Targeted Information Gain for Real-Time Reactive Policy Learning for Robotic Weed Killing

Wyatt McAllister1

[1]University of Illinois at Urbana-Champaign

November 14, 2017

Motivation

Distributed Autonomous Systems

- Increase the capability of autonomous devices for distributed applications in dynamic and uncertain environments.
- Enhance the applicability of such devices for widespread industrial applications.
- Guarantee that such systems have high performance and safety while being fully autonomous, but have the capability to improve their performance by interaction with users and with the environment.
- Use such systems to augment human capability by both improving quality of life through improved infrastructure, and compensating users in goods and services for their contribution to teaching this infrastructure.

Key Directions to Achieve These Goal

Critical Goals

- Demonstrate the capability of current autonomous decision making frameworks for novel industrial applications while enhancing their performance in such applications.
- Develop a scalable multi-tiered tasks communication framework for specifying clear instructions to different systems in different human languages.
- Create a hierarchical learning and decision framework that dynamically integrates multiple task specific learners and planners into a system which has the ability to plan abstractly to complete high level objectives.

Key Directions to Achieve These Goal

Step 1

- **Demonstrate the capability of current autonomous decision making frameworks for novel industrial applications while enhancing their performance in such applications.**
- Develop a scalable multi-tiered tasks communication framework for specifying clear instructions to different systems in different human languages.
- Create a hierarchical learning and decision framework that dynamically integrates multiple task specific learners and planners into a system which has the ability to plan abstractly to complete high level objectives.

Industrial Application

High Level Learning and Planning for the TERRA Project

- Design distributed autonomous systems for scalable applications in industrial agriculture.
- Use many small low cost robots to provide for small agriculture applications in developing nations as well as large farms.

Problem in Application

Weeding Under a Crop Canopy

- For many crops, such as corn, weeding must be done under a canopy, and therefore under partially observable conditions.
- Robots should have the ability to coordinate their weeding under this partial observability, to optimize yield using minimal time and resources.
- In these circumstances, robots can only classify weeds within a local radius, and thus only information about the neighboring rows is known.

Relevant Sources

Robot Foraging

- Foraging has long been considered a key problem in multi-agent robots [Cao et al., 1997].
- Recent work in multi-agent has solved this problem under partial observability, solving a search and rescue problem with ground robots and UAVs [Liu et al., 2017].
- However, this work assumed the capability of the UAVs to localize the victims. We aim to design a real-time system for the weeding task using only the information collected from the surroundings of the ground robots.

MDP Formulation

Station, Action, Reward

- State: Current row for each agent.

$$S\left(t\right)=\left\{x_i\left(t\right),i\in I\right\},I\equiv\left\{1,...,N_{\mathrm{agents}}\right\}$$

- Action: Target row for each agent.

$$A\left(t\right)=\left\{a_i\left(t\right)=x_i\left(t+1\right),i\in I\right\}$$

- Reward: We want to simultaneously maximize the number of weeds killed and minimize the operation time. Therefore, the reward is the sum over all agents of number of weeds killed, time discounted by the operation time. This structure is similar to that used in past work on multi-robot foraging [**?**].

$$R\left(t\right)=\sum_{i\in I}\gamma^{T_{\textbf{operation},\mathrm{i}}\left(a_i\left(t\right)\right)}R_{\text{weeds killed},\mathrm{i}}\left(a_i\left(t\right)\right)$$

Assumptions for MDP Approach

Full Communication - Centralized Approach

- For this project, we start with the assumption of full communication between all the agents about their current location, the action they have selected, and the total reward they have collected from the environment.

$$\{S(t), A(t), R(t)\} \Rightarrow \text{Known}$$

- The reason for this is that each acre of crop land is only 209 feet by 209 feet, so each agent will be at most 295 feet away, well within the range of shortwave communication.
- Assumptions about partial communication will only decrease performance, so it is sensible to start with the case of full communication to characterize the performance of the learning approach as compared to the naive approach.

Assumptions for MDP Approach

Single Agent Selection

- In our environment, we assumed robots can only cross rows at the edges of the field, and two robots cannot always move side by side. Therefore, selecting one agent per row is sensible in order to maximize efficiency, so no agent picks the same row.

$$A = \{a_i\left(t\right) : a_i\left(t\right) \neq a_j \quad \forall i \neq j\left(t\right)\}$$

- Identifying breaks in the rows which are traversable or points at which agents may move side by side might improve performance, but we will first explore this simplified environment before more complex scenarios.
- We therefore select the agent with the maximum value for each row, and break ties based on increasing order of agent indices.

Assumptions for MDP Approach

One Step Look Ahead

- In this set of experiment, all agents are homogeneous, meaning they have identical capabilities. Therefore, they each have identical value for the same rows under identical initial conditions.

$$x_i\left(t\right) = x_j\left(t\right) \Rightarrow R_i\left(a_i\left(t\right)\right) = R_j\left(a_j\left(t\right)\right) \quad \forall\left(i,j\right) \in I$$

- This implies that the value of the one-step policy for different time instances will not change depending on the states of the agents, and thus one step learning is sensible for this environment.
- In the long term, it will be useful to create robots with varying specializations and incorporate them into the simulation. However, current TERRA bots are homogeneous, so this is the framework we start with.

Assumptions for MDP Approach

The Case of Full Observability

- As a baseline, we consider the case of full observability, where we assume the number and heights of weeds in every row is known.
- In this case, we can easily compare the naive approach to the learning approach, which we expect to have comparable or superior performance.

$$R_W(x) \Rightarrow \text{Known} \quad \forall x$$

The Case of Partial Observability

- In the partially observable case, we assume that robots can classify weeds within a certain radius.

$$R_W(x) \Rightarrow Known \forall \{x : \{x_{visited}\} \pm r_{obs.}\}$$

- The current hardware system uses vision for weed classification and may be expected to do well classifying nearby weeds.
- We collect information on the neighboring rows as we move down the current row, and update the global environmental model with this information.

Explanation of Reward

- The reward model was chosen to maximize the rate of weeding over all the agents, to weed the field as efficiently as possible with the fewest resources.
- The current one-step reward for each agent is the expected reward of the proposed row.

$$R_i\left(a_i\left(t\right)\right) = \sum_{y=0}^{N_{\mathrm{dim}}} R_W\left(a_i\left(t\right), y\right)$$

- The current one-step operation time for each agent is the expected operation time of the proposed row.

$$T_i\left(a_i\left(t\right)\right) = \frac{\left(x_i\left(t+1\right) - x_i\left(t\right)\right)}{v_i} + \frac{Y_{\mathrm{dim}}}{v_i} + T_{\mathrm{kill}} \cdot N_W\left(x_i\left(t+1\right)\right)$$

- Here, N_{dim} is the number of squares in a row (in this case 85), Y_{dim} is the length of each row (in this case 209 feet), $N_W\left(x\right)$ is the number of weeds in each row x, and $R_W\left(x, y\right)$ is the reward for each weed at each location $\left(x, y\right)$ (the reward increases from a baseline value as the weed grows).

RP Approach

Multi-Agent Reactive Policy Learning with One Step Look Ahead

- We take a factored MDP approach to this problem, which allows us to break up the problem into a set of decentralized MDPs for each agent.
- This decentralized approach is guaranteed to find an optimum solution, since factored MDPs which are observation, transition, and reward independent may optimize the total additive reward by optimizing each agents local reward [Amato et al., 2013].
- For the Reactive Policy (RP), we plan simultaneously across all the agents, evaluating the expected return for a transition from the agent's current state to any other new state.

$$Q^i_{t+1}\left(x_i\left(t\right), a_i\left(t\right)\right) += \alpha\left(\gamma^{T_i\left(a_i\left(t\right)\right)} \cdot R_i\left(a_i\left(t\right)\right) - Q^i_t\left(x_i\left(t\right), a_i\left(t\right)\right)\right)$$

- By simultaneously optimizing over the value function for each agent, we plan a coordinated policy sending each agent to the row with maximum value.

$$A\left(t+1\right) = \left\{\underset{a_i\left(t\right)}{\arg\max}\, Q^i_{t+1}\left(x_i\left(t\right), a_i\left(t\right)\right) \quad \forall i \in I\right\}$$

Factored MDP Model

Observation, Transition, and Reward Independence [Amato et al., 2013]

- In this case, the total observation space, when not fully observed, is constructed via the observations of each agent, and the factored model is thus observation independent.
- Our chosen transition model is such that the transition for each agent is uniquely determined by their current state and action, and does not depend on the other agents, and the factored model is thus transition independent.
- Our chosen reward model is such that the reward for each agent is uniquely determined by their current state and action, and does not depend on the other agents, and the factored model is thus reward independent.
- Since the overall reward is the sum of each agents individual reward, the MDP factorization enables the efficient simultaneous RP learning of the agent value functions to achieve a policy which is guaranteed to optimize the overall reward objective.

Simulation Environment

JavaScript Weed World

- Implements the Weed World Environment as a grid world of 85 rows of 2.5 foot squares, totaling 4400 square feet or one acre.
- Includes realistic weed generation with existing weeds growing at a fixed rate and new weeds spontaneously being seeded by weeds above a certain height.
- Allows for real-time visualization in at varying speeds, plotting the reward at each state.
- JavaScript implementation allows portable visualization and allows for the possibility of on line deployment for educational purposes.

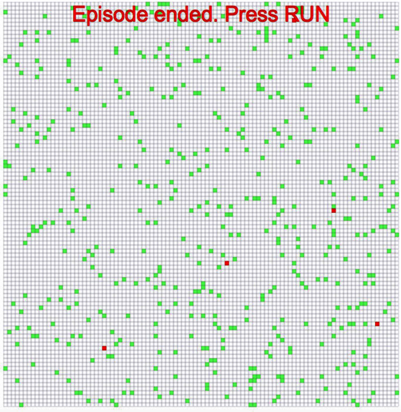

Figure 1: Simulation Environment

Experiment 1

- To establish a baseline, we first assume full observability, giving the planner full knowledge of the locations of all of the weeds.
- We then assume that the environment is partially observable, giving the planner knowledge of the weeds adjacent to squares the robots have passed before as the simulation runs.
- We compare the performance of RP approach under full observability to that of partial observability.

Experiment 1

Full Vs. Partial Observability - RP Approach

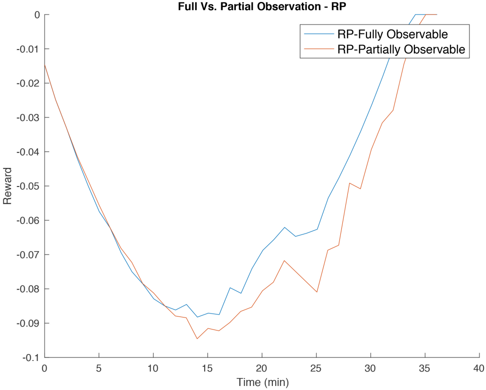

Figure 2: Full Vs. Partial Observability - RP

- We see that the performance of the RP approach does drop for the partially observable case, as expected.

Next Steps

- In these experiments, the RP approach uses Sequential Information Gain (SIG) when there is no prior information in the partially observable case, simply going to the next available adjacent unexplored row.
- We would like to explore the case of targeted Information Gain (TIG), inspired by [**?**], to see if this improves performance.
- We aim to give exploration preference to rows near those known to have high reward, and characterize the change in performance for the RP approach.

Targeted Information Gain (TIG

```
Input: $x_i(t)$: state of agents
Input: $R(x)$: reward for each row
Input: $N_w(x)$: number of weeds in each row
Output: $a_i(t)$: action for each agent
  Step 1: Rank all unexplored rows adjacent to those previously explored by the value of the adjacent row. Call this ranking the exploration value of the row.
  Step 2: If an agent is not assigned to a row and some rows have been explored, send the agent to the row with the highest exploration value.
  Step 3: If an agent is not assigned to a row and no rows have been explored, follow Sequential Information Gain (SIG).
```

Experiment 2

- We now compare the RP approach with sequential information gain to that with targeted information gain for the partially observable case.

Experiment 2

Sequential Vs. Targeted Information Gain

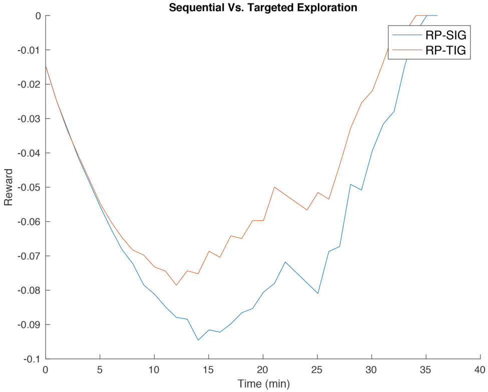

Figure 3: Sequential Vs. Targeted Information Gain RP Approach

- We see significant improvement with targeted information gain, as desired.

Discussion

Performance of Reactive Policy

- We have demonstrated here that we are able to learn an optimum reactive policy for our Weed World environment.
- Performance may be improved if the entire policy is learned in real-time (if the optimal sequence of rows for each agent is learned at every step based on available information).

Challenges and Future Work

Hybridized Targeted Information Gain and Neural Network Approach

- Due to the size of the state space, this approach will require a Neural Network framework, and will thus not be guaranteed to find an optimal solution.
- We hypothesize that the highest performing algorithm will utilize targeted information gain while gathering information and training the neural network, using the trained policy when there is sufficient information.
- We will next extend our simulation to utilize neural networks in order to test this hypothesis.

References I

Amato, C., Chowdhary, G., Geramifard, A., Ure, N. K., and Kochenderfer, M. J. (2013).
Decentralized control of partially observable markov decision processes.
In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 2398–2405. IEEE.

Cao, Y. U., Fukunaga, A. S., and Kahng, A. (1997).
Cooperative mobile robotics: Antecedents and directions.
Autonomous robots, 4(1):7–27.

Liu, M., Sivakumar, K., Omidshafiei, S., Amato, C., and How, J. P. (2017).
Learning for multi-robot cooperation in partially observable stochastic environments with macro-actions.
CoRR, abs/1707.07399.