle F, A, I, G\rangle$, with an action cost function, $C : A \mapsto \mathbb{Z}$ and a utility function, $u : S \mapsto \mathbb{Z}$, allocating utility to the final state. A solution is still a plan, π, leading to some state, s_n. The net benefit of an action sequence is given as: $NB(\pi, s_n) = u(s_n) - \sum_{a\in\pi} C(a)$.*

In net benefit planning the aim is to find sequences of actions that optimise overall net benefit.

Partially Observable Planning Problem

A partially observable planning problem, e.g., (Bonet and Geffner 2011), can be defined as follows.

Definition 3. *A Partially Observable Planning Problem, is defined by a tuple, $POP = \langle F, A, M, I, G\rangle$, with fluents F, actions A, sensor model M, the initial state clauses, I, and goal G. The clauses of the initial state provide both the known positive and negative literals, as well as constraints over the currently unknown parts of the initial state. A solution is a branched plan (a tree), where the nodes are actions or sensing actions. The plan branches on the possible values of the sensing action. The tree should describe a solution for any of the possible initial states that are consistent with the initial state constraints.*

Motivating Example: Tour Guide Agent

In this section we introduce an example scenario involving interaction between an agent and a user that will be used as an example throughout the paper. A tour guide agent directs a user through a tour of e.g., a town, stopping at the important landmarks on the route. An agent may aim to select a subset of possible sites (landmarks) that the user is most likely to enjoy (Castillo et al. 2008). Moreover, (as with the example used in the introduction), ideally a tour guide agent will use the feedback from the user, gathered during the tour, in order to influence the remainder of the tour. For example, if the agent notices that the user is disinterested during a museum trip it can use that observation in their subsequent selections.

Related Work

Preferences in planning (Jorge and McIlraith 2008), were introduced to the planning domain definition language in version 3.0 (Gerevini and Long 2005) and the use of heuristic search has proven successful (Baier, Bacchus, and McIlraith 2009). It was observed in (Nguyen et al. 2012) that having an accurate preference model is not always possible. This has led to approaches that generate a diverse set of plans that the user can pick from (Nguyen et al. 2012), planning based on assumptions of the preference model (Davis-Mendelow, Baier, and McIlraith 2013) and learning user preferences based on previous experience (Floyd, Drinkwater, and Aha 2015; Woodworth et al. 2018).

The problem of user preference or utility model elicitation has been studied, typically focused on optimising the quality of the model (Boutilier et al. 2006). In (Boutilier 2002) they develop a partially observable Markov decision process (POMDP) model, which allows the expected reward of asking a query to be considered while determining the questions to ask. However, preference elicitation is still completed prior to any task decisions are made. In the SAMAP system (Castillo et al. 2008), similar to our approach, user attributes (e.g., preference of sports as a leisure activity) are used as indirect elicitation of user preferences. However, in SAMAP the user model is selected before planning. Within social robotics, (Gucsi et al. 2020) use an annoyance cost model, which assigns a level of annoyance to each of the query actions. The intention then is to identify an optimal sequence of questions given the annoyance budget.

There are also approaches that combine elicitation and planning/execution within a single framework, such as the factory setting, user tailoring, execution tuning (FUTE) framework (Canal, Alenyà, and Torras 2016). These frameworks are typically iterative, specialising plan generation through either interleaving elicitation and planning episodes (Sanneman 2019), or learning from observations over time (Canal, Alenyà, and Torras 2016). The approach presented in (Das et al. 2018) supports actively eliciting preferences over task decompositions from a human expert during the planning process. In (Gilroy et al. 2012; Behnke et al. 2020) the systems can be adapted during execution (e.g., adding temporal constraints), although the planners do not reason about the possible user preferences or whether to elicit them. Using sensing or non-deterministic actions have been used to elicit user response to agent queries in dialogue systems (Petrick and Foster 2013; Botea et al. 2019). The robot bartender, presented in (Petrick and Foster 2013, 2016), used sensing actions to determine users' orders during a task-based social interaction. In (Chatterjee et al. 2020) they organise user data around various dimensions (e.g., age

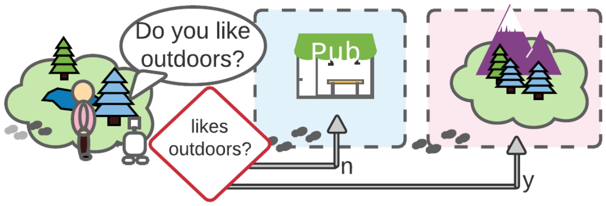

Figure 1: During the task the robot is able to elicit whether the user likes outdoors or not. This is then used in subsequent decisions of what activities should be planned.

and sex) and show that a Multiple-Environment Markov Decision Process can be used to capture the alternative user probability functions (e.g., what the user is likely to do given a certain recommendation). It is aimed at improving individual recommendation episodes, rather than reasoning about preference elicitation and sequential optimisation within the same framework.

A Within Task Elicitation Planning Problem

In this work we consider the problem of within task elicitation. Intuitively we aim to move elicitation from a process that happens before planning begins, to a process that happens during execution. As a consequence, at planning time there is uncertainty about the preferences of the users. Therefore offline planning is performed under uncertainty, and identifies the elicitation that should be performed during execution. Our approach parameterises a utility model for a net benefit planning task with a set of user attributes. This set of user attributes are represented as unknown values in a partially observable planning model and can be accessed through guarded sensing actions. The planner is provided with these sensing actions that can be used to elicit the information from the user. In this section we define our problem as a net benefit partially observable planning problem.

Within Task Elicitation

We follow (Castillo et al. 2008) in assuming that the preferences of a particular user can be determined based on a set of observations. For example, a tour guide might ask a question like ‘Do you like the outdoors?’ and if they answer positively, they might assume that the user will prefer landmarks such as parks, gardens and zoos. In the within task elicitation setting the values of these user attributes for a user during a specific interaction are either not known, or only partially known in advance. This means that the planner must reason about the set of possible valuations of user attributes and their associated utility functions. As the task unfolds and the agent has the opportunity to discover the attribute values for the particular user then the uncertainty about the user’s utility function reduces, allowing the subsequent part of the plan to be better tailored for the user.

User Attributes In this work we assume that the preferences of any given user can be determined based on a set of observations, which might be made during a task. To this end we define a set of user attributes, $X^U = X^U_0, \dots, X^U_p$, with domain of X^U_i denoted $\mathbb{D}(X^U_i)$. For example, a tour guide scenario might include the user attributes: $\{$*likes-outdoors*^{U},*likes-educational*^{U},*likes-social*$^U\}$, each with Boolean domains (e.g., *likes-outdoors*^{U} is `true` if the user likes being outdoors).

Elicitation Actions Each user attribute is associated with one or more sensing actions, which are called elicitation actions. These actions discover the value of the user attributes during execution. A user’s preferences and choices can be elicited through questions (Petrick and Foster 2016), but might also be inferred through implicit signals, e.g., (Gilroy et al. 2012; Izquierdo-Reyes et al. 2018). For example, we noted that a tour guide agent might ask the user whether they like outdoor activities and that a positive answer might lead the agent to promote outdoor landmarks. Similarly, during a trip to a museum it might become apparent that the user is not engaged and the agent may choose fewer educational sites in subsequent parts of the tour. In this work we do not distinguish between sensing of user’s preferences through observations and dialogue. However, it is worth noting that elicitation actions should be associated with carefully modelled constraints so that elicitations are only used when it is appropriate, e.g., observing a user’s response to some activity will only be available at relevant points during the task.

Possible Users We assume that the set of possible users are described by the enumeration of the possible value assignments to the user attributes. This set of assignments is denoted $\mathbf{X}^U$. We assume that each type of user, associated with some attribute values $\mathbf{x} \in \mathbf{X}^U$, are as important or preferred as any other.

The Problem Specification

We set the problem as a partially observable planning problem, e.g., (Bonet and Geffner 2011), using the hidden part of the state to represent a set of user attributes (X^U). Our problem is then set as a net benefit optimisation task, where the utility function can depend on the user’s attributes and the elicitation actions can incur cost. This allows the planner to trade-off the cost of elicitation with the utility benefits of knowing the user’s preferences.

Definition 4. *A Within Task Elicitation Planning Problem is a Partially Observable Net Benefit Planning Problem, defined from a partially observable planning problem,* $POP = \langle F, A, M, I, G\rangle$ *and extended with an action cost function,* $C : A \mapsto \mathbb{Z}$ *and a utility function,* $u : S \mapsto \mathbb{Z}$, *allocating utility to the final state. A solution,* π, *is a branched plan of actions and elicitation actions. In evaluating the utility of a solution plan we sum the utility of execution (utility minus cost) for each of the possible user types:*

$$\texttt{utility}(\pi) = \sum_{\mathbf{x}\in\mathbf{X}^U} [u(\texttt{apply}(\pi, \mathbf{x}, I)) - \sum_{a\in\texttt{lin}(\pi,\mathbf{x})} C(a) - \sum_{e\in\texttt{sens}(\pi,\mathbf{x})} C(e)]$$

Where $\mathbf{X}^U$ is the set of valuations of the user attributes, X^U; $\texttt{apply}(\pi, \mathtt{x}, I)$ applies the plan by following the branches consistent with $\mathtt{x}$ (returns the resulting state); $\texttt{lin}(\pi, \mathtt{x})$ is the action sequence (linearisation) extracted from π found by following branches consistent with $\mathtt{x}$; and $\texttt{sens}(\pi, \mathtt{x})$ is the same for sensing actions.

We can represent a tour guide scenario as a within task elicitation planning problem, by allowing the utility of the potential activities (e.g., climbing a hill) to vary depending on user attributes (e.g., whether they like outdoors). An example solution for a simple tour guide scenario is illustrated in Figure 1.

Partially Observable User Utility Model

The final states of a plan for a within task elicitation problem may be partial states, with partial information regarding the attributes of the user. This could be dealt with by either allowing the utility model to attribute utility to partial states, or to aggregate the utility of each of the potential concrete states described by the final partial state. In setting the general ITAPE problem we add structure into the model to ensure that the user attributes are all known in the final state. This has the following benefits:

- the model accurately reflects the utility valuation of each of the possible users (not aggregated);
- the utility model must be specified for the tuples of value assignments to the user attributes (and not unknowns);
- any goal can be achieved when its actual utility is not known;
- this added structure is a modelling artefact and therefore the required additional elicitation actions will not be executed (described below).

To this end the planning task is split into three distinct sections, which are illustrated in Figure 2. The first section captures the strategy adopted while performing the task (Figure 2 i.). In this case the planner selects elicitation questions that make important distinctions between user groups. In this section the elicitation of the user's attributes are associated with cost. At the end of the first section the task is completed and no further task actions can be applied.

The second part of the planning task completes the elicitation task (Figure 2 ii.), discovering the valuation of all user attributes not already discovered. In this case there is zero cost allocated with these elicitations. In the final step (Figure 2 iii.) each of the achieved goals is evaluated using the utility model, as the values of all user attributes are now known.[1] This final step is similar to an approach used in net benefit planning, which is used to compile utility scores into a cost function (Keyder and Geffner 2009). Notice during execution the second and third sections of the action traces are omitted: their role is to force the planner to consider the alternative costings of the plan.

This requires some machinery in terms of the planning model. Fluents are defined for each goal, which mark the progress of each goal through: *unachieved*, *achieved* and *costed*. A pair of actions, applicable in phase iii., is defined for each of the goals: the first replaces *unachieved* with *costed*, for no reward; the second replaces *achieved* with *costed* and has conditional effects for utility depending on the appropriate utility model.

A planner that reasons about the possible alternative utility functions, will be able to weigh up whether eliciting information and therefore allowing more tailored selection of goals, is worth the cost of the elicitation. Notice, if the planner does not elicit information during the task, then it will be selecting goals without certainty of the utility associated with the goal (with respect to the current user).

Properties of Within Task Preference Elicitation Planning Problems

Proposition 1. *The Uncertainty of a Within Task Preference Elicitation Planning Problem is monotonically decreasing.*

This follows from the type of partially observable planning problem that we use: the set of partially observable variables are defined up front and once valuated can not become unknown.

Proposition 2. *Any linearisation,* $\texttt{lin}(\pi, \mathbf{x})$ *for some* $\mathbf{x} \in \mathbf{X}^U$*, of a solution,* π*, is a solution for any other* $\mathbf{x}' \in \mathbf{X}^U$.

This holds because the user attributes are used to determine the appropriate user utility model and do not impact the causal structure or the cost model. This observation can be exploited in order that a plan, π, can be revised after construction and elicitation actions removed. Thus the balance between elicitation and utility can be further explored (in a similar manner to decision tree pruning) e.g., plan explanations could be included in the costing.

Representing Multi-User Preferences

An important aspect of the within task elicitation planning problem is the definition of a utility model that captures the preferences of a set of potential users. As with (Castillo et al. 2008) we assume that the preferences of any of these potential users can be determined based on a set of observations (e.g. questions or observations). Our approach builds on Generalised Additive Independence (GAI) models, which provide a general representation for capturing user preferences (Fishburn 1967; Braziunas and Boutilier 2006). GAIs combine additive functions for a set of attributes, which makes them appropriate models for defining utility functions for net benefit problems. A key advantage is that GAIs isolate the dependencies of these functions, allowing a utility model to be captured concisely. In this section we first define standard GAI models and then extend them so that the user attributes can be involved in determining plan utility.

GAI models

We follow the definition of GAI models presented in (Braziunas and Boutilier 2006). A GAI is defined as a set of attributes, $X = X_0, \ldots, X_n$, with finite domains: the domain of X_i is denoted $\mathbb{D}(X_i)$. Typically the set, $\mathbf{X}$, of possible outcomes (the solutions to be selected amongst) instantiate these attributes. For the purposes of this work, we associate an attribute with each goal of our net benefit problem.

[1] The figure assumes an additive factoring of the utility function.

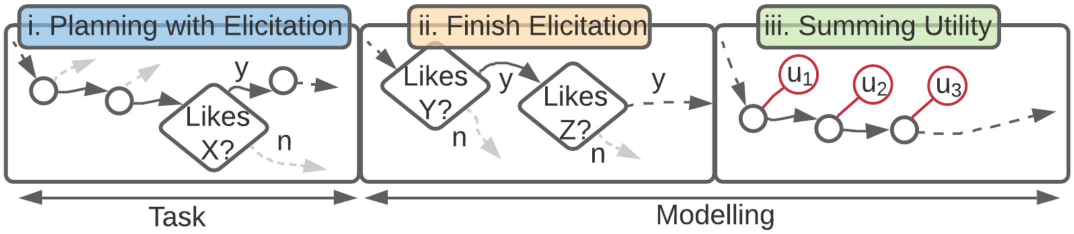

Figure 2: The three stages forced in the planning model. Stage i. involves performing the task and can include elicitation. Steps ii. and iii. are then model artefact to force the planner to reason about the user's true preferences.

Each attribute is a Boolean: the goal is either achieved or missed. E.g., in a tour guide scenario, the agent attributes might be: X={*pub*,*museum*,*park*,*art gallery*} (e.g., alternative landmarks that could be visited on the tour). Notice that the value of these attributes follow from a plan (i.e., the goals achieved in the final state) and are therefore selected by the planning agent.

The utility function is then defined by summing terms for each of the attributes (e.g., there are individual utility contributions associated with each attribute). The utility is determined for each attribute by a subset of the attributes that entirely determine its contribution to the overall utility of an outcome. In particular, each attribute is associated with a collection, $I_1, ..., I_m$, of possibly intersecting attribute index sets (or factors). For example, in the tour guide example, X_0 might only rely on $\{0\}$, i.e., the utility of the pub only depends on whether the tour visited the pub; whereas X_1 may depend on $\{1, 3\}$, i.e., the utility of the museum is dependent on whether the tour also includes the art gallery, as these might be considered similar. The utility function is then defined using sub-utility functions, e.g., $u_i(\mathbf{x}_{I_i})$, is the utility function for attribute i, which depends on the attributes in index set I_i, as follows:

$$u(\mathbf{x}) = \sum_{i=1}^{i \leq m} u_i(\mathbf{x}_{I_i})$$

GAI models can be used to capture a wide range of preferences, including preferences over the possible trajectories (e.g., describe as trajectory constraints and associate with a Boolean attribute). The main limitation is that the set of attributes must be identified up front.

User Observations as Attributes Our intention is to capture the preferences of all potential users in a single GAI model. In this work we assume that the preferences of any given user can be determined based on a set of observations, which might be made during a task (see 'Elicitation Actions' above). To this end we extend the GAI with the user attributes described above (see 'User Attributes'), allowing the utility of each goal to be dependent on these user attributes. These are incorporated into a GAI model as additional attributes, which allows the utility of goals to depend on them. Therefore the GAI attributes have two parts: they contain an attribute for each goal in the net benefit problem (chosen by the agent as described above) and a set of user attributes (i.e., that cannot be selected by the agent). We distinguish agent attributes with a superscript A and user attributes with a superscript U, e.g., the attributes can be written $X_1^A, \ldots, X_m^A, X_{m+1}^U, \ldots, X_{m+p+1}^U$. For example, a tour guide scenario might include the user attributes: {*likes-outdoors*^{U},*likes-educational*^{U},*likes-social*^{U}} (e.g., *likes-outdoors*^{U} is `true` if the user likes being outdoors), and agent attributes: {*pub*^{A},*museum*^{A},*park*^{A},…} (as before). As an example, the utility model for the pub goal might depend on the value of *likes-social*^{U}:

$$u(\mathit{pub}^A) = \begin{cases} 40 & \mathit{pub}^A = \texttt{true} \bigwedge \mathit{likes\text{-}social}^U = \texttt{true} \\ 20 & \mathit{pub}^A = \texttt{true} \bigwedge \mathit{likes\text{-}social}^U = \texttt{false} \\ 0 & \mathit{otherwise} \end{cases}$$

As a convenience, we denote the specific utility contribution, $u_{\mathbf{x}}(g)$, for a specific set of attributes, $\mathbf{x}$, and agent attribute, g. A GAI is used to capture the ITaPE utility model, such that the utility of any final state can be calculated by summing the corresponding contribution for each goal. We assume for this presentation that the utility of not achieving a goal is always zero, although there is no theoretical requirement. To summarise, the approach relies on the following assumptions:

- user preferences can be represented using the net benefit utility model (Jorge and McIlraith 2008);
- a set of user attributes are sufficient to distinguish between users with differing preferences (Castillo et al. 2008);
- an appropriate GAI model can be defined: either by domain experts, or extracted from user observations.

An Optimistic Approach for Solving Within Task Preference Elicitation Problems

We have developed a framework that takes as input: a net benefit planning model, a set of user attributes with associated constraints and a GAI, which describes the possible user utility models. The output is an ITaPE Problem, which is solved using an extended partially observable planner. Our approach is built on K-Replanner (Bonet and Geffner 2011), which is a partially observable planning system that uses a compilation to classical planning approach. Our initial investigations with full exploration of the problem in an AND-OR tree indicated that a full exploration approach is currently only feasible for small problems. Moreover the compilation approach is particularly suited to extending to the cost sensitive setting, which is not typically supported in partially observable planners. However, whereas, the formulation of the ITAPE problem is appropriate for solving approaches that reason directly with the uncertainty, optimistic approaches (such as K-Replanner) are likely to result in no user elicitation. In this section we demonstrate this issue and the approach that we use in order to allow the optimistic approach to solve these problems.

Limitations of an Optimistic Approach

In practice, a popular approach to partially observable problems has been to compile the problem into a classical planning problem. A key aspect of this encoding is that each sensing action is replaced by a pair of standard actions: one captures the effect of the sensor in the case that its proposition holds in the world and the other for the negative case. As a result the valuation of the sensors becomes a choice for the planner to make. Thus the classical planner will build an optimistic plan, which is based on the assumption that it can pick the values of sensor actions. In the case of ITAPE problems it means that in the compiled model, part ii. (see Figure 2) will allow any user model to be selected as part of a plan. The returned plans therefore use the context of an optimistic selection of the user attributes and do not need to elicit any information during the task. In part ii., where elicitation has zero cost, the planner selects the appropriate sensing options. For example, if the agent must select between a hill walk and a pub on a (one stop) tour, it can first visit the site (e.g., the pub) and end the task and then it can select the appropriate user attribute values (*likes-social*$^U = \texttt{true}$) that will lead to its selection attaining the best score.

Reformulation For an Optimistic Planner

We therefore propose an alternative statement of the problem tailored for this optimistic approach. Instead of the approach illustrated in Figure 2 utility is accumulated during the task stage. This is achieved by creating additional actions for each goal satisfying action. This additional one is connected with the utility (see next part) and commits to the goal remaining achieved (i.e., every goal destroying action is subsequently prevented). Notice that although the preventing goals to be unachieved could cause a typical partially observable planning problem to become unsolvable (i.e., for certain hidden states), in the case of ITAPE problems, because the classical planner finds a plan for a user attribute valuation and Proposition 2 states that this is suitable for any other user then the planner will still find a solution (perhaps not as good as it might have been).

Goal Utility for Partial Attribute Valuations

Notice that this has implications on the cost model. Whereas the machinery presented above guarantees that the complete attribute valuation was known, we now must contend with the possibility that at any given state the agent's knowledge of the user's attributes will be only partially known. And of course any goal might need a utility score attributed in any of these possible states. We have therefore considered approaches to calculate the utility of achieving a goal in a partial state, as an aggregation of the utility attributable to each of the possible concrete states.

The GAI model compartmentalises the utility function (e.g., the utility score for a museum only depends on *likes-educational*^{U}). As such the utility of each goal is determined by a mapping of partial attribute valuations to utility scores. We consider two aggregation approaches in this work:

1. Fully enumerate each partial attribute valuation and map them to a utility score. Each partial valuation maps to the mean of all relevant concrete state valuations.
2. For each attribute only extend the utility model with a single entry for when that attribute is unknown (irrespective of any other attributes). Attribute the minimum utility for that goal.

Because the first approach allocates utility individually to each of the partial states its value is sensitive to partial knowledge about the user attribute values. Its value will be higher than (or equal to) the lowest value, which promotes unknown (potentially better utility) goals over the lowest utility goals in tie-break situations. The second approach only requires a number of additional entries in the utility mapping linear in the number of relevant attributes. However, it is therefore much less sensitive to partial user information. Notice in both cases the utility score is lower than (or equal to) the maximum utility value for the goal (given the known user attributes), which incentivises the planner to elicit user attributes prior to achieving goals (this only holds because we use the minimum in the second approach).

Encoding as a Minimisation Optimisation Problem

Our current framework uses Lama as the planner that underpins the K-Replanning system, allowing us to exploit its powerful heuristics and optimisation. We therefore translate the net benefit optimisation problem to an equivalent minimisation problem (Keyder and Geffner 2009). The translation requires the following steps:

- For each goal, g, identify the maximum utility for that goal: $u_g^{MAX} = \max_{\mathbf{x} \in \mathbf{X}} u_x(g)$
- For each goal, g, add u_g^{MAX} to the cost of the miss action, $\texttt{miss}_g$ (each goal is either committed to or explicitly missed using the corresponding $\texttt{miss}_g$ action).
- For each goal, g, replace the utility attributed to a goal, $u_x(g)$, with the cost: $C_x(g) = u_g^{MAX} - u_x(g)$.

Intuitively, when a goal is missed, instead of missing utility, high cost is incurred instead. Similarly, when achieving the goal, if the goal is associated with high utility (i.e., to a specific user) then it will incur low cost.

Evaluation

In this section we first present the setup of our evaluation and then present the results. We have implemented the presented framework and have performed a proof of concept evaluation to investigate its performance on ITAPE problems. The baseline approach, used for comparison in the evaluation, assumes no information can be elicited and so all goal utility scores are averaged over the possible utility scores for the goal. We have used synthetic data sets, which

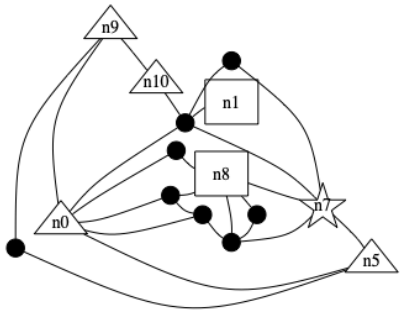

Figure 3: An example of a map for the Tour Guide domain with 15 nodes, including optional sights (triangles), compulsory sights (boxes) and the start/end of the tour (star). Unlabelled dots are locations with no landmarks.

have varied utilities for each preferred and non-preferred alternatives. We assume that users associated with a specific set of attributes will agree with the utility score represented in the GAI for those attributes. In particular, we have abstracted from the noise of real scenarios in order to test the feasibility of our system in responding appropriately to user preferences uncovered during execution.

The approaches we compare in the evaluation are:

K-R-E Our approach was built on K-Replanner (Bonet and Geffner 2011). K-Replanner has been extended to generate the complete contingency tree and for the cost sensitive planning setting. The system also includes the remodelling steps and precompilation of the GAI utility model described above;

Baseline The baseline uses LAMA-11 and has no access to within task elicitation. The baseline used averaged cost models (i.e., no assumption made about likelihood of preferences).

The baseline planner and the classical planner used with K-Replanner was the LAMA-11 configuration of Fast Downwards (Richter and Westphal 2010) with 6Gb RAM and a 10 minute time-out. The problems presented to the systems only differ in the following: the K-R-E model is extended with the variables associated with the user's attributes and the cost model is predicated on those variables.

For the purposes of this study we have created a set of benchmark problems (either created or adapted from existing domains). We test two versions: *unconstrained*, where sensing actions (elicitations) have no conditions and can be done at any state; and *constrained* where the modelled condition constrains elicitation:

tour-guide: A new domain involving navigating a map and visiting different landmarks. Problem generation starts from a planar graph, following the approach in (Gregory and Lindsay 2007) that aims to make more realistic map networks, see Figure 3. Landmarks are organised into general categories (in this case, educational, social or outdoors) and the follower has preferences over the categories (e.g., likes outdoors). Constraint: When they are located at a landmark of a certain type the agent can query whether the user likes that type of landmarks.

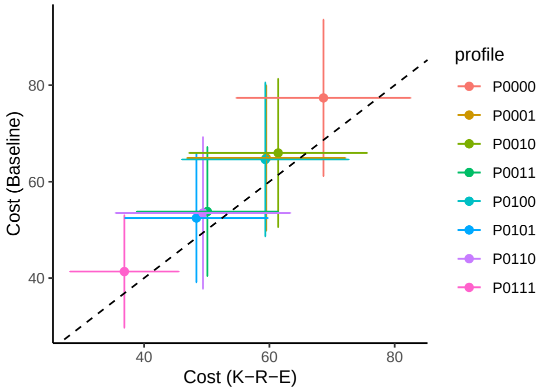

Figure 4: Results in the tour domain aggregated for each type of user. The codes indicate 0: dislikes and 1: likes, for each attribute. E.g., P0111 means that the user likes social, educatory and outdoor activities.

instruction giving: the briefcase domain was extended with alternative methods of providing instructions to the follower. The packages are divided into heavy and light and the follower has preferences for the alternative styles of instructions for collecting/depositing each of the types of package. Constraint: the follower's response to a type of instruction can be sensed.

rovers: simulates an interested operator, who has preferences for types of goals, observing plan execution of a rover. The agent can ask the operator's interests (i.e., rock, soil or image goal types). Constraint: Questions are applicable at relevant states (e.g., ask about preference for rock goals at locations with rocks).

bar-tender: the bartender chooses either to make and serve a cocktail (goal) or miss it out. The customer's utility is based on their preferences for the constituent ingredients (e.g., likes ingredient1, but does not like ingredient2). No constrained version.

Improving Plan Utility

Table 1 presents the results of our experiment. The accumulated cost is presented for the Baseline and K-R-E approaches, alongside the percentage reduction. In the constrained sensing setting (upper part of the table) the results show that the K-R-E approach leads to an overall reduction in cost compared with Baseline in the three domains. As expected, the results for the unconstrained models outperform the constrained models. The results demonstrate that over constraining elicitation actions can lead to weaker performance. This suggests that allowing weakly constrained alternatives (e.g., with higher cost), where appropriate, might allow the planner to determine the most appropriate use of elicitation. For example, if discovering whether the user likes educational activities heavily dictates the best course of action from the beginning then the agent might choose an elicitation option to discover this preference upfront. This action would be associated with a cost to reflect its impact

	Domain	Solved	Avg. Nodes	Avg. Goals	User Atts	Accumulated Cost		Cost Reduction(%)
						Baseline	**K-R-E**	
Constrained	tour-guide(20)	20	171.6	13.5	3	9472	8775	7.36%
	instuct(10)	10	75.3	11.6	2	2438	2244	7.96%
	rovers(20)	20	96.8	8.95	3	6896	6435	6.69%
Unconstrained	tour-guide(20)	20	173.85	13.5	3	9472	8663	8.54%
	instuct(10)	10	71.9	11.6	2	2438	2196	9.93%
	rovers(20)	20	106.2	8.95	3	6896	6249	9.38%
	bar-tender(10)	10	122.1	2.2	3	6126	5724	6.56%

Table 1: The table reports the accumulated cost for Baseline and K-R-E along with the percentage reduction, for four domains (number of instances in brackets) in controlled (top) and uncontrolled (bottom) sensing modes. The costs are accumulated for all user types and problems. The averaged number of nodes in the generated trees are also presented.

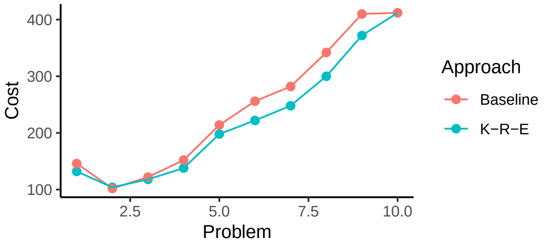

Figure 5: Mean cost by problem in the instruction giving domain. The plot demonstrates the improvement window of the approach.

on the interaction. We can further analyse the results to determine how user types are effected. In the bar-tender and tour-guide domains, K-R-E improves the cost most for users with fewer positive attributes (e.g., likes fewer ingredients). Figure 4 plots the average costs for specific user attribute valuations in the tour-guide domain. E.g., P0101 gathers the users that like social and outdoor activities, but do not like educational sites. Points above the line indicate improvement for the K-R-E approach. These results suggest that the baseline plans involve committing to goals (e.g., serving the majority of the drinks), which have higher cost for users that have fewer positive attributes, providing more room for improvement for these user types. The improvement is evenly distributed across the user types in the rovers domain.

Problem Size

Further analysis of the results also demonstrate that the problem size and structure play important roles in the performance. In particular, for each domain there will be a level of difficulty where the K-R-E approach will be most effective. Figure 5 illustrates this window in the instruction giving domain. If the problem is too small there is no opportunity to elicit information, allowing no customisation. If too large then the underlying classical planning approach can fail to commit to any goals, leading to weaker performance.

The problem with larger problems is inherited from limitations of using the conversion of a net-benefit to a minimisation problem. In solving these problems it is typical for the planner to first find a plan that misses all of the goals, as this will be the shortest plan. In order to find improved cost solutions requires an exploration of longer plans. In tour-guide additional goals can often be added to a plan with only some small number of additional steps, allowing improved cost bounds to be discovered incrementally. This has led to the approach being effective in relatively large problems (an average of 13.5 goals). In contrast in bar-tender each of the goals is fairly independent and takes a long sequence of actions to achieve. As the number of goals is increased, the space of alternative solutions of each length grows and finding better plans becomes infeasible with the resources. As a result the approach is only suitable for problems with far fewer goals (an average of 2.2 goals). Notice that for problem 10 in Figure 5 both approaches fail to commit to any goals, leading to the generated plans having the same cost. However, as the K-R-E approach adds some complexity to the model, there may be cases where the baseline approach is able to find interesting solutions where the K-R-E approach cannot find them (within the given resources).

Conclusion and Future Work

We are interested in allowing an agent to adapt its behaviour to a particular human interaction partner during an interaction. To this end we have posed the within task elicitation problem, which allows the agent to reason over multiple potential user preference models. User attributes are represented as initially unknown variables in the planning model and associated with elicitation actions, which can be used to distinguish between user preferences. We presented a framework that supports our optimistic approach to solving these problems. In our evaluation we observed that the framework could discover better plans than the baseline in each of the tested domains. This provides a new approach for specialising plan-based agent behaviour in situations where upfront elicitation is not feasible. We are currently running a user study examining human response within human agent interactions (Lindsay et al. 2020b,a) and are collecting user preference information as part of this study. We aim to explore the use of this data in constructing ITAPE problems.

Acknowledgements

This work was funded by the UK's EPSRC Human-Like Computing programme under grant number EP/R031045/1 and the ORCA Hub (`orcahub.org`), under EPSRC grant EP/R026173/1.

References

Baier, J. A.; Bacchus, F.; and McIlraith, S. A. 2009. A heuristic search approach to planning with temporally extended preferences. *Artificial Intelligence* 173(5-6): 593–618.

Behnke, G.; Bercher, P.; Kraus, M.; Schiller, M.; Mickeleit, K.; Häge, T.; Dorna, M.; Dambier, M.; Manstetten, D.; Minker, W.; Glimm, B.; and Biundo, S. 2020. New Developments for Robert – Assisting Novice Users Even Better in DIY Projects. In *Proceedings of the 30th International Conference on Automated Planning and Scheduling*.

Bonet, B.; and Geffner, H. 2011. Planning under partial observability by classical replanning: Theory and experiments. In *International Joint Conference on Artificial Intelligence*.

Botea, A.; Muise, C.; Agarwal, S.; Alkan, O.; Bajgar, O.; Daly, E.; Kishimoto, A.; Lastras, L.; Marinescu, R.; Ondrej, J.; Pedemonte, P.; and Vodolán, M. 2019. Generating dialogue agents via automated planning. *arXiv preprint arXiv:1902.00771* .

Boutilier, C. 2002. A POMDP formulation of preference elicitation problems. In *AAAI/IAAI*, 239–246. Edmonton, AB.

Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H. H.; and Poole, D. 2004. CP-nets: A tool for representing and reasoning withconditional ceteris paribus preference statements. *Journal of artificial intelligence research* 21: 135–191.

Boutilier, C.; Patrascu, R.; Poupart, P.; and Schuurmans, D. 2006. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence* 170(8-9): 686–713.

Braziunas, D.; and Boutilier, C. 2006. Preference elicitation and generalized additive utility. In *AAAI*.

Canal, G.; Alenyà, G.; and Torras, C. 2016. Personalization framework for adaptive robotic feeding assistance. In *International Conference on Social Robotics*, 22–31. Springer.

Canal, G.; Alenyà, G.; and Torras, C. 2017. A taxonomy of preferences for physically assistive robots. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*.

Castillo, L.; Armengol, E.; Onaindía, E.; Sebastiá, L.; González-Boticario, J.; Rodríguez, A.; Fernández, S.; Arias, J. D.; and Borrajo, D. 2008. SAMAP: An user-oriented adaptive system for planning tourist visits. *Expert Systems with Applications* 34(2).

Chatterjee, K.; Chmelík, M.; Karkhanis, D.; Novotný, P.; and Royer, A. 2020. Multiple-Environment Markov Decision Processes: Efficient Analysis and Applications. In *Proceedings of the International Conference on Automated Planning and Scheduling*.

Das, M.; Odom, P.; Islam, M. R.; Doppa, J. R.; Roth, D.; and Natarajan, S. 2018. Preference-Guided Planning: An Active Elicitation Approach. In *Proceedings of the International Conference on Autonomous Agents and Multi Agent Systems*.

Davis-Mendelow, S.; Baier, J. A.; and McIlraith, S. 2013. Assumption-based planning: Generating plans and explanations under incomplete knowledge. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Fishburn, P. C. 1967. Interdependence and Additivity in Multivariate, Unidimensional Expected Utility Theory. *International Economic Review* 8(3): 335–342.

Floyd, M.; Drinkwater, M.; and Aha, D. 2015. Trust-guided behavior adaptation using case-based reasoning. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Gerevini, A.; and Long, D. 2005. Plan constraints and preferences in PDDL3. Technical report, Technical Report 2005-08-07, Department of Electronics for Automation.

Gilroy, S.; Porteous, J.; Charles, F.; and Cavazza, M. 2012. Exploring passive user interaction for adaptive narratives. In *Proceedings of the ACM international conference on Intelligent User Interfaces*.

Gregory, P.; and Lindsay, A. 2007. The dimensions of driverlog. In *Workshop of the UK Planning and Scheduling SIG (PlanSIG)*.

Gucsi, B.; Tarapore, D.; Yeoh, W.; Amato, C.; and Tran-Thanh, L. 2020. To Ask or Not to Ask: A User Annoyance Aware Preference Elicitation Framework for Social Robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-20)*.

Izquierdo-Reyes, J.; Ramirez-Mendoza, R. A.; Bustamante-Bello, M. R.; Pons-Rovira, J. L.; and Gonzalez-Vargas, J. E. 2018. Emotion recognition for semi-autonomous vehicles framework. *International Journal on Interactive Design and Manufacturing* .

Jorge, A.; and McIlraith, S. A. 2008. Planning with preferences. *AI Magazine* 29(4): 25–25.

Keyder, E.; and Geffner, H. 2009. Soft goals can be compiled away. *Journal of Artificial Intelligence Research* 36: 547–556.

Kragic, D.; Gustafson, J.; Karaoguz, H.; Jensfelt, P.; and Krug, R. 2018. Interactive, Collaborative Robots: Challenges and Opportunities. In *IJCAI*, 18–25.

Lindsay, A.; Craenen, B.; Dalzel-Job, S.; Hill, R. L.; and Petrick, R. P. A. 2020a. Investigating Human Response, Behaviour, and Preference in Joint-Task Interaction. In *ICAPS 2020 Workshop on Explainable Planning (XAIP)*.

Lindsay, A.; Craenen, B.; Dalzel-Job, S.; Hill, R. L.; and Petrick, R. P. A. 2020b. Supporting an Online Investigation of User Interaction with an XAIP Agent. In *ICAPS 2020 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*.

Menkes Van Den Briel, R. S.; Do, M. B.; and Kambhampati, S. 2004. Effective approaches for partial satisfaction (oversubscription) planning. In *Proceedings of the 19th national conference on Artifical intelligence*, 562–569.

Nguyen, T. A.; Do, M.; Gerevini, A. E.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence* .

Petrick, R. P.; and Foster, M. E. 2016. Using General-Purpose Planning for Action Selection in Human-Robot Interaction. In *AAAI Fall Symposium on Artificial Intelligence for Human-Robot Interaction 2016*.

Petrick, R. P. A.; and Foster, M. E. 2013. Planning for Social Interaction in a Robot Bartender Domain. In *ICAPS*.

Richter, S.; and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research* 39: 127–177.

Sanneman, L. 2019. Preference Elicitation and Explanation in Iterative Planning. In *Proceedings of the International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

Smith, D. E. 2004. Choosing Objectives in Over-Subscription Planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*.

Woodworth, B.; Ferrari, F.; Zosa, T. E.; and Riek, L. D. 2018. Preference learning in assistive robotics: Observational repeated inverse reinforcement learning. In *Machine Learning for Healthcare Conference*, 420–439.