Incremental Estimation of Users' Expertise Level

Pamela Carreno-Medrano Abhinav Dahiya Stephen L. Smith Dana Kulić

***Abstract*— Estimating a user's expertise level based on observations of their actions will result in better human-robot collaboration, by enabling the robot to adjust its behaviour and the assistance it provides according to the skills of the particular user it's interacting with. This paper details an approach to incrementally and continually estimate the expertise of a user whose goal is to optimally complete a given task. The user's expertise level, here represented as a scalar parameter, is estimated by evaluating how far their actions are from optimal. The proposed approach was tested using data from an online study where participants were asked to complete various instances of a simulated kitting task. An optimal planner was used to estimate the "goodness" of all available actions at any given task state. We found that our expertise level estimates correlate strongly with observed after-task performance metrics and that it is possible to differentiate novices from experts after observing, on average, 33% of the errors made by the novices.**

HRI, human profiling, expertise estimation

I. Introduction

The success of human-robot teams in domains such as manufacturing, logistics and in-home assistance crucially depends on understanding the human factors that influence team dynamics and performance. In these domains, the productivity and efficiency of a human-robot team is strongly affected by the capability of the robot to model its partner and to adapt its own behavior accordingly [1]. This includes, for example, modifying the robot's behaviour based on the human's intent, awareness, engagement or trust.

Although it is less often studied, expertise is another factor that can play an important role in human-robot collaboration [2], [3]. With accurate knowledge of the user's expertise for a given task, the robot can better decide the frequency and level of assistance that its teammate requires. Knowledge of the user expertise level will also help the robot to plan possible interventions during the task in order to maximize the team's overall performance while minimizing interruptions. Moreover, if the robot cannot complete its own actions successfully, due to an error or a novel situation, the robot can decide to observe and learn from a known expert [4].

In order for a robot to provide expertise-aware assistance, we first need to reliably estimate a user's expertise level during a task. Existing approaches rely on task-dependent performance metrics [3] and can only be estimated upon task completion [2], which limits the possibility of a robot adapting to the user's expertise level online during task execution. In this paper, we propose a method that incrementally and continually estimates a user's expertise level. In particular, we are interested in the early detection of non-experts, also called novices, since they are likely to benefit the most from the robot's timely assistance. We estimate a user's expertise level by looking at their actions rather than after-task performance metrics. To do so, we leverage the *rationality coefficient* first introduced in [5] as an indicator of an individual's ability to choose actions that will result in better performance [6]. That is, a user that frequently chooses the best actions is considered to be an expert while a user that often makes mistakes is considered to be a novice. We refer to this rationality coefficient as our expertise level estimate.

This paper makes three key contributions: (1) an extension of the Bayesian framework proposed in [6] to estimate a user's expertise level in an incremental and continuous manner; (2) the use of an optimal planner to determine the value of all actions available at each task state; and (3) the validation of the proposed framework on human data collected during execution of a simulated kitting task. Together, these three contributions lay the foundation of a future robotic system capable of inferring, reasoning about and adapting its behavior to the expertise level of the user with whom it interacts.

The following section briefly discusses the existing work on the estimation of a user's expertise level. Section III describes the proposed approach for incrementally estimating a user's expertise level. Section IV details the task, planner, and user study used for validation of the proposed approach. In Section V we present the results of the proposed expertise estimation on our user study's data. We conclude the paper in Section VI with a discussion of our findings, limitations and future work.

II. Related Work

The study and inference of a user's expertise level is a shared problem between the human-computer (HCI) and human-robot (HRI) interaction research communities. In this section, we briefly review existing work in both fields. Similarly, since the aforementioned *rationality coefficient* is a key aspect of the method proposed in this paper, we also review past applications of this coefficient.

A. Expertise in Human-Robot Interaction

One common approach for estimating an individual's expertise is supervised learning, where task-specific measures of performance are used directly to predict expertise. In [3], an individual's skill level is predicted using a binary classifier

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1 Canada (pcarreno@uwaterloo.ca; a4dahiya@uwaterloo.ca; stephen.smith@uwaterloo.ca; dana.kulic@uwaterloo.ca)

that receives as input a vector containing 7 carefully selected performance metrics. The binary classifier was trained using data from a sample of users trained across several days. Data from the first and last days of training was used to define the novice and expert categories respectively. In [7], Hidden Markov Models (HMM) were used to define an expert model that encodes expert surgical gestures. This expert model was trained using the kinematic data of experienced surgeons on a surgical simulator. The expertise level of a new user was determined by comparing their performance to the reference expert model.

Other approaches for expertise prediction include questionnaires that measure a user's knowledge about a specific task [8] and task modeling approaches such as Markov Decision Processes (MDP). In [2], the authors used a hand-tuned partially observable MDP (POMDP) model to predict the likelihood of a user being an expert. The robot holds an initial belief about a user's expertise and makes an update to this belief after each task instance has been completed.

In this paper, instead of employing task-dependent performance metrics, we propose an incremental approach that predicts a user's expertise level by looking at their action choices and the values of those choices. The value of each action is obtained using a general task planner.

B. Expertise in Human-Computer Interaction

Similar performance-based approaches for expertise prediction have been used by the HCI research community. However, instead of relying on expert demonstrations, expertise is established on the basis of an observed population. In [9], the authors defined a set of performance metrics that best summarized the skills needed to succeed at a rhythmic target-hitting task. The expertise of each participant was later established by comparing their individual performance measures to the mean and standard deviation of the full group.

The prediction of expertise and/or skill level is particularly relevant in the domain of intelligent learning systems. Two main approaches are commonly used: Bayesian knowledge tracing (BKT) and item response theory (IRT) [10]. In BKT, a skill is modeled as a latent variable in a HMM. Given an interaction in which this skill is applied, updates on the latent variable are done based on the correctness of the interaction. In IRT, given a sequence of observed actions or answers, skill mastery is estimated by maximizing its probability of occurrence given the observed sequence. Both approaches require an expert to define the relations between the performance indicators, skills and actions [6].

In this paper, expertise predictions are based on an individual's data alone and population samples or measures are not needed. Similarly, rather than identifying and using performance indicators for each skill of interest, we predict expertise at a task in which multiple skills are involved by considering the value of each action taken during the task.

C. Human Modeling and Prediction

First introduced in [5] as a measure of the degree of confidence on an agent's ability to chose highly rewarded actions, the aforementioned *rationality coefficient* has been applied in multiple HRI scenarios in the last decade. In their "Bayesian theory of mind", [11] used this coefficient to model a human's capacity for reasoning about the goals and beliefs of others. In [12], the *rationality coefficient* was used to infer a human's understanding on the effect of their actions on the dynamics of the world.

Instead of using the *rationality coefficient* to reason about a human's mental model and capacities, [13] used it as an indicator of how well a robot's model can describe a human's current motion. Similarly, in [14], this same coefficient is used to detect whether a human's demonstrations and corrections are relevant for behaviours a robot is currently learning. Closer to the work proposed in this paper, in [6], the *rationality coefficient* is used to infer an individual's expertise level. However, their approach requires fully observable sequences of actions. Additionally, an MDP model has to be solved for each possible coefficient value. In [15], a Bayesian framework is proposed in order to estimate a user's adaptability. In this paper we apply a similar formulation to the estimate of a user's expertise level.

III. Problem Statement and Approach

A. Human Expertise Modeling

We consider the setting where an observer $\mathcal{R}$ (the robot) watches an agent $\mathcal{H}$ (the human) complete a given task. Both the observer and the agent share the same knowledge about the states, actions and goals of the task. We model each task as a transition system tuple $\langle S, A \rangle$, where S is the finite set $\{s_1, ..., s_N\}$ of task states and A is the transition relation between different states. In our case, A corresponds to the finite set of actions allowed across all states and $A(s_t)$ defines the set of actions available at state s_t. We define a policy π as a mapping from the state space to a probability distribution over the action space, that is, $\pi : S \times A \rightarrow [0, 1]$. Similarly, we define an optimal policy $\pi^*(s, a)$ as a stationary policy that incurs the minimum cost,

$$\pi^*(s_t, a_t) = \begin{cases} 1, & \text{if } \arg\min_a Q^*(s_t, a_t) \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $Q^*(s_t, a_t) = C(a_t) + \arg\min_a Q^*(s_{t+1}, a_{t+1})$, referred hereinafter as the action value, is the cost of action a_t, $C(a_t)$, plus the total cost required to complete the task after executing action a_t at state s_t and transitioning to state s_{t+1}.

We consider tasks whose goal is to minimize the number of actions required for completion. Given this type of task and our action value definition, the set of actions $A(s_t)$ available at each state s_t can be grouped into 3 different categories: optimal ($A^*(s_t)$), redundant ($A^{\text{red}}(s_t)$) or mistakes ($A^{\text{miss}}(s_t)$); with $A(s_t) = A^*(s_t) \cup A^{\text{red}}(s_t) \cup A^{\text{miss}}(s_t)$. In our test scenario, all actions have the same cost $C(a_i) = 1$. Thus, although the number of actions in each category varies between different states, the same action value is shared among all actions within the same category. We assume that $\mathcal{R}$ knows the optimal policy, $\pi^*(s_t, a_t)$, required to complete such a task, and it uses this knowledge to gauge $\mathcal{H}$'s expertise. If the action choices made by $\mathcal{H}$ result in a lower total cost, the observer $\mathcal{R}$ considers the agent $\mathcal{H}$ to be an expert. If, on the contrary, $\mathcal{H}$ makes choices that result in a higher total cost, the observer $\mathcal{R}$ considers the agent $\mathcal{H}$ to be an non-expert.

The behavior of agent $\mathcal{H}$ is modeled according to the *principle of rational action*, which states that an agent is more likely to choose actions that they believe will help them achieve their goals [16]. In other words, agent $\mathcal{H}$ also aims at selecting actions with the smallest cost. However, $\mathcal{H}$'s capability to choose such actions is limited by their expertise. That is, the more skilled (or expert) agent $\mathcal{H}$ is, the more likely they are to choose optimal actions. We model the human's level of expertise through a Boltzmann policy [5], [11], [12],

$$\pi(s_t, a_t) = p(a_t|s_t, \beta) = \frac{1}{Z_{s_t}} \exp(-\beta Q^*(s_t, a_t)) \quad (2)$$

where s_t corresponds to the current task state, $Q^*(s_t, a_t)$ defines the cost of taking action a_t at the given state s_t under the optimal action policy $\pi^*(s_t, a_t)$ known by the observer $\mathcal{R}$, and Z_{s_t} represents an appropriate normalization constant. In our case, this constant was adapted to be independent of the number of actions of each type (i.e., optimal, redundant or mistakes) available at each state and it is defined as $Z_{s_t} = \text{sum}(Q^*(x, s_t), Q^*(y, s_t), Q^*(w, s_t)$, with $x \in A^*(s_t), y \in A^{\text{red}}(s_t)$, and $w \in A^{\text{miss}}(s_t)$. Similarly, to guarantee that $\sum_{\tilde{a} \in A(s_t)} p(\tilde{a}|s_t, \beta) = 1$, we compute probabilities over the action categories available at each state rather than over the set of all individual actions.

In this model, the coefficient $\beta \in [0, \infty)$ is called the expertise coefficient and determines the degree to which agent $\mathcal{H}$ is capable of following the optimal policy known by $\mathcal{R}$. As $\beta \to \infty$, $\mathcal{H}$ is seen as more of an expert ($p(a_t|s_t, \beta) \to \pi^*(a_t|s_t)$ for all $s_t \in S$ and $a_t \in A$); and as $\beta \to 0$, $\mathcal{H}$ is seen as more of a novice who selects actions uniformly at random from the set of available actions ($p(a_t|s_t, \beta) \to \text{Unif}(A(s_t))$ for all $s_t \in S$ and $a_t \in A$). While this model is simple, similar scalar models have been applied to other human factors including trust [17] and intent [18], and have resulted in improved performance and human perception.

B. Expertise Estimation Through Bayesian Inference

Following the model proposed in [6], we assume that for the *j*-th agent $\mathcal{H}$ there is a specific β_j parameter that provides a good estimate of this agent's expertise. Given a series of observations of the behavior of an agent $\mathcal{H}_j$ at a given task,

$$O_j = \{(s_1, a_1), (s_2, a_2), ..., (s_T, a_T)\}, \quad (3)$$

where in state s_i agent $\mathcal{H}_j$ took action a_i, we want to make inferences about the expertise level of agent $\mathcal{H}_j$. Formally, we want to compute the posterior distribution over possible β_j values and determine which one is the most probable given the observed behavior of agent $\mathcal{H}_j$,

$$p(\beta_j|O_j) \propto p(O_j|\beta_j)p(\beta_j) \quad (4)$$

To calculate this distribution, the prior probability $p(\beta_j)$ needs to be specified and the likelihood of the observed data $p(O_j|\beta_j)$ computed. Since our formulation allow us to take each action to be conditionally independent given the current state and the agent's expertise coefficient β_j, the likelihood $p(O_j|\beta_j)$ can be written as

$$p(O_j|\beta_j) = \prod_{t=1}^{T} \frac{1}{Z_{s_t}} \exp\left(-\beta_j Q^*(s_t, a_t)\right). \quad (5)$$

After the posterior is computed, the maximum a posteriori (MAP) estimate is taken as the predicted expertise coefficient $\hat{\beta}_j$ of agent $\mathcal{H}_j$. This point estimate corresponds to the β_j value at which the posterior distribution is maximized,

$$\hat{\beta}_j = \arg\max_{\beta_j} p(\beta_j|O_j). \quad (6)$$

C. Incremental Expertise Inference

Given that each observed action is conditionally independent given the current task state and the agent's expertise coefficient β_j, the proposed Bayesian inference framework provides us with a simple way to incrementally compute the expertise coefficient β_j of any agent $\mathcal{H}_j$. That is, for $O = \{o_1, o_2, o_3\}$ with $o_i = (s_i, a_i)$,

$$\begin{aligned} p(\beta_j|o_1, o_2, o_3) &\propto p(o_1, o_2, o_3|\beta_j)p(\beta_j), \\ &= p(o_2, o_3|\beta_j)p(o_1|\beta_j)p(\beta_j), \\ &= \text{Likelihood of}(o_2, o_3) \times \text{posterior after } o_1 \end{aligned} \quad (7)$$

More generally, the posterior having observed action-states pairs $(o_1, ..., o_K)$ is treated as the prior for any new set of incoming observations $(o_{K+1}, ..., o_T)$. A β_j estimate obtained in this way after observing all T *state-action* pairs is equivalent to the result obtained after seeing all observations at once.

IV. Experimental Design

The proposed expertise inference approach was tested using observations obtained on a simulated kitting task.

A. Task Definition

Kitting is a common task in manufacturing assembly that can benefit from human-robot collaboration [19]. Kitting refers to the procedure of gathering all the components and tools required for a task, and arranging them in a specific way on a *kit tray* such that they can be efficiently picked up or used during the manufacturing process. For this paper, a simulated version of a kitting task was implemented. Figure 1 shows the configuration of the simulation environment at the beginning of a kitting task. The task starts with a number of *containers* (item (i) in Figure 1) stacked together on a

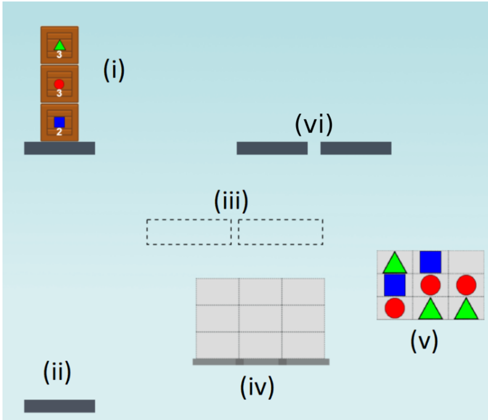


Fig. 1: View of simulated kitting task showing the elements of the simulation: (i) containers, (ii) unloading platform, (iii) unloading locations, (iv) kit tray, (v) target kit arrangement, and (vi) container platforms.

container platform (item (vi) in Figure 1), each containing a number of different *components*.

In order to place a component on the *kit tray*[1] the user needs to execute the following sequence of actions: **1)** bring the *container* that holds the target component to the *unloading location* (item (ii) in Figure 1); **2)** unload the component onto the unloading location (item (iii) in Figure 1); **3)** place the component at its target location in the *kit tray* (item (iv) in Figure 1); and **4)** return the container to the *container platforms* (item (vi) in Figure 1). There is a limit on the total number of *components* that can be unloaded at a given time and the number of *container platforms* on which the containers can be stacked. Similarly, the number of *containers* and *components* to be placed can vary between kitting tasks.

In order to complete this task, an agent or participant must arrange, on the *kit tray*, all components as indicated by the *target kit arrangement* (item (v) in Figure 1) while aiming for minimum number of actions possible. Transitions between task states are deterministic and all actions have 1 unit cost.

B. Estimating Action Values With a Planner

The action values $Q^*(s_t, a_t)$ required for computing the likelihood of any β expertise coefficient are obtained from a planner built on the Planning Domain Definition Language (PDDL) framework [20]. The PDDL provides a formal way to represent the planning domain, problem instances, constraints, and actions. After encoding our kitting task in this representation, a symbolic planner was implemented to compute a plan with the minimal number of actions as well as the values of all actions available at a given task state. The value $Q^*(s_t, a_t)$ of a state-action pair (s_t, a_t) is then defined as the number of actions needed to reach the task's goal given that action a_t has been taken at state s_t.

[1] A video showing how a simulated kitting task is completed can be be found at https://youtu.be/hsg6aCZYQCo

The planner takes a PDDL-domain file and problem instances (initial state, goal etc.) as inputs and implements an A^* search algorithm to find all possible optimal sequences of actions leading from initial to the goal state. For the kitting task, a state is defined by a set of *predicates* representing the properties of the problem objects (e.g., location of containers and components) and a set of *functions* which are the properties that have numerical values (e.g., number of unloading locations, number of components etc.).

The performance of any A^* based planner depends on the heuristic function used, which provides a lower bound on the distance from current state to the goal (e.g. using geometric distances in path planning domains). However, in task planning problems like the kitting task, it is difficult to formulate a measure of *closeness* of a state to the goal. This, coupled with a large action and state space (200-400 grounded actions and over 4 million valid states), makes the problem of determining the optimal plan in our kitting task quite challenging. Therefore, we implemented a domain-specific heuristic which calculates a lower-bound on the cost of the optimal plan by taking into account the features of the problem. The complete task is decomposed into sub-tasks (moving containers, unloading and arranging components). The heuristic value is determined based on the minimum number of actions required to achieve the goal specifications (e.g., the minimum number of containers to be moved or components to be placed), while ensuring that the calculated value is admissible to maintain optimality.

C. User Study

The data used to validate the proposed approach was obtained through an online user study. The study was approved by the Office of Research Ethics at the University of Waterloo. Participants were recruited through Amazon Mechanical Turk (MTurk). Participation in this study took approximately one hour and was remunerated up to \$7.25 USD. A total of 35 participants took part in the study, with 28 completing all tasks. The results and analyses presented in this paper only take into account the latter.

Participants were asked to complete 7 distinct instances of a kitting task. These instances were selected from a pool of randomly generated candidates to include different numbers of containers, components, and assembling resources, i.e., platforms and unloading locations, as well as distinct difficulty levels. The complexity of each instance, i.e., easy, medium or hard, was first assessed using task characteristics obtained from the planner such as the total cost of optimal solution, number of states explored and branching factor. A pilot test in which users' performance was compared to the known optimal solution was used to confirm the initial complexity ranking obtained from the planner. Task complexity of all kitting instances is summarized in I.

The user study was comprised of two stages: training and testing. During the training stage, each participant was shown a short video that explained the main actions and features of the simulation. They were also asked to complete a training kitting task (task 1). Upon completion of this

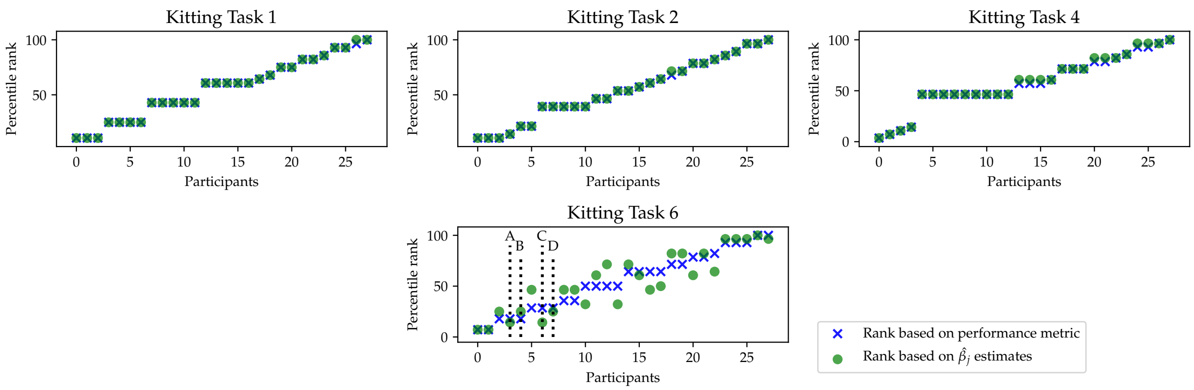


Fig. 2: Comparison of participants' percentile ranking based on the total number of actions (blue crosses) and β_j coefficients (green dots). β_j coefficients were estimated using full length observations. Dotted vertical lines in the task 6 plot highlight participants whose sequences of actions are discussed in detail in Sec. VI

.

stage, participants were asked to complete 6 different kitting instances of varying complexity. They completed a first set with tasks of easy, medium and hard complexity, followed of a second set with the same complexity order. All participants were shown the same sequence of task instances. Participants were tasked with completing all kitting instances with the minimum number of actions. For each task instance, all participants' actions were logged.

V. Results

We apply empirical Bayesian estimation of expertise on the data obtained from our user study. We consider $\beta_j \in \{0.5, 0.6, 0.7, \dots, 5\}$ as the discrete set of possible expertise coefficients. This set of values was initially proposed in [11] and later used in [12]. Additionally, given the size and configuration of the tasks we are considering, the $\beta = 0.5$ $\beta = 5$ values result in probabilities of taking optimal actions that vary between $33\% - 50\%$ and $93\% - 99\%$ respectively. A discretized normal prior ($\mu = 2.75, \sigma = 1$) over this set was used. This prior allows us to capture that almost all participants chose at least one or two non-optimal actions during a task, and that although some participants made more non-optimal choices than others, they did not act entirely at random. The likelihood of a sequence of actions executed by the j-th participant on the i-th task, O_j^i, given all possible β_j coefficients is computed using Eq. 5. The $Q*(s_t, a_t)$ values for all actions and states are obtained from the planner. Maximum a posteriori estimates are taken as the predicted $\hat{\beta}_j^i$ expertise coefficient of the j-th participant on the i-th kitting task.

Three different analyses were conducted to evaluate the proposed expertise estimate. First, we validate the accuracy of the β coefficients against a population-based estimate. Second, we study the impact of the number of observed actions in the accuracy of the β estimates. Last, we investigate whether there are differences between the number of observations needed to identify novices and experts, and how these numbers can vary according among tasks of different complexity.

A. Comparison With Other Empirical Expertise Estimates

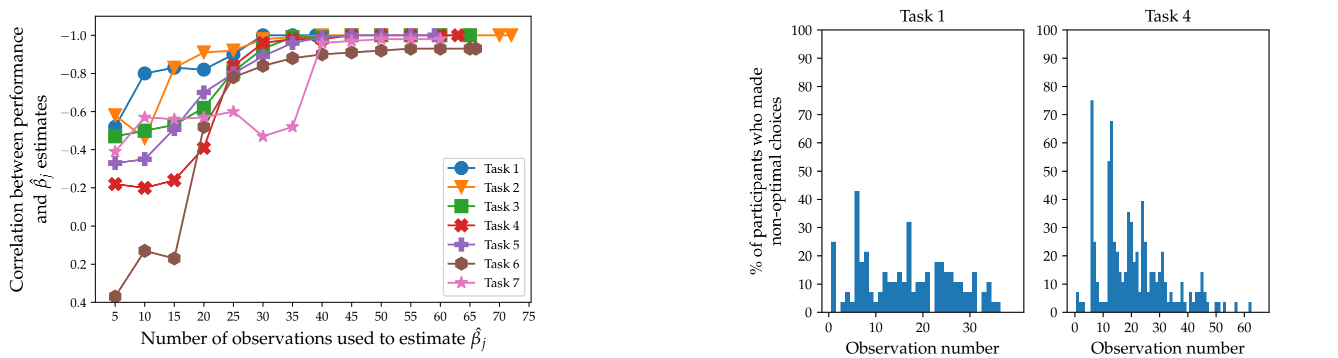
We use the expertise level relative to the population as a baseline against which we measure the quality of the $\hat{\beta}_j^i$ expertise estimates. Since participants were asked to complete each kitting task with the minimum number of actions, we use the total number of actions taken by each participant at each task as a metric of their performance. Participants are ranked based on this metric and their expertise corresponds to their percentile rank within the population sample. The results of the comparison between the performance-based percentile ranking and estimated $\hat{\beta}_j^i$ expertise coefficients for a subset of task instances are shown in 2. Full length observations were used during estimation.

Overall, we find that the estimated expertise coefficients follow closely the baseline expertise ranking. Participants with a low number of actions were predicted to have a high $\hat{\beta}_j^i$ coefficient. Similarly, participants with a high number of actions were predicted to have a low $\hat{\beta}_j^i$. This close relation between the baseline performance-based expertise ranking and the estimated $\hat{\beta}_j^i$ coefficients is further supported by strong *Spearman* correlation coefficients (ρ), with $p-$value ≤ 0.01, between both expertise estimates (see I).

TABLE I: Correlation coefficient between baseline expertise estimate and predicted $\hat{\beta}_j^i$ coefficients

Task Num.	Task Complexity	Coefficient (ρ)
1	Easy	-1.0
2	Easy	-1.0
3	Medium	-1.0
4	Hard	-0.998
5	Easy	-0.999
6	Medium	-0.929
7	Hard	-0.975

We observe also that task 6 has a lower correlation coefficient than the other tasks. This might be explained by



(a) *Spearman* correlation coefficients between $\hat{\beta}^i_j$ estimates and observed performance at different observations lengths.

(b) Percentage of participants who made non-optimal choices along Task 1 and 4

Fig. 3: Correlation coefficients (all tasks) and percentage of participants who made non-optimal (task 1 and 4) measured at different observations lengths.

differences between the type of non-optimal choices made by participants and when those choices were made. Given our current formulation, making mistakes is more costly than taking redundant actions and non-optimal choices made at the end of the task have a stronger influence on the resulting $\hat{\beta}^i_j$ estimates than those made at the beginning of the task. The first point is illustrated by participants A and B (see task 6 plot in 2). Both participants made 4 non-optimal choices in total; while A made 2 mistakes and 2 redundant actions, B took 3 redundant actions and made 1 mistake. The second point is illustrated by participants C and D. Both participants made 5 non-optimal choices in total, however contrary to C, D made all those choices at the beginning of the task.

Compared to previous work, the proposed method results in higher correlation coefficients, -0.985 in average, than the ones reported by [2], 0.412 and 0.427. Although our results are not directly comparable to [3], it is important to notice that, contrary to their work, the proposed approach does not require predefined performance metrics or several task executions in order to determine a participant's expertise level.

B. Impact of Number of Observations

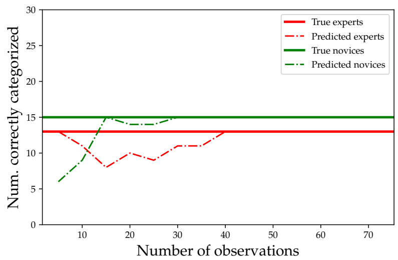
Since the proposed expertise estimates can be computed in an incremental manner, we next investigate the impact that the number of observations available during inference has on these estimates. To do so, we employ the same Bayesian empirical estimation procedure previously outlined. However, we limit the number of observations used to calculate the posterior distribution to different observation lengths. For each participant, we consider the first k observations seen at any given task, where $k \in \{5, 10, 15, \ldots, T^i_j\}$ and T^i_j indicates the total number of actions the *j*-th participant took during the *i*-th kitting task.

We use *Spearman* correlation coefficients (ρ) between the predicted $\hat{\beta}^i_j$ estimates and the chosen performance metric, i.e., total number of actions taken to complete each task, as our measure of accuracy. The results are shown in Figure 3a. Two main tendencies are observed. On the one hand, we observe that four of the seven tasks, i.e., task 1, 2, 3, and 7, registered moderate correlation coefficients, i.e., $-0.4 \leq \rho \leq -0.6$ with p-value ≤ 0.05, after only 5 to 10 actions have been seen during inference. Tasks 4, 5 and 6 required, however, between 15 and 20 observations before reaching similar correlation values. On the other hand, we observe that for almost all tasks, strong correlation coefficients, i.e., $\rho >= -0.7$ with p-value ≤ 0.05, were obtained after seeing at least 25 observations. Task 7, however, required 15 additional observations.

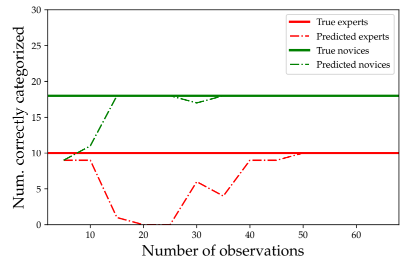
These differences in the number of observations needed to reach both moderate and strong correlation coefficients can be explained by how many participants made non-optimal choices and when these choices were made during a task. To illustrate this, we compare the distribution of the percentage of participants that made non-optimal choices along task 1 and task 4 (see 3b). In the case of task 1, we observe that: **i)** between $5\% - 45\%$ of the total number of participants made non-optimal choices within their first 10 actions, and **ii)** most of these non-optimal choices were made by participants with the highest number of total actions. On the contrary, with task 4, we observe that: **i)** almost all participants made few or non optimal choices at the beginning of the task, and **ii)** between 5% and 75% of all participants made non-optimal choices during the following 10 to 15 observations. Hence, compared to task 1, in task 4 the proposed expertise estimation approach required more observations in order to differentiate the participants who made errors only during those first 25 observations from those who took more non-optimal actions later on in the task. This pattern was reflected on the low correlation registered during the first 20 observations. The same pattern was observed for tasks 6 and 7.

C. Relation Between Task Complexity, Observation Length and β_j Estimates

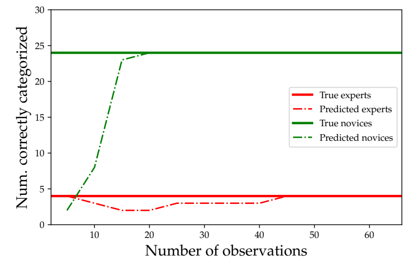
Our previous analysis showed that for almost all kitting tasks we needed between 20 and 30 observations before obtaining reasonable expertise estimates. However, it might be possible that the number of observations required to obtain



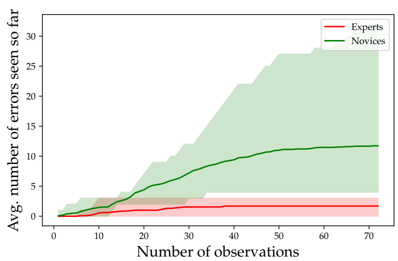
(a) Accuracy rates for task 2 (easy) at different observation lengths



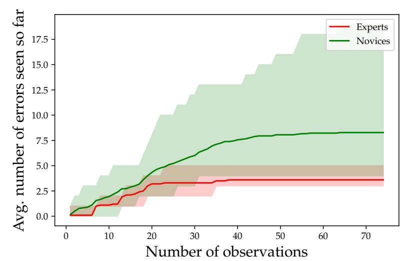
(b) Accuracy rates for task 3 (medium) at different observation lengths



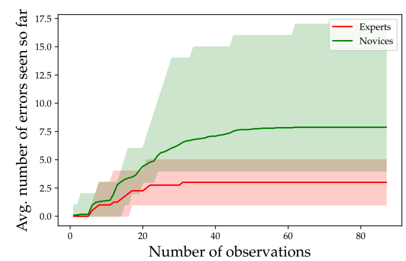
(c) Accuracy rates for task 4 (hard) at different observation lengths



(d) Avg. number of non-optimal actions for all expertise groups along task 2



(e) Avg. number of non-optimal actions for all expertise groups along task 3



(f) Avg. number of non-optimal actions for all expertise groups along task 4

Fig. 4: Evolution of expertise accuracy and avg. number of non-optimal actions observed for different expertise groups, i.e., experts, novices and intermediates, along tasks of varying complexity.

these estimates differ among expertise groups, i.e., experts and novices, and tasks of different complexity. For instance, we expect that: **i)** the identification of experts might require more observations than that of the novices, since we need to make sure that none or few non-optimal choices were made along the task, and **ii)** more complex[2] tasks might allow us to identify novices faster than easier tasks, since novices would be prone to make more mistakes during the former.

To measure the impact of these two factors on the number of observations required to obtain reasonable expertise estimates, participants were separated into two expertise groups: experts and novices. Groups were defined in a similar manner to the one described in [9]. Participants whose total number of actions was $\sigma = 0.6$ standard deviation below the mean were considered to be experts. The remaining participants were considered as novices. A classification threshold between both groups, β_{experts}, was determined based on the estimated β_j coefficients obtained after using full length observations. Expertise groups and classification thresholds were defined for each kitting task. The expertise group, g_j^i, of the j-th participant after observing their first k actions on the i-th tasks was determined as follows

$$g_j^i = \begin{cases} \text{expert}, & \text{if } \beta_j \geq \beta_{\text{experts}} \\ \text{novice}, & \text{otherwise.} \end{cases} \tag{8}$$

The total number of participants correctly assigned to each group was used as the accuracy metric. We present results for a subset of the kitting tasks, one at each complexity level, in 4a to 4c. We also present the evolution of the number of errors made by participants on each expertise group along these tasks (see 4d to 4f). Overall, we observe that as task complexity increases, the number of observations required to separate novices from experts decreases. While for the easiest task (task 2) between 10 and 30 observations were needed to fully identify all novices, for the most complex one (task 4), similar accuracy was obtained after only 20 observations. We also found that, compared to novices, experts required more observations before being fully identified.

These results as well as the number of observations required to accurately identify all expertise groups are closely related to the distribution of non-optimal actions along each task (see 4d to 4f). We found that as soon as the number of non-optimal choices made by both groups deviates from each other, 100% accuracy can be obtained. For example, in task 2, the overlap between the number of non-optimal choices made by experts and novices ends after approximately 35 actions have been observed. Hence, between 35 and 40 observations are required to fully identify the participants of each expertise group. The same pattern is observed in tasks 3 and 4.

VI. Discussion and Conclusion

In this paper, we propose a method that incrementally estimates a user's expertise level by evaluating how optimal the action choices made by a user are. A key aspect of the proposed method is the ability to quantify the *goodness* of all actions available at any given state visited by the user. Contrary to [6], where an MDP problem must be solved for each task and user, the proposed method benefits from the use of an optimal planner as an oracle that provides accurate estimates of any action's value. By doing so, we are able

[2]Task complexity was assigned as described in Sec. IV-C

to test the proposed method in a task scenario that expands over 400 possible actions and over 4 million valid states. Also note that while the solving time of such problems does not scale well with the number of states, optimal plans can be computed offline because of the structure of a kitting task (deterministic actions and fully observable states).

We evaluated the proposed approach on 7 instances of a simulated kitting task. We compared our expertise estimates against a baseline obtained from the overall population's performance (i.e., total number of actions taken to complete a task) for each task. Our results indicate that our estimates of expertise are good indicators of observed performance. Note that the proposed approach could also have been used to evaluate whether someone is an overall expert or novice at kitting (rather than on each task). This would be done by setting the posterior probability distribution of the previous task as the prior of the next task.

We also evaluated the number of actions required during inference in order to obtain accurate expertise estimates. Our results indicate that, for the type of task we consider, observation lengths between 20 and 25 actions are sufficient to obtain β_j estimates that strongly correlate with final task performance. We also find that identifying a novice requires fewer observations than identifying an expert. As a possible application of this early detection, we could envision a robot that guides the user through the best solution plan as soon as it determines the user is a novice. Such an interaction, in the context of the simulated kitting tasks, would result in a performance improvement of 24.4% in average. Finally, we observe that the observation length required to identify novices decreases with task complexity. This offers pointers towards active-learning settings in which a robot can propose to its human partner to work on a more complex task in order to quickly determine their expertise level.

A limitation of the work we present in this paper are two assumptions that **i)** the planner provides informed and correct estimates of any action's value, and **ii)** the human and the robot share a similar understanding of what makes some actions more valuable than others. Thus, when our estimates determine that a participant has a low expertise coefficient it is because they are often choosing in a poor manner. However, in a real case scenario, it might happen that the robot's (and planner) representation of the task does not completely match the task's current state and hence actions that are judged as sub-optimal are in fact optimal. Similarly, it might happen that the user's understanding of the dynamics of the environment and task does not match that of the robot, and the user's sub-optimal behavior is due to this mismatch rather than due to the user's inability to identify optimal actions [21]. As future work, we will extend the current model so the robot can further reason about the significance of a poor β estimate and act accordingly.

References

[1] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita, "Adapting robot behavior for human-robot interaction," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 911–916, 2008.

[2] L. Milliken and G. A. Hollinger, "Modeling user expertise for choosing levels of shared autonomy," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2285–2291.

[3] N. Enayati, G. Ferrigno, and E. De Momi, "Skill-based human–robot cooperation in tele-operated path tracking," *Autonomous Robots*, pp. 1–13, 2018.

[4] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction*. ACM, 2015, pp. 189–196.

[5] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," *Urbana*, vol. 51, no. 61801, pp. 1–4.

[6] M. M. LaMar, "Markov decision process measurement model," *Psychometrika*, vol. 83, no. 1, pp. 67–88, 2018.

[7] G. Megali, S. Sinigaglia, O. Tonet, and P. Dario, "Modelling and evaluation of surgical performance using hidden markov models," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 10, pp. 1911–1919, 2006.

[8] C. Torrey, A. Powers, M. Marge, S. R. Fussell, and S. Kiesler, "Effects of adaptive robot dialogue on information exchange and social relations," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. ACM, 2006, pp. 126–133.

[9] J. C. Huegel, O. Celik, A. Israr, and M. K. O'Malley, "Expertise-based performance measures in a virtual training environment," *Presence*, vol. 18, no. 6, pp. 449–467, Dec 2009.

[10] M. C. Desmarais and R. S. Baker, "A review of recent advances in learner and skill modeling in intelligent learning environments," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1-2, pp. 9–38, 2012.

[11] C. Baker, R. Saxe, and J. Tenenbaum, "Bayesian theory of mind: Modeling joint belief-desire attribution," in *Proceedings of the annual meeting of the cognitive science society*, vol. 33, no. 33, 2011.

[12] A. N. Rafferty, M. M. LaMar, and T. L. Griffiths, "Inferring learners' knowledge from their actions," *Cognitive Science*, vol. 39, no. 3, pp. 584–618, 4 2015.

[13] J. F. Fisac, A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, S. Wang, C. J. Tomlin, and A. D. Dragan, "Probabilistically safe robot planning with confidence-based human predictions," *CoRR*, vol. abs/1806.00109, 2018.

[14] A. Bobu, A. Bajcsy, J. F. Fisac, and A. D. Dragan, "Learning under misspecified objective spaces," *CoRR*, vol. abs/1810.05157, 2018.

[15] S. Nikolaidis, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in collaborative tasks: Models and experiments," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 618–634, 2017.

[16] D. C. Dennett, *The intentional stance*. MIT press, 1989.

[17] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2018, pp. 307–315.

[18] M. Chen, D. Hsu, and W. S. Lee, "Guided exploration of human intentions for human-robot interaction," *CoRR*, vol. abs/1812.04728, 2018.

[19] A. G. Banerjee, A. Barnes, K. N. Kaipa, J. Liu, S. Shriyam, N. Shah, and S. K. Gupta, "An ontology to enable optimized task partitioning in human-robot collaboration for warehouse kitting operations," in *Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX*, vol. 9494. International Society for Optics and Photonics, 2015, p. 94940H.

[20] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "Pddl—the planning domain definition language."

[21] S. Reddy, A. D. Dragan, and S. Levine, "Where do you think you're going?: Inferring beliefs about dynamics from behavior," *arXiv preprint arXiv:1805.08010*, 2018.