Multi-Armed Bandit Algorithms for Spare Time Planning of a Mobile Service Robot

Robotics Track

Max Korein
Robotics Institute, Carnegie Mellon University
Pittsburgh, USA
mkorein@cs.cmu.edu

Manuela Veloso
Machine Learning Department, Carnegie Mellon University
Pittsburgh, USA
mmv@cs.cmu.edu

ABSTRACT

We assume that service robots will have spare time in between scheduled user requests, which they could use to perform additional unrequested services in order to learn a model of users' preferences and receive reward. However, a mobile service robot is constrained by the need to travel through the environment to reach a user in order to perform a service for them, as well as the need to carry out scheduled user requests. We present modified versions of Thompson Sampling and UCB1, existing algorithms used in multi-armed bandit problems, which plan ahead considering the time and location constraints of a mobile service robot. We compare them to existing versions of Thompson Sampling and UCB1 and find that our modified planning algorithms outperform the original versions in terms of both reward received and the effectiveness of the model learned in a simulation.

KEYWORDS

Machine learning for robotics; Robot planning and plan execution; Robot autonomy

ACM Reference Format:
Max Korein and Manuela Veloso. 2018. Multi-Armed Bandit Algorithms for Spare Time Planning of a Mobile Service Robot. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

Consider a service robot in an office building that performs services for users. Users can request services, but the robot can also approach a user's office to offer a service that they have not requested. It would be desirable for such a robot to perform as many successful services as possible, but some users in a given building might be more interested in receiving the robot's services when they have not requested them than others.

In order to maximize the service it provides, the robot would have to learn a model of how likely different users in the building are to be interested in its services when they have not made any requests. Then, during the time between scheduled service requests, it would choose which services to perform on the way to its next user request, striking a balance between learning more about users' preferences and offering services that it knows are likely to succeed. Its ability to do this is constrained by the need to spend time traveling to users' offices to offer services, as well as scheduled requests that must be performed at specific locations and times.

Previous work with service robots operating in their spare time has focused on robots that gather knowledge they can use to perform scheduled requests. For example, the Dora the Explorer robot can search for information about where objects can be found that might be requested by users in the future[3]. In our own previous work, we studied how the CoBot mobile service robots can use their spare time to observe when users' doors are open in order to improve their ability to schedule user-requested services[4, 5].

In this paper we explore how a service robot can gather information in its spare time and use that information to perform additional services for users. We treat the problem as a variant of the multi-armed bandit problem (MAB). In a MAB, an agent can pull a variety of levers, each of which yields a reward from an unknown random distribution, and must maximize the reward received over time. We focus on two existing MAB algorithms – Thomspon Sampling, a probability matching algorithm, and UCB1, an optimistic algorithm [1, 2, 6] – and adapt them to plan ahead, taking into account the constraints of a service robot operating in its spare time. We show that our adapted algorithms outperform the standard versions of the algorithms used with regular MABs.

2 PROBLEM DESCRIPTION

The environment is represented by an undirected graph. Nodes on the graph represent offices where the robot can go to offer services to users, and edge lengths represent the time to travel between adjacent offices. The robot has a complete and accurate map, and thus can find the shortest path between any two locations and accurately predict how long it will take to travel along that path.

We will divide the robot's time into the segments between scheduled user requests. During a segment, the robot begins at location L_{start} and time $t = t_{start}$, the location and time at which it completed its last scheduled request. We will assume $t_{start} = 0$ for the sake of simplicity. The robot must arrive at the location of its next scheduled request, L_{end}, by time t_{end}.

In the meantime, the robot can offer services to users. To do so, it must travel to a user's office and interact with the user. Each user u has some probability p_u of accepting services offered by the robot. The robot begins with no knowledge of p_u for any user.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden. © 2018 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

This research is partially sponsored by DARPA under agreements FA87501620042 and FA8750172015 2. The views and conclusions contained in this document are those of the authors only.

If the user accepts the service, the robot receives a reward of 1, otherwise it receives a reward of 0. This process takes a time of $t_{service}$, regardless of whether the service is accepted or rejected. The robot may only offer a service to each user once per segment. Once it has offered a service to a user, whether it succeeded or failed, it may not offer a service to the same user again until after it has completed its next scheduled request.

The goal is to maximize the reward the robot receives over time. In order to do this, the robot must both learn a model of the likelihood of a service being successful for any given user, and manage its time during each segment as it travels from user to user offering services while ensuring that it reaches L_{end} in time.

3 ALGORITHMS

In this section, we describe the algorithms we created based on two MAB algorithms, Thompson Sampling and UCB1. For each algorithm, we describe the standard MAB version of the algorithm, then contribute our own planning version which better takes into account the constraints of the spare-time planning problem.

3.1 Planning Thompson Sampling

Thompson sampling is a probability-matching algorithm, in which the probability the robot chooses an action is equal to the probability that the action is optimal. This is accomplished by sampling from the distribution of possible expected rewards based on the samples the robot has gathered from each office.

For a given office u_i, the function $B_i(n_1, n_0)$ gives the probability distribution of the true expected reward of that office given that the robot has performed successful services for that office n_1 times and failed services n_0 times, where $B(x, y) = \frac{(x-1)!(y-1)!}{(x+y-1)!}$ is the beta function. If the robot samples a reward from $B_j(n_1, n_0)$ for each office u_j, then the probability of office u_i having the highest sampled reward is equal to the probability that u_i has the highest true expected reward of any office given the robot's data.

The standard MAB version of Thompson Sampling works by repeatedly sampling a reward from each possible action and performing the action with the highest sampled reward. In the context of the spare-time service problem, the robot chooses the office with the highest sampled reward that is has time to visit within the constraints. It travels to that office and offers a service, then repeats the process until it does not have time to perform any more services, and travels to the end location. We refer to this as **Naive Thompson Sampling**.

We contribute an algorithm that considers the constraints of the problem, which we call **Planning Thompson Sampling**. With Planning Thompson Sampling, the robot samples a reward from every office in the environment, finds the plan of services that maximizes the sampled rewards received that meets the time and location constraints, and executes that plan.

3.2 Planning UCB1

UCB1 is an optimistic algorithm that intentionally overestimates the expected reward of an office based on the uncertainty of its model of that reward. That way, the robot will be biased towards gathering data from offices for which the reward is highly uncertain in order to improve its model. Specifically, the robot assigns each office u_i a value of $r' = r + \beta\sqrt{\frac{2 \ln n}{n_i}}$, where r is the average observed reward from that office, n is the total number of attempted services the robot has ever made, n_i is the number of attempted services the robot has made for office u_i, and β is a tuning parameter.

As with Thompson sampling, we refer to the standard MAB version of UCB1, which always offers a service to the office that it has time to reach with the highest optimistic reward, as **Naive UCB1**. We contribute **Planning UCB1**, which finds the plan that will receive the maximum optimistic reward in the available time.

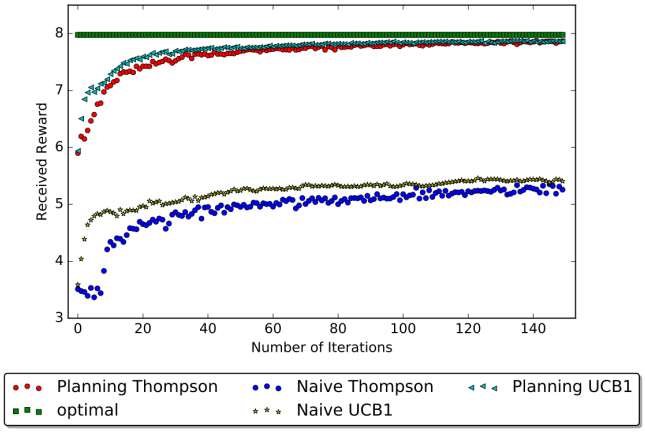

Figure 1: The reward received by the robot. Planning Thompson Sampling and Planning UCB1 had the best performances.

4 EXPERIMENTAL RESULTS

We compared the algorithms described in Section 3 in simulation. The environment consisted of six hallways, each containing six offices spaced a distance of 15 apart, radiating out from a center room. The value of $t_{service}$ was 20, t_{end} was 500, and L_{start} and L_{end} were always the center office. Both versions of the UCB1 algorithm used a value of 0.1 for β, determined empirically. Figure 1 shows the reward received by the robot using each algorithm, averaged over 200 trials.

As can be seen, the two planning algorithms received significantly more reward than the two naive algorithms. Running the same experiment in environments with as few as four or as many as eight hallways gave similar results. Additionally, we analyzed the model learned by the robot, judging it based on the expected reward of the best plan according to the robot's model – in other words, how effectively the robot's model could be used to find an effective plan if the robot assumed its model was perfectly accurate. By this metric we similarly found that the two planning algorithms performed significantly better than the naive ones.

These results show that, in a specific structured environment, our modified versions Thompson Sampling and UCB1 significantly outperform the naive versions used in standard MAB problems for a service robot performing services in its spare time. In fact, our planning algorithms received a near-optimal reward on average within 150 iterations. These algorithms are very promising and we will test and improve them further in our future work.

REFERENCES

[1] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[2] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*. 2249–2257.

[3] Marc Hanheide, Nick Hawes, Jeremy Wyatt, Moritz Gobelbecker, Michael Brenner, Kristoffer Sjoo, Alper Aydemir, Patric Jensfelt, Hendrik Zender, and Geert-Jan M. Kruijff. 2010. A Framework for Goal Generation and Management. In *Proceedings of the AAAI Workshop on Goal-Directed Autonomy*.

[4] Max Korein, Brian Coltin, and Manuela Veloso. 2013. Scheduling mobile exploration tasks for environment learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1255–1256.

[5] Max Korein, Brian Coltin, and Manuela Veloso. 2014. Constrained scheduling of robot exploration tasks. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 429–436.

[6] Steven L Scott. 2010. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry* 26, 6 (2010), 639–658.