Learning and Reasoning for Robot Sequential Decision Making under Uncertainty

Saeid Amiri,[1] **Mohammad Shokrolah Shirazi,**[2] **Shiqi Zhang**[1]
[1]SUNY Binghamton, Binghamton, NY 13902 USA
[2]University of Indianapolis, Indianapolis, IN 44227 USA
samiri1@binghamton.edu; shirazim@uindy.edu; szhang@cs.binghamton.edu

Abstract

Robots frequently face complex tasks that require more than one action, where sequential decision-making (SDM) capabilities become necessary. The key contribution of this work is a robot SDM framework, called LCORPP, that supports the simultaneous capabilities of supervised learning for passive state estimation, automated reasoning with declarative human knowledge, and planning under uncertainty toward achieving long-term goals. In particular, we use a hybrid reasoning paradigm to refine the state estimator, and provide informative priors for the probabilistic planner. In experiments, a mobile robot is tasked with estimating human intentions using their motion trajectories, declarative contextual knowledge, and human-robot interaction (dialog-based and motion-based). Results suggest that, in efficiency and accuracy, our framework performs better than its no-learning and no-reasoning counterparts in office environment.

1 Introduction

Mobile robots have been able to operate in everyday environments over extended periods of time, and travel long distances that have been impossible before, while providing services, such as escorting, guidance, and delivery (Hawes et al. 2017; Veloso 2018; Khandelwal et al. 2017). Sequential decision-making (SDM) plays a key role toward robot long-term autonomy, because real-world domains are stochastic, and a robot must repeatedly estimate the current world state and decide what to do next.

There are at least three AI paradigms, namely *supervised learning*, *automated reasoning*, and *probabilistic planning*, that can be used for robot SDM. Each of the three paradigms has a long history with rich literature. *However, none of the three completely meet the requirements in the context of robot* SDM. First, a robot can use supervised learning to make decisions, e.g., to learn from the demonstrations of people or other agents (Argall et al. 2009). However, the methods are not designed for reasoning with declarative contextual knowledge that are widely available in practice. Second, knowledge representation and reasoning (KRR) methods can be used for decision making (Gelfond and Kahl 2014). However, such knowledge can hardly be comprehensive in practice, and robots frequently find it difficult to survive from inaccurate or outdated knowledge. Third, probabilistic planning methods support active information collection for goal achievement, e.g., using decision-theoretic frameworks such as Markov decision processes (MDPs) (Puterman 2014) and partially observable MDPs (POMDPs) (Kaelbling, Littman, and Cassandra 1998). However, the planning frameworks are ill-equipped for incorporating declarative contextual knowledge.

In this work, we develop a robot SDM framework that enables the simultaneous capabilities of learning from past experiences, reasoning with declarative contextual knowledge, and planning toward achieving long-term goals. Specifically, we use *long short-term memory* (LSTM) (Hochreiter and Schmidhuber 1997) to learn a classifier for passive state estimation using streaming sensor data, and use *commonsense reasoning and probabilistic planning* (CORPP) (Zhang and Stone 2015) for active perception and task completions using contextual knowledge and human-robot interaction. Moreover, the dataset needed for supervised learning can be augmented through the experience of active human-robot communication, which identifies the second contribution of this work. The resulting algorithm is called learning-CORPP (LCORPP), as overviewed in Figure 1.

We apply LCORPP to the problem of *human intention estimation* using a mobile robot. The robot can passively estimate human intentions based on their motion trajectories, reason with contextual knowledge to improve the estimation, and actively confirm the estimation through human-robot interaction (dialog-based and motion-based). Results suggest that, in comparison to competitive baselines from the literature, LCORPP significantly improves a robot's performance in state estimation (in our case, human intention) in both accuracy and efficiency. [1]

2 Related Work

KRR and SDM: SDM algorithms can be grouped into two classes depending on the availability of the world model, namely probabilistic planning, and reinforcement learning (RL). Next, we select a sample of the SDM algorithms, and

[1] A demo video is available: https://youtu.be/YgiB1fpJgmo

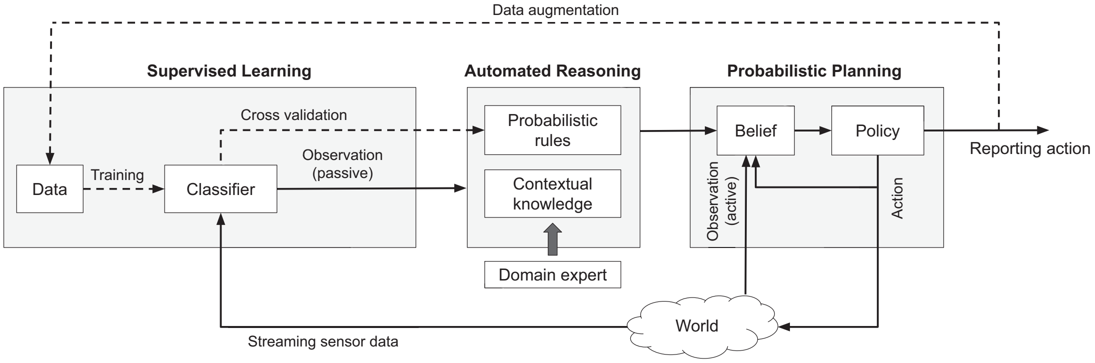


Figure 1: An overview of LCORPP that integrates supervised learning, automated reasoning, and probabilistic planning. Streaming sensor data, e.g., from RGB-D sensors, is fed into an offline-trained classifier. The classifier's output is provided to the reasoner along with classifier's cross-validation. This allows the reasoner to encode uncertainty of the classifier's output. The reasoner reasons with declarative contextual knowledge provided by the domain expert, along with the classifier's output and accuracies in the form of probabilistic rules, and produces a prior belief distribution for the probabilistic planner. The planner suggests actions to enable the robot to actively interact with the world, and determines what actions to take, including when to terminate the interaction and what to report. At each trial, the robot collects the information gathered from the interaction with the world. In case of limited experience for training, LCORPP supports data augmentation through actively seeking human supervision. Solid and dashed lines correspond to Algorithms 1 and 2 respectively, as detailed in Section 4.

make comparisons with LCORPP.

When world model is unavailable, SDM can be realized using RL. People have developed algorithms for integrating KRR and RL methods (Leonetti, Iocchi, and Stone 2016; Yang et al. 2018; Lyu and others 2019; Sridharan and Rainge 2014; Griffith, Subramanian, and others 2013; Illanes et al. 2019). Among them, Leonetti, Iocchi, and Stone used declarative action knowledge to help an agent to select only the reasonable actions in RL exploration. Yang et al. developed an algorithm called PEORL that integrates hierarchical RL with task planning, and this idea was applied to deep RL by Lyu and others in 2019.

When world models are provided beforehand, one can use probabilistic planning methods for SDM. Contextual knowledge and declarative reasoning have been used to help estimate the current world state in probabilistic planning (Zhang, Sridharan, and Wyatt 2015; Zhang and Stone 2015; Sridharan et al. 2019; Ferreira et al. 2017; Chitnis, Kaelbling, and Lozano-Pérez 2018; Lu et al. 2018; 2017). Work closest to this research is the CORPP algorithm, where hybrid reasoning (both logical and probabilistic) was used to guide probabilistic planning by calculating a probability for each possible state (Zhang and Stone 2015). More recently, researchers have used human-provided declarative information to improve robot probabilistic planning (Chitnis, Kaelbling, and Lozano-Pérez 2018).

The main difference from the algorithms is that LCORPP is able to leverage the extensive data of (labeled) decision-making experiences for continuous passive state estimation. In addition, LCORPP supports collecting more data from the human-robot interaction experiences to further improve the passive state estimator over time.

KRR and supervised learning: Reasoning methods have been incorporated into supervised learning in natural language processing (NLP) (Chen, Tan, and others 2019; Zellers, Holtzman, and others 2019) and computer vision (Zellers et al. 2019; Aditya et al. 2019; Chen et al. 2018) among others. For instance, Chen, Tan, and others in 2019 used commonsense knowledge to add missing information in incomplete sentences (e.g., to expand "pour me water" by adding "into a cup" to the end); and Aditya et al. used spatial commonsense in the Viual Question Answering (VQA) tasks. Although LCORPP includes components for both KRR and supervised learning, we aim at a SDM framework for robot decision-making toward achieving long-term goals, which identifies the key difference between LCORPP and the above-mentioned algorithms.

SDM and supervised learning: Researchers have developed various algorithms for incorporating human supervision into SDM tasks (Taylor and Borealis 2018; Amiri et al. 2018; Thomaz, Breazeal, and others 2006). Among them, Amiri et al. used probabilistic planning for SDM under mixed observability, where the observation model was learned from annotated datasets. Taylor and Borealis surveyed a few ways of improving robots' SDM capabilities with supervision (in the form of demonstration or feedback) from people. In comparison to the above methods, LCORPP is able to leverage human knowledge, frequently in declarative forms, toward more efficient and accurate SDM.

To the best of our knowledge, this is the first work on robot SDM that simultaneously supports supervised learning for passive perception, automated reasoning with contextual knowledge, and active information gathering toward achieving long-term goals under uncertainty.

3 Background

In this section, we briefly summarize the three computational paradigms used in the buildings blocks of LCORPP.

LSTM: Recurrent neural networks (RNNs) (Hopfield 1982) are a kind of neural networks that use their internal state (memory) to process sequences of inputs. Given the input sequence vector $(x_0, x_1, ..., x_n)$, at each time-step, a hidden state is produced that results in the hidden sequence of $(h_0, h_1, ..., h_n)$. LSTM (Hochreiter and Schmidhuber 1997) network, is a type of RNN that includes LSTM units. Each memory unit in the LSTM hidden layer has three gates for maintaining the unit state: input gate defines what information is added to the memory unit; output gate specifies what information is used as output; and forget gate defines what information can be removed. LSTMs use memory cells to resolve the problem of vanishing gradients, and are widely used in problems that require the use of long-term contextual information, e.g., speech recognition (Graves, Mohamed, and Hinton 2013) and caption generation (Vinyals et al. 2015). We use LSTM-based supervised learning for passive state estimation with streaming sensor data in this work.

P-log: Answer Set Prolog (ASP) is a logic programming paradigm that is strong in non-monotonic reasoning (Gelfond and Kahl 2014; Lifschitz 2016). An ASP program includes a set of rules, each in the form of:

$$l_0 \leftarrow l_1,\ \cdots,\ l_n,\ \text{not}\ l_k,\ \cdots,\ \text{not}\ l_{n+k}$$

where l's are literals that represent whether a statement is true or not, and symbol *not* is called default negation. The right side of a rule is the *body*, and the left side is the *head*. A rule head is true if the body is true.

P-log extends ASP by allowing probabilistic rules for quantitative reasoning. A P-log program consists of the logical and probabilistic parts. The logical part inherits the syntax and semantics of ASP. The probabilistic part contains *pr-atoms* in the form of:

$$pr_r(G(\eta) = w | B) = v$$

where $G(\eta)$ is a random variable, B is a set of literals and $v \in [0, 1]$. The pr-atom states that, if B holds and experiment *r* is fixed, the probability of $G(\eta) = w$ is v.

POMDPs: Markov decision processes (MDPs) can be used for sequential decision-making under full observability. Partially observable MDPs (POMDPs) (Kaelbling, Littman, and Cassandra 1998) generalize MDPs by assuming partial observability of the current state. A POMDP model is represented as a tuple $(S, A, T, R, Z, O, \gamma)$ where S is the state-space, A is the action set, T is the state-transition function, R is the reward function, O is the observation function, Z is the observation set and γ is discount factor that determines the planning horizon.

An agent maintains a belief state distribution b with observations ($z \in Z$) using the Bayes update rule:

$$b'(s') = \frac{O(s', a, z) \sum_{s \in S} T(s, a, s') b(s)}{pr(z|a, b)}$$

where s is the state, a is the action, $pr(z|a, b)$ is a normalizer, and z is an observation. Solving a POMDP produces a policy that maps the current belief state distribution to an action toward maximizing long-term utilities.

CORPP (Zhang and Stone 2015) uses P-log (Baral, Gelfond, and Rushton 2009) for knowledge representation and reasoning, and POMDPs (Kaelbling, Littman, and Cassandra 1998) for probabilistic planning. Reasoning with a P-log program produces a set of possible worlds, and a distribution over the possible worlds. In line with the CORPP work, we use P-log for reasoning purposes, while the reasoning component is not restricted to any specific declarative language.

4 Framework

In this section, we first introduce a few definitions, then focus on LCORPP, our robot SDM framework, and finally present a complete instantiation of LCORPP in detail.

Definitions: Before describing the algorithm, it is necessary to define three variable sets of $\mathbf{V}^{lrn}$, $\mathbf{V}^{rsn}$ and $\mathbf{V}^{pln}$ that are modeled in the learning, reasoning, and planning components respectively. For instance, $\mathbf{V}^{lrn}$ includes a finite set of variables:

$$\mathbf{V}^{lrn} = \{V_0^{lrn}, V_1^{lrn}, ...\}$$

We consider factored spaces, so the three variable sets can be used to specify the three state spaces respectively, i.e., S^{lrn}, S^{rsn} and S^{pln}. For instance, the learning component's state space, S^{lrn}, can be specified by $\mathbf{V}^{lrn}$, and includes a finite set of states in the form of:

$$S^{lrn} = \{s_0^{lrn}, s_1^{lrn}, ...\}$$

Building on the above definitions, we next introduce the LCORPP algorithm followed by a complete instantiation.

4.1 Algorithms

We present LCORPP in the form of Algorithms 1 and 2, where Algorithm 1 calls the other.

The input of LCORPP includes the dataset Ω for sequence classification, declarative rules θ, logical facts β, and a POMDP model $\mathcal{M}$. Each sample in Ω is a matrix of size $T \times N$, where T is the time length and N is the number of features. Each sample is associated with a label, where each label corresponds to state $s^{lrn} \in S^{lrn}$. Logical-probabilistic rules, θ, are used to represent contextual knowledge from human experts. Facts, β, are collected at runtime, e.g., current time and location, and are used together with the rules for reasoning. Finally, POMDP model $\mathcal{M}$ includes world dynamics and is used for planning under uncertainty toward active information gathering and goal accomplishments.

Algorithm 1 starts with training classifier ρ using dataset Ω, and confusion matrix C that is generated by cross-validation. The probabilities in C are passed to the reasoner to update probabilistic rules θ. Action policy π is then generated using the POMDP model $\mathcal{M}$ and off-the-shelf solvers from the literature. Matrix $\hat{C}$ (of shape C) and counter c are initialized with uniform distribution and 0 respectively.

Algorithm 1 LCORPP

```
Require: A set of instance-label pairs $\Omega$ (training dataset),
    Logical-probabilistic rules $\theta$, Termination probability thresh-
    old $\epsilon$, POMDP model $\mathcal{M}$, Batch size $K$
 1: Compute classifier $\rho$ using $\Omega$
 2: $C \leftarrow$ CROSS-VALIDATE($\rho$), where $C$ is a confusion matrix
 3: Update rules in $\theta$ with the probabilities in $C$
 4: Compute action policy $\pi$ with $\mathcal{M}$ (using POMDP solvers)
 5: Initialize $\hat{C}$ (same shape of $C$) using uniform distributions
 6: Initialize counter: $c \leftarrow 0$
 7: while SSUB($\hat{C} - C$) $> \epsilon$ do
 8:     Collect new instance $I$
 9:     $L \leftarrow$ DT-CONTROL($\rho, I, \mathcal{M}$), where $L$ is the label of $I$
10:     Store instance-label pair: $\omega \leftarrow (I, L)$
11:     $\Omega \leftarrow \Omega \cup \omega$
12:     $c \leftarrow c + 1$
13:     if $c == K$ then
14:         Compute classifier $\rho$ using augmented dataset $\Omega$
15:         $\hat{C} \leftarrow$ CROSS-VALIDATE($\rho$)
16:         Update rules in $\theta$ with the probabilities in $\hat{C}$
17:         $C \leftarrow \hat{C}$
18:         $c \leftarrow 0$
19:     end if
20: end while
```

Algorithm 2 DT-CONTROL: Decision Theoretic Control

```
Require: Classifier $\rho$, Instance $I$, and POMDP model $\mathcal{M}$
 1: Update state: $s^{lrn} \leftarrow \rho(I)$, where $s^{lrn} \in S^{lrn}$
 2: Collect facts $\beta$ from the world, and add $s^{lrn}$ into $\beta$
 3: $\hat{\mathbf{V}}^{rsn} \leftarrow \{\hat{V} | \hat{V} \in \mathbf{V}^{rsn}$, and $\hat{V} \in \mathbf{V}^{pln}\}$
 4: Use $\hat{\mathbf{V}}^{rsn}$ to form the state space of $\hat{S}^{rsn}$, where $\hat{S}^{rsn} \subset S^{rsn}$
    and $\hat{S}^{rsn} \subset S^{pln}$
 5: Reason with $\theta$ and $\beta$ to compute belief $b^{rsn}$ over $S^{rsn}$
 6: Compute $\hat{b}^{rsn}$ (over $\hat{S}^{rsn}$), i.e., a marginal of $b^{rsn}$
 7: Initialize belief $b^{pln}$ (over $S^{pln}$) using $\hat{b}^{rsn}$
 8: repeat
 9:     Select action $a \leftarrow \pi(b^{pln})$, and execute $a$
10:     Make observation $o$
11:     Update $b^{pln}$ based on $a$ and $o$ using Bayes update rule
12: until reaching terminal state $term \in S^{pln}$
13: $L \leftarrow$ EXTRACTLABEL($a$)
```

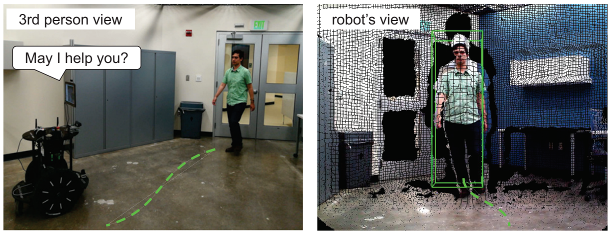


Figure 2: Robot estimating human intention, e.g., human intending to interact or not, by analyzing human trajectories, reasoning with contextual knowledge (such as location and time), and taking human-robot interaction actions.

SSUB($\hat{C} - C$) is a function that sums up the absolute values of the element-wise subtraction of matrices $\hat{C}$ and C. As long as the output of SSUB is greater than the termination threshold ϵ, the robot collects a new instance I, passes the instance along with the classifier and the model to Algorithm 2 to get label L (Lines 8-9). The pair of instance and label, ω, is added to the dataset Ω and the counter (c) is incremented. If c reaches the batch size K, new classifier ρ is trained using the augmented dataset. Then, it is cross-validated to generate $\hat{C}$, and the rules θ are updated. Also, c is set to 0 for collection of a new batch of data (Lines 13-18).

Algorithm 2 requires classifier ρ, data instance I, and POMDP model $\mathcal{M}$. The classifier (ρ) outputs the learner's current state, $s^{lrn} \in S^{lrn}$. This state is then merged into β in Line 2, which is later used for reasoning purposes. A set of variables, $\hat{\mathbf{V}}^{rsn}$, is constructed in Line 3 to form state space $\hat{S}^{rsn}$, which is a partial state space of both S^{rsn} and S^{pln}. b^{rsn} is the reasoner's posterior belief. Belief distribution $\hat{b}^{rsn}$ over $\hat{S}^{rsn}$ bridges the gap between LCORPP's reasoning and planning components: $\hat{b}^{rsn}$ is computed as a marginal distribution of the reasoner's output in Line 6; and used for generating the prior distribution of b^{pln} for active interactions. LCORPP initializes POMDP prior belief b^{pln} over the state set S^{pln} with $\hat{b}^{rsn}$ in Line 7, and uses policy π to map b^{pln} to actions. This sense-plan-act loop continues until reaching the terminal state and the estimation label would be extracted from the reporting action (described in detail in subsection 4.2).

4.2 Instantiation

We apply our general-purpose framework to the problem of human intention estimation using a mobile robot, as shown in Figure 2. The robot can observe human motion trajectories using streaming sensor data (RGB-D images), has contextual knowledge (e.g., visitors tend to need guidance help), and is equipped with dialog-based and motion-based interaction capabilities. The objective is to determine human intention (in our case, whether a human is interested in interaction or not). In the following subsections, we provide technical details of a complete LCORPP instantiation in this domain.

Learning for Perception with Streaming Data In order to make correct state estimation based on the streaming sensor data while considering the dependencies at various time steps, we first train and evaluate the classifier ρ using dataset Ω. We split the dataset into training and test sets, and produce the confusion matrix C, which is later needed by the reasoner. Human intention estimation is modeled as a classification problem for the LSTM-based learner:

$$s^{lrn} = \rho(I)$$

where robot is aiming at estimating $s^{lrn} \in S^{lrn}$ using streaming sensor data I. In our case, streaming data is in the form of people motion trajectories; and there exists only one binary variable, *intention* $\in \mathbf{V}^{lrn}$. As a result, state set S^{lrn} includes only two states:

$$S^{lrn} = \{s_0^{lrn}, s_1^{lrn}\}$$

where s_0^{lrn} and s_1^{lrn} correspond to the person having intention of interacting with the robot or not. Since the human trajectories are in the form of sequence data, we use LSTM to train a classifier for estimating human intentions with motion trajectories. Details of the classifier training are presented in Section 5. Next, we explain how the classifier output is used for reasoning.

Reasoning with Contextual Knowledge Contextual knowledge, provided by domain experts, can help the robot make better estimations. For instance, in the early mornings of work days, people are less likely to be interested in interacting with the robot, in comparison to the university open-house days. The main purpose of the reasoning component is to incorporate such contextual knowledge to help the passive state estimation.

The knowledge base consists of logical-probabilistic rules θ in P-log (Baral, Gelfond, and Rushton 2009), a declarative language that supports the representation of (and reasoning with) both logical and probabilistic knowledge. The reasoning program consists of random variables set $\mathbf{V}^{rsn}$. It starts collecting facts and generating β. The confusion matrix generated by the classifier's cross-validation is used to update θ. The variables that are shared between the reasoning and planning components are in the set $\hat{\mathbf{V}}^{rsn}$. The reasoner produces a belief b^{rsn} over S^{rsn} via reasoning with θ and β.

In the problem of human intention estimation, the reasoner contains random variables:

$\mathbf{V}^{rsn} = \{\texttt{location}, \texttt{time}, \texttt{identity}, \texttt{intention}, \cdots\}$,

where the range of each variable is defined as below:

```
location: {classroom, library}
time: {morning, afternoon, evening}
identity: {student, professor, visitor}
intention: {interested, not interested}
```

We further include probabilistic rules into the reasoning component. For instance, the following two rules state that the probability of a visitor showing up in the afternoon is 0.7, and the probability of a professor showing up in the library (instead of other places) is 0.1, respectively.

```
pr(time=afternoon|identity=visitor)=0.7.
pr(location=library|identity=professor)=0.1.
```

It should be noted that *time* and *location* are facts that are fully observable to the robot, whereas human *identity* is a latent variable that must be inferred. Time, location, and intention are probabilistically determined by human identity. We use time and location to infer human identity, and then estimate human intention.

The binary distribution over human intention, $\hat{b}^{rsn}$, a marginal distribution of b^{rsn} over $\hat{S}^{rsn}$, is provided to the POMDP-based planner as informative priors.[2]

Active Perception via POMDP-based HRI Robots can *actively* take actions to reach out to people and gather information. We use POMDPs to build probabilistic controllers. A POMDP model can be represented as a tuple $(S^{pln}, A, T, R, Z, O, \gamma)$. We briefly discuss how each component is used in our models:

- $S^{pln} : \hat{S}^{rsn} \times S_l^{pln} \cup \{term\}$ is the state set. $\hat{S}^{rsn}$ includes two states representing human being interested to interact or not. S_l^{pln} includes two states representing whether the robot has turned towards the human or not and $term$ is the terminal state.
- $A : A_a \cup A_r$ is the set of actions. A_a includes both motion-based and language-based interaction actions, including *turning* (towards the human), *greeting*, and *moving forward* slightly. A_r includes two actions for reporting the human being interested in interaction or not.
- $T(s, a, s') = P(s'|s, a)$ is the transition function that accounts for uncertain or non-deterministic action outcomes where $a \in A$ and $s \in S$. Reporting actions deterministically lead to the *term* state.
- $Z = \{pos, neg, none\}$ is the observation set modeling human feedback in human-robot interaction.
- $O(s', a, z) = P(z|a, s')$, where $z \in Z$, is the observation function, which is used for modeling people's noisy feedback to the robot's interaction actions.
- $R(s, a)$ is the reward function, where costs of interaction actions, $a_a \in A_a$, correspond to the completion time. A correct (wrong) estimation yields a big bonus (penalty).

Reporting actions deterministically lead to the $term$ state. We use a discount factor $\gamma = 0.99$ to give the robot a long planning horizon. Using an off-the-shelf solver (e.g., (Kurniawati, Hsu, and Lee 2008)), the robot can generate a behavioral policy that maps its belief state to an action toward efficiently and accurately estimating human intentions.

To summarize, the robot's LSTM-based classifier estimates human intention based on the human trajectories. The reasoner uses human knowledge to compute a distribution on human intention. The reasoner's intention estimation serves as the prior of the POMDP-based planner, which enables the robot to actively interact with people to figure out their intention. The reasoning and planning components of CORPP are constructed using human knowledge, and do not involve learning. The reasoning component aims at correcting and refining the LSTM-based classifier's output, and the planning component is for active perception.

5 Experiments

In this section, we describe the testing domain (including the dataset), experiment setup, and statistical results.

5.1 Dataset and Learning Classifiers

We use a human motion dataset (Kato, Kanda, and Ishiguro 2015) to train the LSTM-based classifier, where the dataset was collected using multiple 3D range sensors mounted overhead in a shopping center environment. Each instance

[2] The reasoning component can be constructed using other logical-probabilistic paradigms that build on first-order logic, such as Probabilistic Soft Logic (PSL) (Bach and others 2017) and Markov Logic Network (MLN) (Richardson and Domingos 2006). In comparison, P-log directly takes probabilistic, declarative knowledge as the input, instead of learning weights with data, and meets our need of utilizing human knowledge in declarative forms.

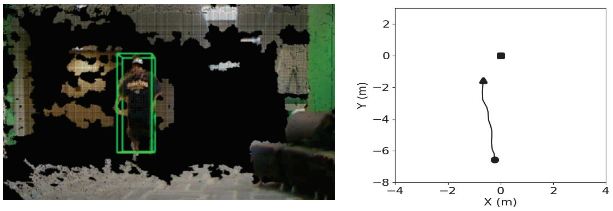


Figure 3: (Left) A human detected and tracked by our Segway-based robot in the classroom building. (Right) The corresponding collected trajectory, where the robot's position is shown in square "□", and the human trajectory starts at dot and finishes in "△" position. In this example, the human was interested in interactions.

in the dataset includes a human motion trajectory in 2D space, and a label of whether the human eventually interacts with the robot or not. There are totally 2286 instances in the dataset, where 63 are positive instances (2.7%). Each trajectory includes a sequence of data fields with the sampling rate of 33 milliseconds. Each data field is in the form of a vector: $(x_i,\ y_i,\ z_i,\ v_i,\ \theta_{m_i},\ \theta_{h_i})$. Index i denotes the timestep. x_i and y_i are the coordinates in millimeter. z_i is the human height. v_i is human linear velocity in mm/s. θ_{m_i} is the motion angle in $radius$. θ_{h_i} is the face orientation in $radius$. We only use the x and y coordinates, because of the limitations of our robot's perception capabilities.

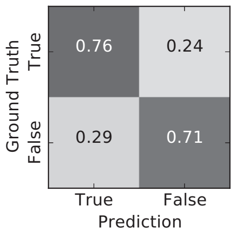
The input vector length is 60 including 30 pairs of x and y values. Our LSTM's hidden layer includes 50 memory units. In order to output binary classification results, we use a dense layer with sigmoid activation function in the output layer. We use Adam (Kingma and Ba 2014), a first-order gradient method, for optimization. The loss function was calculated using binary cross-entropy. For regularization, we use a dropout value of 0.2. The memory units and the hidden states of the LSTM are initialized to zero. The epoch size (number of passes over the entire training data) is 300. The batch size is 32. The data was split into sets for training (70%) and testing (30%). To implement the classifier training, we used Keras (Chollet and others 2015), an open-source python deep-learning library.

5.2 Illustrative Example

Consider an illustrative example: a *visitor* to a *classroom* building in the *afternoon* was *interested* to interact with the robot. The robot's goal is to identify the person's intention as efficiently and accurately as possible.

Human motion trajectory is captured by our robot using RGB-D sensors. Figure 3 presents a detected person, and the motion trajectory. The trajectory is passed to the LSTM-based classifier, which outputs the person being not interested in interaction (false negative).

The robot then collected facts about *time* and *location*. Domain knowledge enables the robot to be aware that: *professors* and *students* are more likely to show up in the classroom; and *visitors* are more likely to show up in the afternoon and to interact with the robot, whereas they are less likely to be present in the classroom building. Also, the



Ground Truth	True	False
True	0.76	0.24
False	0.29	0.71

Prediction

Figure 4: Confusion matrix of the LSTM classifier

LSTM classifier's confusion matrix, as shown in Figure 4, is encoded as a set of probabilistic rules in P-log, where the true-negative probability is 0.71. Therefore, given all the declarative contextual knowledge, the reasoner computes the following distribution over the variable of human identity, V_{id}^{rsn}, in the range of [*student, visitor, professor*]:

$$Dist(V_{id}^{rsn}) = [0.36, 0.28, 0.36] \tag{1}$$

Given the observed facts and the classifier's output, the robot queries its reasoner to estimate the distribution over possible human intentions

$$\hat{b}^{rsn} == [0.22, 0.78] \tag{2}$$

where 0.22 corresponds to the human being interested in interaction. $\hat{b}^{rsn}$ is the belief over state set $\hat{S}^{rsn}$ (a marginal distribution of both S^{rsn} and S^{pln}).

The reasoner's output of $\hat{b}^{rsn}$ is used for initializing the belief distribution, b^{pln}, for the POMDP-based planner:

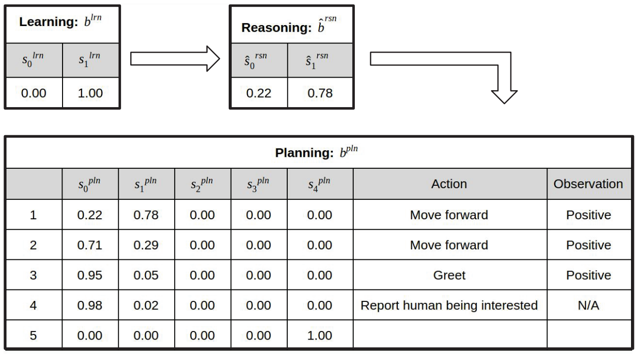
$$b^{pln} = [0.22, 0.78, 0, 0, 0]$$

where b^{pln} is over the state set of S^{pln} as described in Section 4.2. For instance, $s_0^{pln} \in S^{pln}$ is the state where the robot has not taken the "turn" action, and the human is interested in interaction. Similarly, $s_3^{pln} \in S^{pln}$ is the state where the robot has taken the action "turn", and the human is not interested in interaction.

During the action-observation cycles (in simulation), policy π maps b_{pln} to "greet" and "move forward" actions, and b^{pln} is updated until the robot correctly reported human intention and reached terminal state (s_4^{pln}). The actions, corresponding human feedback, and belief update are presented in Figure 5. Although, the LSTM classifier made a wrong estimation, the reasoner and planner helped the robot successfully recover from the wrong estimation.

5.3 Experimental Results

We did pairwise comparisons between LCORPP with the following methods for human intention estimation to investigate several hypotheses. Our baselines include: **Learning (L)**: learned classifier only. **Reasoning (R)**: reasoning with contextual knowledge. **Planning (P)**: POMDP-based interaction with uniform priors. **Learning+Reasoning (L+R)**: reasoning with the classifier's output and knowledge. **Reasoning+Planning (R+P)**: reasoning with knowledge and planning with POMDPs.



Learning: b^{lrn}	
$s_0{}^{lrn}$	$s_1{}^{lrn}$
0.00	1.00

Reasoning: $\hat{b}^{rsn}$	
$\hat{s}_0{}^{rsn}$	$\hat{s}_1{}^{rsn}$
0.22	0.78

Planning: b^{pln}							
	$s_0{}^{pln}$	$s_1{}^{pln}$	$s_2{}^{pln}$	$s_3{}^{pln}$	$s_4{}^{pln}$	Action	Observation
1	0.22	0.78	0.00	0.00	0.00	Move forward	Positive
2	0.71	0.29	0.00	0.00	0.00	Move forward	Positive
3	0.95	0.05	0.00	0.00	0.00	Greet	Positive
4	0.98	0.02	0.00	0.00	0.00	Report human being interested	N/A
5	0.00	0.00	0.00	0.00	1.00		

Figure 5: An illustrative example of information flow in LCORPP: the learning component generates "facts" and probabilistic rules for the reasoner, and the reasoner computes a prior distribution for the planner, which actively interacts with the environment.

Our experiments are designed to evaluate the following hypotheses. I) Given that LCORPP's prior belief is generated using reasoner and learner, it would outperform baselines in intention estimation in F1 score while receiving less action costs; II) In case of inaccurate knowledge, or III) the learner not having a complete sensor input, LCORPP's planner can compensate these imperfections by taking more actions to maintain higher F1 score. IV) In case of scarcity of data, robot's most recent interaction can be used to augment the dataset and improve overall performance of LCORPP.

In each simulated trial, we first sample human identity randomly, and then sample time and location accordingly, using contextual knowledge, such as professors tend to showing up early. According to the time, location, and identity, we sample human intention. Finally, we sample a trajectory from the test set of the dataset, according to the previously sampled human intention. We added 30% noise to the human reactions (robot's observation) being compliant with the ground truth but independent from robot's actions. LCORPP requires considerable computation time for training the classifier (∼10 min) and generating the POMDP-based action policy (∼1 min). The training and policy generation are conducted offline, so they do not affect the runtime efficiency. Reasoning occurs at runtime, and typically requires less than 1 millisecond.

LCORPP vs. Five Baselines: Figure 6 shows the overall comparisons using the six SDM strategies, each number is an average of 5000 trials, where the setting is the same in all following experiments. The three strategies, **P**, **R+P** and LCORPP, include the POMDP-based planning component, and perform better than no-planning baselines in F1 score. Among the planning strategies, ours produces the best overall performance in F1 score, while reducing the interaction costs (dialog-based and motion-based)(Hypothesis I).

Inaccurate Knowledge: In this experiment, we evaluate the robustness of LCORPP to inaccurate knowledge (Hypothesis II). Our hypothesis is that, in case of contextual knowledge being inaccurate, LCORPP is capable of recovering via actively interacting with people. We used knowledge bases (KBs) of different accuracy levels: **High**, **Medium**, and **Low**. A high-accuracy KB corresponds to the ground truth. Medium- and low-accuracy KBs are incomplete, and misleading respectively. For instance, low-accuracy knowledge suggests that *professors* are more likely to interact with the robot, whereas *visitors* are not, which is opposite to the ground truth. Figure 7 shows the results where we see the performances of **R** and **L+R** baseline strategies drop to lower than 0.4 in F1 score, when the contextual knowledge is of low accuracy. In F1 score, neither **R+P** nor LCORPP is sensitive to low-accuracy knowledge, while LCORPP performs consistently better than **R+P**. In particular, when the knowledge is of low accuracy, LCORPP retains the high F1 score (whereas **R+P** could not) due to its learning component.

Additional experimental results on Hypotheses III and IV are provided in the supplementary document.

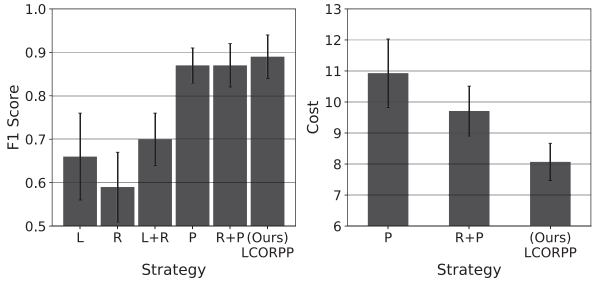


Figure 6: Pairwise comparisons of LCORPP with five baseline sequential decision-making strategies. The right subfigure excludes the strategies that do not support active human-robot interaction and hence produce zero costs.

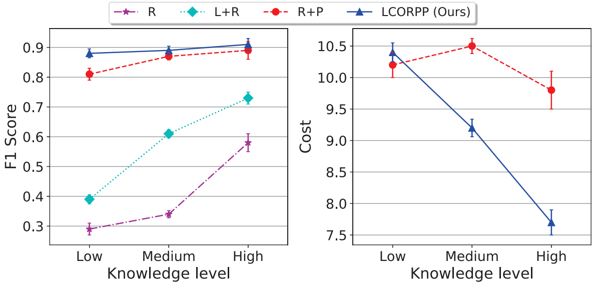


Figure 7: Performances of LCORPP and baselines given contextual knowledge of different accuracy levels: High, Medium and Low. Baselines that do not support reasoning with human knowledge are not included in this experiment.

6 Conclusions

In this work, we develop a robot sequential decision-making framework that integrates supervised learning for passive state estimation, automated reasoning for incorporating declarative contextual knowledge, and probabilistic planning for active perception and task completions. The developed framework has been applied to a human intention estimation problem using a mobile robot. Results suggest that the integration of supervised deep learning, logical-probabilistic reasoning, and probabilistic planning enables simultaneous passive and active state estimation, producing the best performance in estimating human intentions.

Acknowledgments

This work has taken place in the Autonomous Intelligent Robotics (AIR) Group at SUNY Binghamton. AIR research is supported in part by grants from the National Science Foundation (IIS-1925044), Ford Motor Company, and SUNY Research Foundation.

References

Aditya, S.; Saha, R.; Yang, Y.; and Baral, C. 2019. Spatial knowledge distillation to aid visual reasoning. In *WACV*, 227–235.

Amiri, S.; Wei, S.; Zhang, S.; Sinapov, J.; Thomason, J.; and Stone, P. 2018. Multi-modal predicate identification using dynamically learned robot controllers. In *IJCAI*, 4638–4645.

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57(5):469–483.

Bach, S. H., et al. 2017. Hinge-loss markov random fields and probabilistic soft logic. *The JMLR* 18(1):3846–3912.

Baral, C.; Gelfond, M.; and Rushton, N. 2009. Probabilistic reasoning with answer sets. *TPLP* 9(1):57–144.

Chen, X.; Li, L.-J.; Fei-Fei, L.; and Gupta, A. 2018. Iterative visual reasoning beyond convolutions. In *CVPR*.

Chen, H.; Tan, H.; et al. 2019. Enabling robots to understand incomplete natural language instructions using commonsense reasoning. *arXiv preprint arXiv:1904.12907*.

Chitnis, R.; Kaelbling, L. P.; and Lozano-Pérez, T. 2018. Integrating human-provided information into belief state representation using dynamic factorization. *arXiv preprint arXiv:1803.00119*.

Chollet, F., et al. 2015. Keras. https://keras.io.

Ferreira, L. A.; Bianchi, R. A.; Santos, P. E.; and de Mantaras, R. L. 2017. Answer set programming for non-stationary markov decision processes. *Applied Intelligence* 47(4):993–1007.

Gelfond, M., and Kahl, Y. 2014. *Knowledge representation, reasoning, and the design of intelligent agents: The answer-set programming approach*. Cambridge University Press.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *IEEE-ICASSP*.

Griffith, S.; Subramanian, K.; et al. 2013. Policy shaping: Integrating human feedback with reinforcement learning. In *NIPS*.

Hawes, N.; Burbridge, C.; Jovan, F.; et al. 2017. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine* 24(3):146–156.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hopfield, J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* 79(8):2554–2558.

Illanes, L.; Yan, X.; Icarte, R. T.; and McIlraith, S. A. 2019. Symbolic planning and model-free reinforcement learning: Training taskable agents. In *4th Multidisciplinary Conference on RLDM*.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101(1-2):99–134.

Kato, Y.; Kanda, T.; and Ishiguro, H. 2015. May i help you?: Design of human-like polite approaching behavior. In *HRI*.

Khandelwal, P.; Zhang, S.; Sinapov, J.; Leonetti, M.; Thomason, J.; Yang, F.; et al. 2017. BWIBots: A platform for bridging the gap between AI and human–robot interaction research. *The IJRR*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Kurniawati, H.; Hsu, D.; and Lee, W. S. 2008. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*.

Leonetti, M.; Iocchi, L.; and Stone, P. 2016. A synthesis of automated planning and reinforcement learning for efficient, robust decision-making. *Artificial Intelligence* 241:103–130.

Lifschitz, V. 2016. Answer sets and the language of answer set programming. *AI Magazine* 37(3):7–12.

Lu, D.; Zhang, S.; Stone, P.; and Chen, X. 2017. Leveraging commonsense reasoning and multimodal perception for robot spoken dialog systems. In *IROS*.

Lu, K.; Zhang, S.; Stone, P.; and Chen, X. 2018. Robot representation and reasoning with knowledge from reinforcement learning. *arXiv:1809.11074*.

Lyu, D., et al. 2019. Sdrl: interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. In *AAAI*.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine learning* 62(1-2):107–136.

Sridharan, M., and Rainge, S. 2014. Integrating reinforcement learning and declarative programming to learn causal laws in dynamic domains. In *International Conference on Social Robotics*.

Sridharan, M.; Gelfond, M.; Zhang, S.; and Wyatt, J. 2019. REBA: A refinement-based architecture for knowledge representation and reasoning in robotics. *JAIR*.

Taylor, M. E., and Borealis, A. 2018. Improving reinforcement learning with human input. In *IJCAI*, 5724–5728.

Thomaz, A. L.; Breazeal, C.; et al. 2006. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *AAAI*.

Veloso, M. M. 2018. The increasingly fascinating opportunity for human-robot-ai interaction: The cobot mobile service robots. *ACM THRI* 7(1):5.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Yang, F.; Lyu, D.; Liu, B.; and Gustafson, S. 2018. Peorl: Integrating symbolic planning and hierarchical reinforcement learning for robust decision-making. *arXiv preprint arXiv:1804.07779*.

Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Zellers, R.; Holtzman, A.; et al. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Zhang, S., and Stone, P. 2015. Corpp: Commonsense reasoning and probabilistic planning, as applied to dialog with a mobile robot. In *AAAI*, 1394–1400.

Zhang, S.; Sridharan, M.; and Wyatt, J. L. 2015. Mixed logical inference and probabilistic planning for robots in unreliable worlds. *IEEE Transactions on Robotics* 31(3):699–713.