Leveraging Domain Knowledge to Learn Normative Behavior: A Bayesian Approach

Hadi Hosseini[1]* and Mihaela Ulieru[2]

[1] David R. Cheriton School of Computer Science
University of Waterloo
h5hosseini@uwaterloo.ca
[2] Adaptive Risk Management Lab.
University of New Brunswick
ulieru@unb.ca

Abstract. This paper addresses the problem of norm adaptation using Bayesian reinforcement learning. We are concerned with the effectiveness of adding prior domain knowledge when facing environments with different settings as well as with the speed of adapting to a new environment. Individuals develop their normative framework via interaction with their surrounding environment (including other individuals). An agent acquires the domain-dependent knowledge in a certain environment and later reuses them in different settings. This work is novel in that it represents normative behaviors as probabilities over belief sets. We propose a two-level learning framework to learn the values of normative actions and set them as prior knowledge, when agents are confident about them, to feed them back to their belief sets. Developing a prior belief set about a certain domain can improve an agent's learning process to adjust its norms to the new environment's dynamics. Our evaluation shows that a normative agent, having been trained in an initial environment, is able to adjust its beliefs about the dynamics and behavioral norms in a new environment. Therefore, it converges to the optimal policy more quickly, especially in the early stages of learning.

Keywords: Learning and Adaptation::Single agent Learning, Agreement Technologies::Norms

1 Introduction

Norms or conventions routinely guide the choice of behaviors in human societies, and conformity to norms reduces social frictions, relieves the cognitive load on humans, and facilitates coordination and decision making [22][18]. Norms differ in various situations depending on the environment's dynamics, behaviors of other agents (including peers and superiors), and many other factors affecting them. For instance, in a crisis situation caused by flooding or an earthquake, first responders are responsible to control and (sometimes) enforce some rules to

* This work has been mostly done while at the University of New Brunswick

the people such as evacuating the area or preventing people from looting shops. However, a first responder might decide to let people break into a drug store (against his work policy constraints) in order to get urgent access to medical equipments.

When facing different environments, agents tend to spend some time understanding and learning the interaction patterns to adapt to the new setting. Developing a prior belief set about a certain domain, can improve an agent's learning process to adjust its normative behaviors with regards to the new environment's dynamics. An agent's ability to quickly adjust its beliefs and norms to different environments highly affects its performance of learning and, as a result, increases the overall utility of the agent. However, given the unpredictability of the world makes finding an appropriate set of norms or rules to initially code into agents is a highly difficult task. Our purpose is to overcome this difficulty by applying learning techniques to equip agents with proper tools for learning new norms in every different environment. Regardless of the type and origins of norms, they play an important part in forming and alternating beliefs in human societies, as actions are derived from the beliefs about the normative behaviors [20].

This paper proposes a two-level learning algorithm to extract the behavioral norms and reuse them as domain knowledge in future environmental settings. Determining where and when to extract norms is done using probability distributions of the state-action pairs. We would like to investigate the following questions: How effective is adding prior domain knowledge when facing environments with different settings? Having learned some behavioral norms, how quickly does an agent adapt to an environment?

The remainder of this paper is as follows: Section 2 gives a broad overview of the literature on norms, beliefs, and Bayesian model learning. In Section 3, we propose our two-level learning framework to extract norms using the Bayesian model learning technique and then discuss our algorithm for adaptation to change in new environments. Section 4 demonstrates our experimental results to find answers for the motivating questions. Finally, we give a conclusion to our work and propose the future work and possible directions for this area of research in Section 5.

2 Background and Related Work

2.1 Norms and Beliefs

Since norms arise based on interactions with the environment, they are very likely to be altered when there is a change in interaction patterns, goals, and beliefs. Also, conditions will change, which may lead to different behavior of the agents by affecting their belief sets. Norm autonomy is the highest level of autonomy, and it refers to social impacts on agents' choices. At this level, agents choose which goals are legitimate to pursue based on a given set of norms. Such agents (called norm autonomous agents or deliberative normative agents [7][4])

may judge the legitimacy of their own and other agents' goals. Autonomy at this level is defined as the agent's capability to change its norm system when a goal conflict arises , thereby changing priorities of goals, abandoning a goal, generating another goal, *etc* [25]. Dignum [14][15] provides another view of autonomy at the norm level, allowing the agents to violate a norm in order to adhere to a private goal that they consider to be more profitable, including in such consideration the negative value of the repercussions such a violation may have. Less restrictive sets of social norms may be chosen by agents, however, an agent is only allowed to deviate from a norm if it cannot act under the current limitations [5][6].

From the learning perspective, beliefs can be viewed as emergence of norms (from a game-theoretic point of view) and acceptance of norms (individual level of agents) [24][10][9]. While researchers have studied the emergence of norms in agent populations, they typically assume access to a significant amount of global knowledge. However, there is no guarantee that agents always have access to global knowledge. In addition, in some cases the global knowledge can be inconsistent and inaccurate due to the changes that happen over time. Behavioral norms are domain-dependent and context-sensitive norms, meaning that in every situation based on the signals one perceives from its environment these norms can be changed or altered. In the absence of a centralized authority or when facing an environment with different settings, an agent should adjust its belief set to be able to act properly.

Sen et al. [22] studied the emergence of norms in a game-theoretic approach where individual agents learn social norms by interactions with other agents. Moreover, in [21], the emergence of social norms in heterogenous agent societies has been studied to explore the evolution of social conventions based on repeated distributed interactions between agents in a society. The authors considered that norms evolve as agents learn from their interactions with other agents in the society using multi-agent reinforcement learning algorithms [22]. Most of the work in this area fall short in considering norms as changeable elements depending on the environment. Norm adaptation uses an agent's domain knowledge to adjust more quickly in new environments. Unlike [16] that studies norm adaptation and effects of thinking in norms using computational approaches, we are interested in using the very natural way of learning used by humans. In Bayesian reinforcement learning (RL), agents are able to gather information about different environments and settings. After many experiences, this information leads to knowledge of the domain in which the agents are mostly working.

2.2 Bayesian Model Learning

The Bayesian approach is a principled, non problem-specific approach that provides an optimal solution to the action choice problem in RL. The optimal solution to the RL action selection problem or optimal learning, is the pattern of behavior that maximizes performance over the entire history of interactions of an agent with the world [12][11][8]. With Bayesian learning techniques, an agent stores a probability distribution over all possible models, in the form of a belief

state [11]. The underlying (unknown) Markov Decision Process (MDP), thus, induces a belief-state MDP. The transition function from belief state to belief state is defined by Bayes' rule, with the observations being the state and reward signals arising from each environmental transition.

Assume an agent is learning to control a stochastic environment modeled as an MDP, which is a 4-tuple $\langle S, A, P_T, P_R \rangle$ with finite state and action sets S, A, transition dynamics P_T and reward model P_R. The agent is charged with constructing an optimal Markovian policy $\pi : S \mapsto A$ that maximizes the expected sum of future discounted rewards over an infinite horizon. Letting $V^*(s)$ at each $s \in S$ denote the optimal expected discounted reward achievable from state s and $Q^*(s, a)$ denote the value of executing action a at state s, we have the standard Bellman equations [1]:

$$V^*(s) = max_{a \in A} Q^*(s, a) \tag{1}$$

$$Q^*(s, a) = E_{P_R(s,a,r)}[r|s, a] + \gamma \sum_{s' \in S} P_T(s, a, s') V^*(s') \tag{2}$$

At each step in the environment, the learner maintains an estimated MDP $\langle S, A, \widehat{P_T}, \widehat{P_R} \rangle$ based on an experience tuple of $\langle s, a, t, r \rangle$; that is, at each step in the environment the learner starts at state s, chooses an action a, and then observes a new state t and a reward of r. This MDP then can be solved at each stage approximately or precisely depending on an agent's familiarity with state and reward distributions.

A Bayesian agent estimates a model of uncertainty about the environment (discovering P_T and P_R) and takes these uncertainties into account when calculating value functions. In theory, once the uncertainty is fully incorporated into the model, acting greedily with respect to these value functions is the optimal policy for the agent, the policy that will enable it to optimize its performance while learning. Bayesian exploration is the optimal solution to the exploration-exploitation problem [19][2].

In the Bayesian approach a belief state over the possible MDPs is maintained. A belief state defines a probability density. Bayesian methods assume some prior density P over possible dynamics D and reward distributions R, which is updated with an experience tuple $\langle s, a, t, r \rangle$. Given this experience tuple, one can compute a posterior belief state using Bayes' rule. We are looking for the posterior over reward model distribution and also the posterior for the transition model, given an observed history of H. Considering H to be the state-action history of the observer, an agent can compute the posterior $P(T, R|H)$ to determine an appropriate action at each stage. As the density P is the product of two other densities $P(T^{s,a})$ and $P(R^{s,a})$, that is, the probability density of choosing action a in state s and the probability density of getting the reward of r by choosing an action a when in state s, we should make an assumption to simplify this calculation.

Based on [11], our prior satisfies parameter independence, and thus the prior distribution over the parameters of each local probability term in the MDP is independent of the prior over the others. This means that the density P is

factored over R and T with $P(T|R)$ being the product of independent local densities $P(T^{s,a})$ and $P(R^{s,a})$ for each transition and each reward distribution. It turns out that this form is maintained as we incorporate evidence. The learning agent uses the formulation of [11] to update these estimates using Bayes' rule:

$$\begin{aligned} P(T^{s,a}|H^{s,a}) &= zP(H^{s,a}|T^{s,a})P(T^{s,a}) \\ P(R^{s,a}|H^{s,a}) &= zP(H^{s,a}|R^{s,a})P(R^{s,a}) \end{aligned} \tag{3}$$

where $H^{s,a}$ is the history of taking action a in state s, and z is a normalizing constant.

It has been assumed that each density $P(T^{s,a})$ and $P(R^{s,a})$ is a Dirichlet [13] as the transition and reward models are *discrete multinomials*. These priors are conjugate, and thus the posterior after each observed experience tuple will also be a Dirichlet distribution [11][8].

3 The Proposed Two-level Learning Framework

Two types of learning are considered in this framework: first, learning while the agent is exploring and exploiting rewards in each episode[3] of the same simulation (in the same environment) and trying to learn the environment's dynamics, and second, a high-level approach to capture the domain's specific normative behaviors. This framework is able to learn the system's dynamics, specifically the environment's dynamics and interaction patterns for each setting. A key factor for optimizing the performance of agents is to provide them with knowledge about the dynamics of the environment and behavioral norms.

Behavioral norms about the environment's dynamics can be extracted using the probability distribution of each state-action pair after agents get into a reasonable confidence level about their beliefs. Afterwards, this knowledge gets updated and added to all the previous data gained in the past experiences. The overall knowledge represents the agent's belief about the normative actions and can be incorporated into agents as prior knowledge [17].

Every domain has its specific set of norms (known as behavioral norms) that can be generally valid in other environments. There is a mutual connection between behavioral norms and domain-dependent knowledge in reinforcement learning. Norms can be extracted through reinforcement learning (RL), and RL can be improved by incorporating behavioral norms as prior probability distributions into learning agents.

3.1 Adapting to Change

Traditionally a norm, be it an obligation, prohibition or permission, is defined as a rigid value - yet real life exhibits normative behavior more flexible and context

[3] An episode is every trial in which agents begin in the start state and finishes in the goal state.

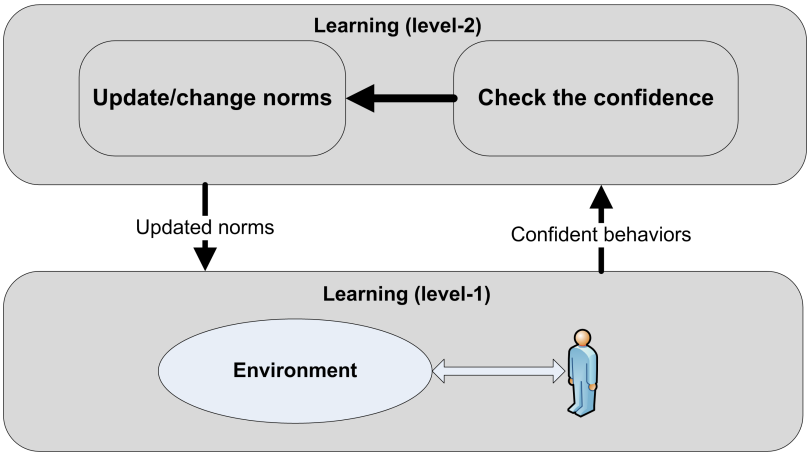


Fig. 1: Simple sketch of the two-level learning framework

dependent since the participants in social situations can behave unpredictably. Take for example the case of a frustrated player after a game. The social norm would dictate this player to shake hands with the winning team, yet he may well choose to ignore the norm under the circumstance. Therefore it makes sense to set the normative framework using a probabilistic model that would enable the agent behavior to be adjusted to the particular situation by assigning to each norm a probability reflecting the degree by which this norm may be followed or not by the agents operating under the respective framework. As agents interact with other agents as well as with the environment in the particular context [21] they can attune their behavior by changing the values of the probabilities assigned to the norms governing them, in a continuous adaptation process.

This is a probabilistic model of expressing norms where a prohibited norm is a norm with low level probability to happen, however, its probability is not necessarily 0 (although it can be close to 0). Similarly, an obligated norm can have a probability close to 1. By modeling norms as probabilistic values over a belief set, we are able to extract these values via reinforcement learning techniques.

3.2 Transition and Reward Densities

In our Bayesian learning model, each density $P(T^{s,a})$ and $P(R^{s,a})$ is a Dirichlet. However, Dirichlet distributions make the implementation and tracking of the algorithm quite hard, since the transition model will be sparse with only a few states that can result from a particular action at a particular state. If the state space is large, learning with a Dirichlet prior can require many examples to recognize that most possible states are highly unlikely [11][23]. To avoid these problems, we use beta distributions for every state and action. In Bayesian statistics, it can be seen as the posterior distribution of the parameter p of a binomial distribution after observing $\alpha - 1$ independent events with probability

p and $\beta - 1$ with probability $1 - p$, if there is no other information regarding the distribution of p. We consider a binomial probability distribution for every state-action pair. These distributions actually show us the number of times in which every state-action pair succeeds or fails during the simulation. We need to maintain the number of times, $N(s \xrightarrow{a} s')$, state s is successful to make transition to s' when action a is chosen, and similarly, $N(s \xrightarrow{a} r)$ for rewards. With the prior distributions over the parameters of the MDP, these counts define a posterior distribution over MDPs.

3.3 Dynamic Norm Adaptation with Bayesian Reinforcement Learning

Agents gain knowledge about the environment's dynamics using dynamic programming iterations and updates. By visiting every state or choosing actions, agents gradually build up their knowledge about the environment as probability distributions over state-action pairs. This information can be considered to be incomplete or false during the simulation until agents are *confident* about their beliefs. From the exploration-exploitation perspective, this confidence is gained when the agent has knowledge about most of the states and the permissible actions in them or the value of each action in every state. Thus, agents are said to be confident about their beliefs when (1) The algorithm has converged into an optimal policy in the learning process(or cumulative reward becomes steady in the recent episodes), and (2) Most of the states have been visited by the agent.

In the first condition, it is not really easy to understand when an algorithm will converge to an optimal policy. It needs complicated and time-consuming mathematical calculations. Bayesian dynamic programming is proved to converge to an optimal policy using some optimization techniques [3]. However, checking this criterion is a complicated process. The algorithm 1 shows the steps within each episode.

We introduce an element to check the confidence level at the end of each episode. When an episode is finished, the goal state is reached, and we are able to look at the cumulative reward gained in that episode by our agent. If this cumulative reward is in a steady state in recent episodes, it is a good measure to be sure that our Bayesian algorithm is in a reliable state, meaning that the algorithm is in equilibrium.

The amount of cumulative reward or the number of steps to the goal is not solely a good metric to measure the *level of confidence* [17]. What also is important for agents is to make sure that they have at least some sort of sufficient information about the current world and the majority of states. This can be measured by counting the number of explored states so far, indicating how many states have been visited by an agent.

The level of exploration (LOE) is defined simply as follows:

$$LOE = \frac{E}{N} \tag{4}$$

Algorithm 1 Generating Behavioral Norms

loop
  at each episode n
  for all $s \in S \wedge a \in A$ **do**
    $V^*(s) = max_{a \in A} Q^*(s,a)$
    $Q^*(s,a) = E_{P_R(s,a,r)}[r|s,a] + \gamma \sum_{s' \in S} P_T(s,a,s') V^*(s')$
  end for
  for all $s \in S, a \in A$ **do**
    $P(T^{s,a}|H^{s,a}) = zP(H^{s,a}|T^{s,a})P(T^{s,a})$
    $P(R^{s,a}|H^{s,a}) = zP(H^{s,a}|R^{s,a})P(R^{s,a})$
  end for
  $LOE \leftarrow \frac{E}{N}$
  if $LOE > threshold$ **then**
    if $n > k$ **then**
      if $CR_n = [\sum_{i=n-k}^{n-1} CR_i/k] \pm (1 - LOE + \epsilon)$ **then**
        $prior_{new} = posterior_{old} + prior_{old}$
      end if
    end if
  end if
end loop

where E is the number of explored states so far in the simulation, and N is the estimated total number of states. We assume that the size of state space is known by the agent in the beginning of each episode. LOE is always smaller than or equal to 1. As it gets closer to 1, more states of the environment have been explored.

It is proposed that the agent can be confident about its beliefs when $LOE \geqslant 0.9$ and CR_n satisfies equation 5.

$$CR_n = [\sum_{i=n-k}^{n-1} CR_i/k] \pm (1 - LOE + \epsilon) \tag{5}$$

where CR_n is the cumulative reward gained in the n_{th} episode, and k is a desired number of recent episodes. Based on every experiment and the size of the state-space, one can decide to consider k previous cumulative rewards to average them (In this paper $k = 5$).

The cumulative reward gained in each episode can be different even after converging to the optimal policy, as the agent is always in the learning process and may explore some other states. Therefore, the value of CR_n should fall into a plus/minus interval to be acceptable. This interval depends on the value of LOE. If not many of the states have been explored so far, the interval gets larger. The cumulative rewards become closer and closer to each other when the majority of states have been covered. In a nutshell, the more states that have been explored by an agent, the smaller the interval gets. Although LOE rarely reaches 1, the ϵ in this formula makes sure that there is always an interval even when $1 - LOE$ is equal to 0.

When an agent meets these two conditions and becomes confident about its information on normative behaviors, it should simply update its belief state and add this newly learned knowledge to its knowledge base.

3.4 Updating Prior Knowledge as Norms

Updating the Bayes parameter estimate with new information is easy by using the concept of a conjugate prior. The parameter estimate obtained from the previous episodes should be combined with the estimates an agent already has about its states and actions. Essentially, a conjugate prior allows agents to represent the Bayes parameter estimation formula in simple terms using the beta parameters a and b:

$$\begin{aligned} a_{posterior} &= a_{prior} + a_{data} \\ b_{posterior} &= b_{prior} + b_{data} \end{aligned} \tag{6}$$

We consider state-action pairs as binomial probability distributions showing us the number of times each state-action succeeds or fails. The beta parameters in beta distributions are the number of successes and the number of failures. The posterior is simply given by adding the prior parameter and data parameter (the number of successful transitions from state s to s' under a). Updating norms is exactly the same as updating posteriors. Agents are continuously building and updating their posteriors using the aforementioned process. As this information is obtained by agents interacting in the environment (to solve a problem or to pursue a goal), it is representative of the environment's dynamics and norms. When an agent is in a confident level about its knowledge, it keeps a copy of the reward and the transition model and then updates its posterior by replacing the posterior gained so far with the prior distribution of tested data (data parameter).

4 Experimental Results and Analysis

Although real-world problems of norm generation are much more complicated, representing the world and its dynamics in a simple way can help us show a proof of concept. Furthermore, every decision-making situation where a learning agent needs to take an action under uncertainty can be easily mapped into a belief-state MDP. As such, using the proposed techniques, an agent will be able to solve the MDP, learn the model of the environment, and generate norms if the confidence level is reached. The implementation framework that is used to code these ideas is the one developed by Dr. Sutton in the RLAI lab[4]. This framework provides the basic tools to implement any desired RL algorithm.

Figure 2 shows a sample map. The agent can move left, right, up, or down by one square in the maze. Every action is representative of a behavioral norm. If

[4] http://rlai.cs.ualberta.ca/

it attempts to move into a wall, its action has no effect. The problem is to find a navigation path from the start state ('S') to the goal state ('G') with the fewest possible steps and the highest cumulative reward. The agent also should gather as much information as possible about the environment and its dynamics. When it reaches the goal, the agent receives a reward equal to 1, and the problem is then reset. Any step has a small negative reward of -0.05. The agent's goal is to find the optimal policy that maximizes its cumulative reward. The problem is made more difficult by assuming that the agent occasionally "slips" and moves in a direction perpendicular to the desired direction (with probability 0.1 in each perpendicular direction).

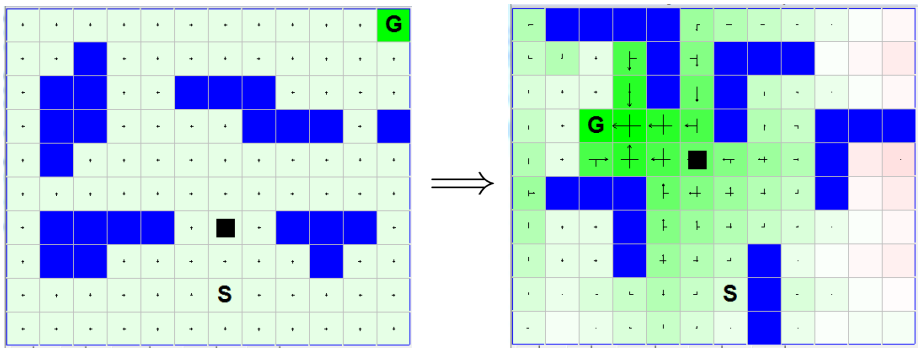


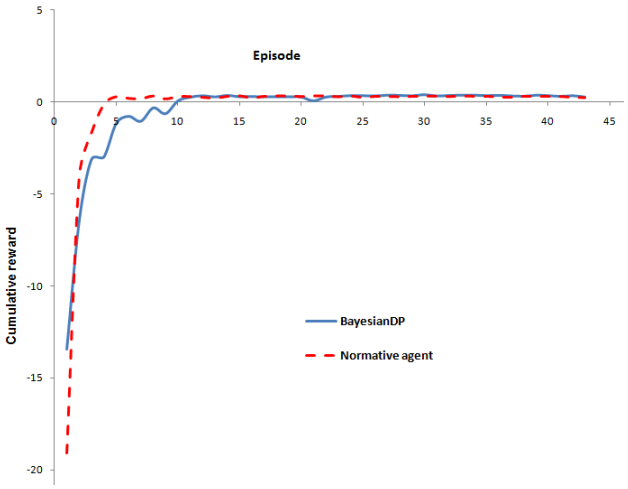
Fig. 2: A sample map showing changes in the environment

4.1 Experiments

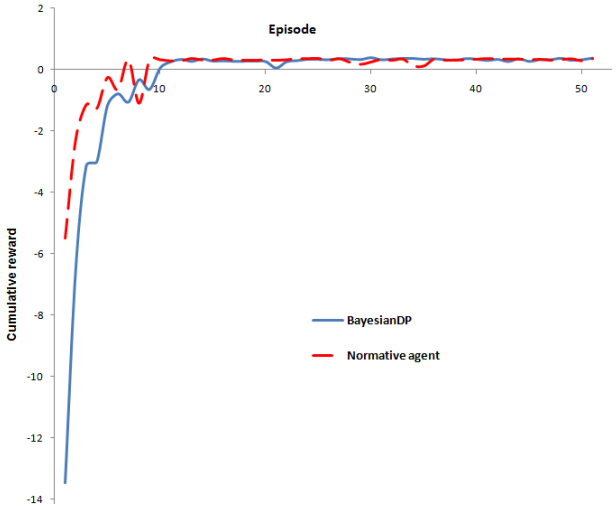
Here we present experiments by which we determine the effectiveness of the two-level reinforcement learning framework to dynamically generate appropriate norms. An agent's behavior in any environment is tightly dependent on its understanding of the surrounding environment.

Three different experiments are considered with two agents: a Bayesian agent with no prior knowledge about the dynamics and behavioral norms, and a Bayesian agent with some training in a different environment under the same domain. The environment's dynamics and its behavioral norms will be changed to study which agent better performs when confronting a new setting.

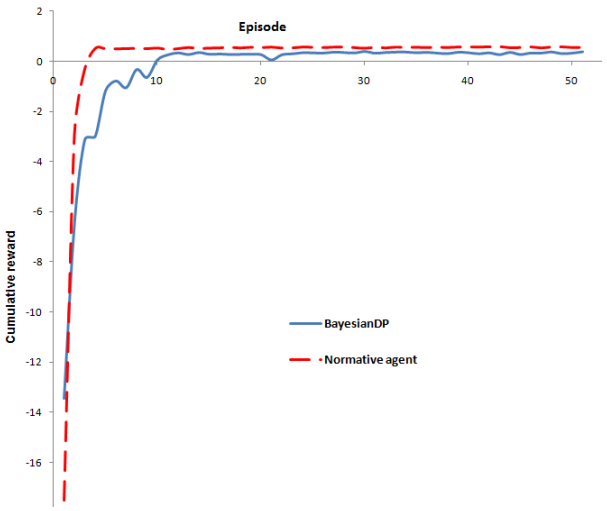
An interesting approach to study this difference is to consider the differences based on the percentage of changes in the settings. This way we are able to study the effectiveness of the learned normative behaviors in different environments. Nonetheless, as it was emphasized earlier, the domain in which the agent is finding an optimal policy to the goal state will remain the same. In these experiments, changes can occur in every element of the environment such as blocked states, goal states, start states, etc. Three different experiments have been done based on the percentage of changes:



(a) goal change



(b) 20% change



(c) 50% change + goal change

Fig. 3: Different percentages of change (averages over 10 runs)

- Only change in the goal state
- 20% change in the environment
- 50% change of dynamics + change in the goal state

Figure 3.a shows the performance of both agents with regard to cumulative rewards gained in each episode. The results are averages over 10 runs. Both of the agents find the best policy quickly in fewer than 15 trials. The normative agent starts up with a worse result compared to the Bayesian agent with no prior. This is due to the fact that the normative agent needs some exploration to adapt its beliefs to the new environment's dynamics, so it has to update its beliefs about the environment. However, after the first exploration of the map it rapidly finds the best policy and converges after 5 trials, as opposed to the Bayesian agent with no training.

In the very first trials of learning, the normative agent starts with finding the new optimal policy. On the other hand, some fluctuations in early phases show the agent's attempts to explore the new environment and find out the dynamics as well as exploiting the already known states. In the early learning process, the increase in performance of the normative agent with prior knowledge is statistically significant, compared to the Bayesian agent with no knowledge about the normative actions. A trained agent learns the probability of finding the goal state in each zone of the map so the agent focuses more on the areas that have been learned to be more probable in containing the goal state. In this example, this leads the agent to focus more on the central areas and avoid exploring behind the blocked states in the right and left sides of the map.

As shown in Figure 3.c , we notice some increased drop in the value of cumulative reward in the first episodes because the agent is adapting its belief state under the new dynamics. However, the value of cumulative reward rises more rapidly and converges to the value of the optimal policy after about 5 episodes. This proves the effectiveness of having prior knowledge about the domain-dependent norms even if the environment changes over time and the agent wants to start learning in a world with a different dynamics and different normative system. A paired t-test demonstrates that the difference in means between the normative agent and the agent with no prior knowledge is statistically significant ($p = 0.022022831$).

4.2 Lessons Learned

The performance of an agent, whether it has prior knowledge about the normative behaviors or not, converges at some point at a reasonable pace. However, an important factor is to avoid any random exploratory behavior at the beginning of a simulation. As we can see in Figure 4, the normative agent performs better both in gaining cumulative reward and finding the optimal policy to the goal. The more similar the new environment is to the environment where the agent has been trained, the faster and better it can adjust its beliefs to the new situations.

One interesting observation is that whenever the goal state is very different from the one learned by the agent, the agent has to violate or alter its beliefs to the new situations. Thus, this adjustment process makes the agent override some of the behavioral norms and spend some time exploring the new environment. However, as the agent carries its domain knowledge from the previous experiments, it easily adapts its normative system after just a couple of episodes. The more it takes for the agent to find the best policy, the more it should update/alter its belief systems on behavioral norms.

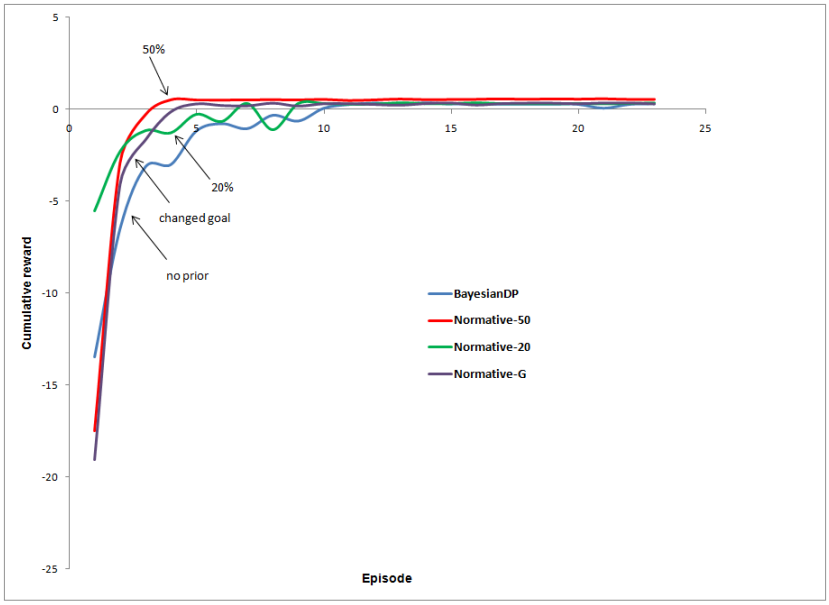


Fig. 4: The comparison between different values of change

The figure shows that the agent performs better in an environment with 20% change in its dynamics. On the other hand, when the agent has to perform in an environment with 50% change, it takes more stages at the beginning for the agent to adjust its knowledge to the new environment. Moreover, in the early stages of learning the agent gets a highly negative reward as the goal has been changed, and the agent needs to explore and unlearn its current beliefs.

5 Conclusion and Future Work

In this paper, we addressed the problem of norm adaptation using Bayesian reinforcement learning. Individuals develop their normative framework via interaction with their surrounding environment (including other individuals). Developing a prior belief set about a certain domain can improve an agent's learning

process to adjust its normative behaviors with regards to the new environment's dynamics. Our evaluation demonstrated that even in the environments with 50 percent of change in the states and the goal state, agents can quickly adapt to new settings using the practiced prior knowledge in a different environment, and thus, the performance of the agent increases, especially in the early stages of the learning process.

As a future work, we would like to run the same experiments in an environment with lower percentage of similarities. It would be interesting to show how fast agents can adapt to the new environment, and if having some knowledge about the domain will help the learning agents improve under different dynamics. We will experiment environments with higher percentage of differences in terms of states, goals, and transition functions to point out a threshold where after that the agent will perform similar to an agent with no prior knowledge.

Another direction might be to consider inconsistency in norms when norms have different origins. As it was shown in [15], the problem of these conflicts is not that they are general (logical) conflicts between the norms, but that they are only conflicts in very specific situations or even in ways in which norms are fulfilled. An important question is how one can handle these conflicting norms when agents confront groups or societies with completely opposite norms.

Acknowledgment

We would like to thank Dr. Michael W. Fleming for his constructive discussions and his fruitful comments.

References

1. R. Bellman. Dynamic Programming, Princeton. *NJ: Princeton UP*, 1957.
2. R. Bellman. Adaptive control processes: a guided tour. *Princeton University Press*, 1:2, 1961.
3. D. Bertsekas. *Dynamic programming: deterministic and stochastic models.* Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1987.
4. G. Boella and L. Lesmo. *Deliberate normative agents.* Norwell: Kluwer, 2001.
5. M. Boman. Norms in artificial decision making. *Artificial Intelligence and Law*, 7(1):17–35, 1999.
6. W. Briggs and D. Cook. Flexible social laws. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 14, pages 688–693. LAWRENCE ERLBAUM ASSOCIATES LTD, 1995.
7. C. Castelfranchi, F. Dignum, C. Jonker, and J. Treur. Deliberative normative agents: Principles and architecture. *Intelligent Agents VI. Agent Theories Architectures, and Languages*, pages 364–378, 2000.
8. G. Chalkiadakis and C. Boutilier. Coalitional bargaining with agent type uncertainty. In *Proc. 20th IJCAI*, 2007.
9. R. Conte and C. Castelfranchi. *Cognitive and social action.* Garland Science, 1995.
10. R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm acceptance. In *Intelligent Agents V. Agent Theories, Architectures, and Languages: 5th International Workshop, ATAL'98, Paris, France, July 1998. Proceedings*, pages 66–66. Springer, 2000.

11. R. Dearden, N. Friedman, and D. Andre. Model based Bayesian exploration. In *Proceedings of the fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 150–159. Citeseer, 1999.
12. R. Dearden, N. Friedman, and S. Russell. Bayesian Q-learning. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 761–768. JOHN WILEY & SONS LTD, 1998.
13. M. DeGroot. *Optimal statistical decisions*. Wiley-IEEE, 2004.
14. F. Dignum. Autonomous agents with norms. *Artificial Intelligence and Law*, 7(1):69–79, 1999.
15. F. Dignum and V. Dignum. Emergence and enforcement of social behavior. In *18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation*, pages 2377–2383, 2009.
16. J. Epstein. Learning to be thoughtless: Social norms and individual computation. *Computational Economics*, 18(1):9–24, 2001.
17. H. Hosseini. A Reinforcement Learning Approach to Dynamic Norm Generation. Master's thesis, University of New Brunswick, 2010.
18. D. Lewis. *Convention: A philosophical study*. Wiley-Blackwell, 2002.
19. J. Martin and O. R. S. of America. *Bayesian decision problems and Markov chains*. Wiley New York, 1967.
20. A. Morris, W. Ross, H. Hosseini, and M. Ulieru. *Modeling Culture with Complex, Multidimensional, Multiagent Systems*. In Virginia Dignum, Frank Dignum, Jacques Ferber, Tiberiu Stratulat (Eds.), Integrating Cultures: Formal Models and Agent-Based Simulations, Springer Series on the Philosophy of Sociality (in print), 2011.
21. P. Mukherjee, S. Sen, and S. Airiau. Emergence of Norms with Biased Interactions in Heterogeneous Agent Societies. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*, pages 512–515. IEEE Computer Society, 2007.
22. S. Sen and S. Airiau. Emergence of norms through social learning. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1507–1512, 2007.
23. M. Strens. A Bayesian framework for reinforcement learning. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 943–950. Citeseer, 2000.
24. R. Tuomela. *The importance of us: A philosophical study of basic social notions*. Stanford Univ Pr, 1995.
25. H. Verhagen. Norms and artificial agents. In *Sixth Meeting of the Special Interest Group on Agent-Based Social Simulation, ESPRIT Network of Excellence on Agent-Based Computing*, 2001.