Complex Robotic Planning Using Reinforcement Learning
Yousuf M. Soliman
Carnegie Mellon University – ysoliman@andrew.cmu.edu

COMPLEX ROBOTIC PLANNING USING REINFORCEMENT LEARNING

1. Introduction

Robotic interaction with complex environments has become increasingly difficult. Many challenges arise with the increased complexity of real world environments; one of the most complex challenges is to deal with the fact that data is often incomplete and censored due to the physical restrictions of the robotic sensors. Many approaches have been well studied to reduce the disturbances of the uncertainty. These approaches include using naïve feedback loops or formulating the problem as a Markov decision problem (MDP) [2, 6].

It is known that this type of problem may be modelled as a Markov Decision Problem or a Partially Observable MDP (POMDP), but these approaches also have significant drawbacks [1]. MDPs have not been extensively applied to robotics due to the fact that they make assumptions about the continuity of the state space. Thus, since their uses have not been well studied in robotics we do not know which problems they would be the most well suited for.

In this paper I will consider the problem of a multiobject manipulation planning problem with the state of the environment is estimated imperfectly using sensors. The covariate vector will be paritally unknown. The goal of this research is to move towards full autonomy of robotic movement and decision making in complex environments when all of the data is *not* available to the robot or operator.

The main theoretical contribution of this paper is an application of a novel reinforcement learning algorithm applied to robotic planning. It is important to note that the parameters do not need to lie in a well defined, finite discrete space.

The novel $\mathcal{Q}$-learning algorithm is experimentally evaluated in simulation and compared to other state of the art techniques. It was found that if some of the following properties hold, it may be beneficial to employ the proposed censored reinforcement learning techniques to improve the long term rewards of the robotic planning:

(1) actions must be taken to gain information about the system dynamics
(2) the system dynamics are uncertain and need to be updated at each new time step (e.g. some objects become more difficult to manipulate based off of their individual properties)
(3) the problem can be formulated as a multistage decision problem

Overall, the paper is the first to propose long term reinforcement learning for manipulating many objects in high dimensional, unknown and complex environments.

2. Related work

Dynamic treatment policies are becoming accepted and used in choosing effective actions for robotic planning. A policy is a set of rules for determining the course of action at a specific time point, depending on both the history of the object interaction and the system dynamics model up until the given time. Although the same set of decision rules is applied to each scenario, the action choice at a given time step will differ due to the differences in the state of the system. The optimal policy is the set of actions which maximize the rewards at the end of the running time.

The task is to determine the set of policies that will effectively plan robot actions in complex environments with uncertain robotic actions. Censoring is natural in complex environments, where the next action depends on the system dynamics manipulation and due to the fact that many aspects of the system dynamics may be censored, or occluded.

Reinforcement learning (RL) is a subset of machine learning which is used to solve dynamic decision problems. RL involves analyzing possible actions, estimating the statistical relationship between the actions and their possible outcomes and then determining a policy that attempts to find the most desirable outcome based on the analysis.

Finding a policy that results in a high expected accuracy and control and low expected error from the goal is the main goal of RL. Naïvely, one could attempt to learn the transition matrices and the reward models using the observed trajectories, and then attempt to solve the reverse Bellman equation. However, this becomes compuationally taxing and inefficient when there is a high dimensional state space and the relationship between the covariate parameters are confounded or uncorrelated.

One of the most prominent tools used in developing these treatment policies is $\mathcal{Q}$-learning [13]. Since the algorithm does not assume the Markov property, where it only is dependent on the previous time step, $\mathcal{Q}$-learning is well suited for determining actions that maximize long term rewards. The recursion used by $\mathcal{Q}$-learning addresses the problems that arise in terms of incorporating information accumulated over time into the policy, as well as avoiding "greedy" treatments which appear optimal in the near future, but have poor expected outcome in the long term.

Sutton and Barto consider $\mathcal{Q}$-learning to be one of the most important developments in RL [11]. $\mathcal{Q}$-learning utilizes recursion without needing to know the full dynamics of the world model.

In a previous paper, novel censored $\mathcal{Q}$-learning and SVM algorithms were developed [5]. I will briefly recall the results here for completion. I will then discuss their implemenation in robotic manipulation planning.

2.1. **Censored SVM.** To develop an effective *censored* reinforcement learning algorithm we must first have a way of estimating the unobserved covariate vectors. A censored SVM algorithm was created to acheive this task.

Let D be the data set of n i.i.d. random triplets of right censored data. Let $L : Z \times \mathcal{Y} \times \mathbb{R} \mapsto [0, \infty)$ be a convex locally Lipschitz continuous loss function. Let H be a seperable reproducing kernel Hilbert space (RKHS) of a bounded measurable kernel in $\mathcal{Z}$. We would like to find an empirical SVM decision function. Formally, we attempt to find the minimizer, f, of

$$\lambda \|f\|_H^2 + \mathcal{R}_{L,D}(f) \equiv \frac{1}{n} \sum_{i=1}^{n} L(Z_i, Y(T_i), f(Z_i)) \tag{1}$$

where $\lambda > 0$ is a fixed constant, and $Y : \mathcal{T} \mapsto \mathcal{Y}$ is a known function. The issue arises when we see that the failure times T_i may be censored, and thus unknown. Simply, we could ignore the censored observations, but this has been shown to induce severe bias [12].

To avoid this bias, the algorithm that reweights the *uncensored* observations. It is evident that at the time T_i, the i^{th} observation has a probability of $G(T_i^- | Z_i) = P(C_i \geq T_i | Z_i)$ of *not* being censored. Thus, we are able to use the inverse of the censoring probability for reweighting [10].

Specifically, the loss function is redefined to be the random loss function $L^n : (\mathcal{Z} \times \mathcal{T} \times \{0,1\})^n \times (\mathcal{Z} \times \mathcal{T} \times \{0,1\} \times \mathbb{R}) \mapsto \mathbb{R}$ by:

$$L^n(D, (z, u, \delta, s)) = \begin{cases} \frac{L(z, Y(u), s)}{\hat{G}_n(u|z)} & \text{if } \delta = 1 \\ 0 & \text{if } \delta = 0 \end{cases} \tag{2}$$

where $\hat{G}_n$ is the estimator of the survival function of the censoring variable based on the set of n random triplets in D. When D is given, $L_D^n(\cdot) \equiv L^n(D, \cdot)$. In [5] it is shown that L_D^n is a measurable function and that the empirical SVM decision function is well defined and will converge optimally *even* in censored and complex environments. Furthermore, we find that ineffeciency in estimating the covariate vector does not lead to a severe bias in the results.

2.2. **Censored $\mathcal{Q}$-learning.** We can find the near optimal policy $\hat{\pi}$ in three steps. First, we transform the problem to the corresponding auxiliary problem to deal with the variable length treatment paths. Then, we approximate the functions $\{Q_1^*, \ldots, Q_T^*\}$ using backwards recursion and obtain the functions $\{\hat{Q}_1, \ldots, \hat{Q}_T\}$. Finally, we define $\hat{\pi}$ by maximizing $\hat{Q}_t(\mathbf{s}_t, (\mathbf{a}_{t-1}, \mathfrak{a}_t))$ over all possible $\mathfrak{a}_t \in \mathfrak{A}$, where $\mathfrak{A}$ is the universe of all actions.

Rather than choosing the $\hat{Q}_t$ using the standard $\mathcal{Q}$-learning algorithm due to the fact that data is subject to censoring and we have a variable length treatment path, we use an alternative recursive minimizer as follows:

$$\text{(3)} \qquad \text{argmin}_{Q_t \in \mathcal{Q}_t} \mathbb{E}_n \left[\left(R_t + \max_{a_{t+1}} \hat{Q}_{t+1}(\mathbf{S}_{t+1}, (\mathbf{A}_t, \mathfrak{a}_{t+1})) - Q_t(\mathbf{S}_t, \mathbf{A}_t) \right)^2 \frac{\delta_t}{\hat{S}_C(\sum_{i=1}^t R_i)} \right]$$

where $\hat{Q}_{T+1} \equiv 0$ and $\hat{S}_C$ is the Kaplan-Meier estimation function of the survival of the censoring variable, C, using the censored SVM algorithm described above.

It then directly follows to define the policies using the approximated $\mathcal{Q}$-functions as follows:

$$\hat{\pi}(\mathbf{s}_t, \mathbf{a}_{t-1}) = \text{argmax}_{a_t} \hat{Q}_t(\mathbf{s}_t, (\mathbf{a}_{t-1}, \mathfrak{a}_t))$$

A framework for solving multistage decision problems with a variable number of stages that may be censored was studied in depth.

2.3. **Manipulation under uncertainty.**

Manipulating objects under uncertainty is not a new problem in robotic research. In the early 1980's, Lozano-Péres et al. [8] considered the automatic creation of fine robotic motions using backward chaining. An extensive summary of the research can be found in [7].

Recent work by Dogar and Srinivasa [3] proposes manipulation of multiple objects using primitive robotic actions. The planning is performed at the level of object position with no semantic information. Monso et al. [9] proposed an environment specific POMDP approach to object manipulation. In contrast, the work that I propose is non-environment specific and can be generalized to any type of multistage decision problem in robotics.

3. Multi-object manipulation using $\mathcal{Q}$-learning

In this paper, I study the problem where robots may control and manipulate objects, or use them in another way to accomplish some set of goals. I focus on deciding how to optimally manipulate objects in a complex environment. Due to the complexity of the environment, only parts of the objects can be observed by sensors. In addition to the uncertainty of the observations, real world manipulaitons induce further uncertainty, especially when there is no model to estimate how objects will respond to robotic actions.

Manipulation planning is traditionally consider a geometrical problem when the system dynamics are known. However, when the environment is complex with unknown objects, the current techniques often plan actions solely based on the observed environment [4]. I extend this notion by planning on the level of semantic actions and locations, which the execution of individual actions is then performed based on the direct positions of the objects. In addition, the algorithm I created also models the connection of the sensor measurments to both observational and semantic models.

I now present a general $\mathcal{Q}$-learning framework for modeling multi-object manipulation. I will then go on to show the implementation in a real world robotic simulation.

In multiobject manipulation, the robot has to decide which objects should be manipulated. I consider problems where the world consists of N objects with varying attributes. A_i is the set of all possible actions for the i^{th} object. Instead of forcing the robotic planning into a manageable discrete state space as is done in approaches such as [9], I use a censored SVM method based on the censored covariate vector that allows the algorithm to estimate complex system information required for efficient and effective multiobject manipulation. The censored SVM techniques is abstracted to allow for varying length covariate vectors.

The state space consists of semantic object locations, object attributes and historical data of observations and action successes for each object. The model naïvely assumes that the objects are stationary and do not change their semantic location without robot action being taken; however, the direct positions may change independently.

The observation history contains information of past observations of object attributes. Past object attribute observations are used to compute SVM regression models to estimate the probability of success of the subsequent action. Actions do not necessarily permanently move objects or change their semantic location; for example, in the experiments objects can be temporarily lifted to gain new information about the system dynamics or positions of the objects.

I assume that censoring effects the grasp probability of all objects in a similar way, but, in addition, that each object has unknown properties that affect the grasp probability of that specific object. For example, it may be more difficult to grasp an object that has fallen down rather than one that is still upright. The probability of a successful grasp can be modelled as follows:

$$P(\text{grasp succeeded}|s_i^{\text{occluded}}, s_i^{\text{history}}) = \mathbb{E}[p_i^{\text{successful}}]^1$$

Naïvely I can model the grasp probability as the mean of the Beta distributed random variable $p_i^{\text{successful}}$. However, I show that the SVM algorithm developed for determining whether the grasp action will succeed was a more successful approach.

3.1. **Experiment Setup: Moving cups.** I now demonstrate how the framework can be used to model the problem of moving cups from a table into a dishwasher as a reinforcement learning problem. In this problem, the robot can gain more information about the attributes by removing occlusions and gaining more information about the system dynamics through attempting subsequent actions.

3.1.1. *State space and actions.* In addition to the grasping and observation history, the world state also consists of semantic locations $s_i^{\text{location}} = \{\texttt{TABLE, DISHWASHER}\}$, and the attributes of the cups are $\{\texttt{CLEAN, DIRTY}\}$. The robot can perform three types of actions.

(1) `WASH` attempts to move an object from the table into the dishwasher. A successful `WASH` is given a large positive reward; an unsuccessful `WASH` is given a smaller, but still significant, negative reward.
(2) `LIFT` attempts to lift an object to gain more information about the system dynamics. `LIFT` is given a negative reward that is a function of the length it takes for the action to complete.
(3) `FINISH` terminates the robot actions. Negative rewards are given for each dirty object left on the table.

3.2. **Simulation Dynamics.** Due to the uncertainty in actions and observations, real robotic decision are often much more complex than this simple example can show or imply.

It directly follows that the null hypothesis is that a heuristic greedy manipulation approach is as sufficient as the censored $\mathcal{Q}$-learning model ($H_0 : \mu_{R(\mathcal{Q})} = \mu_{R(\text{heuristic})}$). The alternative hypothesis is $H_a : \mu_{R(\mathcal{Q})} > \mu_{R(\text{heuristic})}$. To test this hypothesis I experimentally compared heuristic manipulation with the propesed reinforcement learning model. I also compare the $\mathcal{Q}$-learning model to POMDP approaches with different planning horizons. Due to the fact that experiments with a physical robot are not repeatable I performed experiments with simulated world dynamics. Even though I simulate world dynamics, I estimate the grasp and observation probabilities using the physical robot arm and real observed occlusions. Morever, I estimate the occlusions and locations of objects from point clouds captured from the Asus Xtion PRO sensor.

In the simulated dynamics experiments, I setup ten different cup configurations and their respective caputured point clouds as the initial state for simulations. In these experiments, I form a system dynamic model from the point cloud and then repeatedly sample an initial belief and simulate the system using the probability model for 10 time steps.

[1] $\mathbb{E}$ represents the emperical expecation

3.2.1. *$\mathcal{Q}$-learning implementation.* The $\mathcal{Q}$-learning algorithm was developed in MATLAB.

I implemented the algorithm as follows: The input is a set of point clouds obtained based on the system dynamics described previously. First I compute the Kaplan-Meier estimator for the survival function $\hat{S}_C$ of the censoring variable from the observed history. Then, I set $\hat{Q}_{T+1} \equiv 0$ and I compute $\hat{Q}_i$ in reverse order as the minimizer over the functions $Q_i(s_i, a_i)$. Then the policy $\hat{\pi}$ is calculated from the functions in $\{\hat{Q}_i\}$.

3.3. **Results.** First, I examine the effect of both the sample size and the percentage of occlusion on performance. I simulated data sets containing treatment paths of sizes. For each set of treatment paths I considered four possible degress of censoring ranging from no censoring to 50% censoring. A policy $\hat{\pi}$ was calculated for every combination of the size of data set and percentage of censoring. I repeated the simulation 10,000 times for every combination to account for the probabilistic aspect of the algorithm. From Figure 1, it is evident that the approximation policy obtained by the algorithm are superior than any greedy heuristic method, even if it incorporates history. I can also see that, for all percentages of censoring, there is a direct positive correlation between the number of observed treatment paths and the anticipated reward of object manipulation. In Table 1 I compare the speed and accuracy of the algorithms with $\alpha = 0.01$.

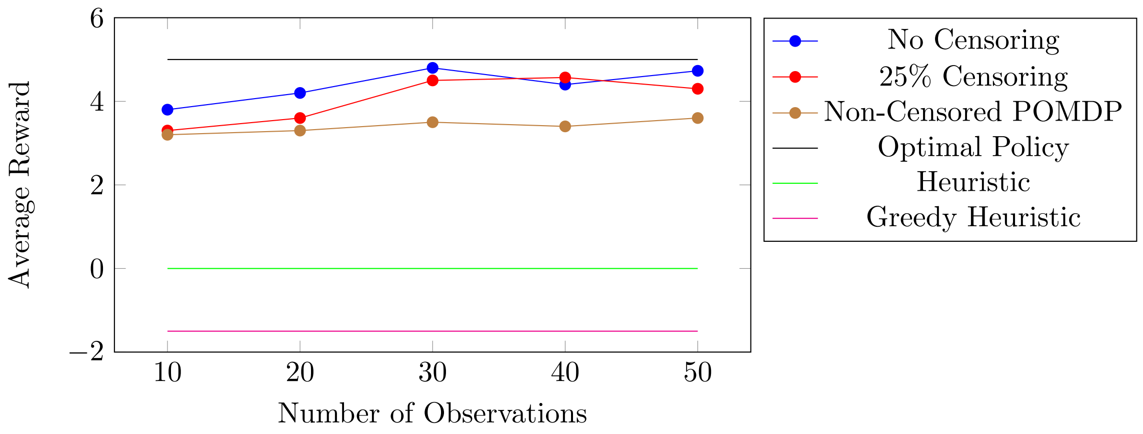

FIGURE 1. The expected reward was calculated as the mean of 10,000 repetitions of the simulation.

Time c$\mathcal{Q}$L vs. POMDP	p-value	Accuracy c$\mathcal{Q}$L vs. POMDP	p-value
H_0 : Both equally fast	4.757×10^{-7}	H_0 : Both equally accurate	9.89×10^{-3}
H_1 : c$\mathcal{Q}$L faster		H_1 : c$\mathcal{Q}$L more accurate	
Time c$\mathcal{Q}$L vs. Greedy Heuristics	p-value	Accuracy c$\mathcal{Q}$L vs. Greedy Heuristics	p-value
H_0 : Both equally fast	1.526×10^{-5}	H_0 : Both equally accurate	6.19×19^{-3}
H_1 : c$\mathcal{Q}$L faster		H_1 : c$\mathcal{Q}$L more accurate	
Time c$\mathcal{Q}$L vs. Heuristics with History	p-value	Accuracy c$\mathcal{Q}$L vs. Heuristics with History	p-value
H_0 : Both equally fast	5.528×10^{-6}	H_0 : Both equally accurate	3.873×10^{-4}
H_1 : c$\mathcal{Q}$L faster		H_1 : c$\mathcal{Q}$L more accurate	

TABLE 1. Statistical tests of the speed and accuracy of the algorithm in comparison to other techniques for multioject manipulation

4. Discussion and Conclusions

In conclusion, I presented a novel implementation of the censored Q-learning algorithm I developed for multiobject manipulation in complex and crowded real world environments. Due to the fact that objects are occluded, or data is subject to censoring, the objects are more difficult to manipulate. To address this problem, the censored SVM technique developed allows us to maintain a belief about the interaction dynamics of the objects.

The experiments provide us with enough evidence to reject the null hypothesis. They confirm that the censored reinforcement learning algorithm is better suited for multistage robotic object manipulation. With uncertain world dynamics, the censored SVM algorithm provides us with the ability to estimate and maintain a belief about the world dynamics. Since I employ reinforcement learning, the robot often will perform actions that do not yield immediate reward in exchange for more information about the objects.

Future lines of research include extending the framework to multiple robots worlking on multistage multiobject manipulation. I am currently working on moving past simulations and creating an actual robot that can employ these techniques. In contrast to geometric approaches or engineered solutions, machine learning offers novel solutions to robotic problems that generalize to unexpected situations and are truly autonomous.

The goal of fully autonomous robots acting in complex environments can only be achieved through employing advanced machine learning techniques and increased sensor sophistication in robotic research. The research presented provides a new foundation for further work towards this goal.

References

[1] Christopher Amato, Daniel S Bernstein, and Shlomo Zilberstein. Optimizing fixed-size stochastic controllers for pomdps and decentralized pomdps. *Autonomous Agents and Multi-Agent Systems*, 21(3):293–320, 2010.
[2] Karl Johan Åström and Tore Hägglund. *Advanced PID control.* ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006.
[3] Mehmet R Dogar and Siddhartha S Srinivasa. A planning framework for non-prehensile manipulation under clutter and uncertainty. *Autonomous Robots*, 33(3):217–236, 2012.
[4] David Fischinger, Markus Vincze, and Yun Jiang. Learning grasps for unknown objects in cluttered scenes. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 609–616. IEEE, 2013.
[5] Yair Goldberg and Michael R. Kosorok. Q-learning with censored data. *The Annals of Statistics*, 40(1):529–560, 02 2012.
[6] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.
[7] Steven Michael LaValle. *Planning algorithms.* Cambridge university press, 2006.
[8] Tomas Lozano-Perez, Matthew T Mason, and Russell H Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984.
[9] Pol Monsó, Guillem Alenyà, and Carme Torras. Pomdp approach to robotized clothes separation. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1324–1329. IEEE, 2012.
[10] James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427):846–866, 1994.
[11] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning.* MIT Press, 1998.
[12] Anastasios Tsiatis. *Semiparametric theory and missing data.* Springer, 2007.
[13] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.