A Contextual Bandit Framework for Adaptive Robotic Bite Acquisition

Ethan K. Gordon[1], Xiang Meng[2], Tapomayukh Bhattacharjee[1], Matt Barnes[1] and Siddhartha S. Srinivasa[1]

***Abstract*— Successful robot-assisted feeding requires bite acquisition of a wide variety of food items. Different food items may require different manipulation actions for successful bite acquisition. Therefore, a key challenge is to handle previously-unseen food items with very different action distributions. By leveraging contexts from previous bite acquisition attempts, a robot should be able to learn online how to acquire those previously-unseen food items. We construct an online learning framework for this problem setting and use the ϵ-greedy and LinUCB contextual bandit algorithms to minimize cumulative regret within that setting. Finally, we demonstrate empirically on a robot-assisted feeding system that this solution can adapt quickly to a food item with an action success rate distribution that differs greatly from previously-seen food items.**

I. INTRODUCTION

Different food items may require different manipulation actions for bite acquisition [1] and a robust system needs to be able to acquire these myriad types of food items that a user might want to eat. While we have achieved some recent successes in developing manipulation actions that can acquire a variety of food items [2], [3], a key challenge is to acquire previously-unseen food items that have very different action distributions. Even food items that look similar, such as ripe and un-ripe banana slices, can have very different consistencies. Our key insight is that by leveraging contexts from previous bite acquisition attempts, an autonomous system should be able to learn online how to acquire these previously-unseen or changing food items. Our major contribution in this work is a contextual bandit framework for bite acquisition in an online learning setting. We propose the use of the ϵ-greedy [4] and LinUCB [5] contextual bandit algorithms, and we furthermore show empirically that these algorithms are effective in picking up new food items. Our current action space of 3 fork roll angles × 2 fork pitch angles limits us to discrete, solid food items, but future work can examine a richer action space to tackle bite acquisition on a more varied, realistic plate.

II. PREVIOUS WORK

Existing autonomous robot-assisted feeding systems such as [2], [3], [6], and [7] can acquire a fixed set of food items and feed people, but it is not clear whether these systems can adapt to very different food items that require completely different strategies. Feng *et al.* [2] developed a network SPANet and show generalization to previously-unseen food items, but only for those with similar bite acquisition strategies.

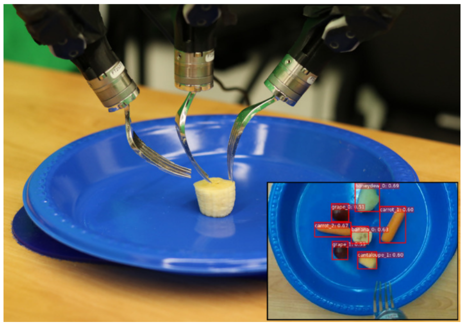

Fig. 1: Different strategies for picking up discrete food items. Each strategy provides a high-level description (i.e., *(left)* three fork pitches: tilted-angled (TA), vertical (VS), and tilted-vertical (TV); and *(right)* two fork roll angles), and still requires lower-level controllers and perception, such as object segmentation (inset).

For a recent and thorough overview of bandit algorithms, we refer the interested reader to [4], [8].

III. ONLINE LEARNING FRAMEWORK

At each round $t = 1, \ldots, T$, the interaction protocol consists of

1) *Context observation* The user selects a food item to acquire. We observe the resulting RGBD image. We pass the image through SPANet and use the penultimate layer as the context features $x_t \in \mathcal{X} = \mathbb{R}^d$.
2) *Action selection* The algorithm selects one manipulation strategy $a_t \in \mathcal{A} = \{1, 2, \ldots, K\}$. In our initial implementation, $K = 6$, with 3 pitch angles and 2 roll angles.
3) *Partial loss observation* The environment provides a binary loss $c_t \in \mathcal{C} = \{0, 1\}$, where $c_t = 0$ corresponds to a successful acquisition

The algorithm itself will consist of a stochastic policy $\pi(x_t) = \mathbb{P}(a_t = a|x_t)$, and the goal is to minimize the cumulative regret of this policy, R_T, which is the difference in performance between our policy π and the best possible policy $\pi^* \in \Pi$ for the lifetime of our program T. With $c_t \in \mathcal{C}$, $x_t \in \mathcal{X}$, $(n_t, a_t) \in \mathcal{A}$ at time t:

$$R_T := \sum_{t=1}^{T} c_t(\pi(\phi(x_t))) - \min_{\pi^* \in \Pi} \sum_{t=1}^{T} c_t(\pi^*(\phi(x_t)))$$

While we could potentially perform multiple actions on the same food item, each individual action only returns partial (or *bandit*) feedback. We are not privy to the rewards of other actions. Therefore, a contextual bandit algorithm is a natural choice.

[1] Ethan K. Gordon, Tapomayukh Bhattacharjee, Matt Barnes and Siddhartha S. Srinivasa are with the Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195 {ekgordon, tapo, mbarnes, siddh}@cs.washington.edu

[2] Xiang Meng is with the Department of Statistics, University of Washington, Seattle, WA 98195, xmeng051@uw.edu

This work was partially funded by the National Institute of Health R01 (#R01EB019335), National Science Foundation CPS (#1544797), National Science Foundation NRI (#1637748), the Office of Naval Research, the RCTA, Amazon, and Honda Research Institute USA. We would also like to thank Joshua Brockschmidt for his help with the robot experiments.

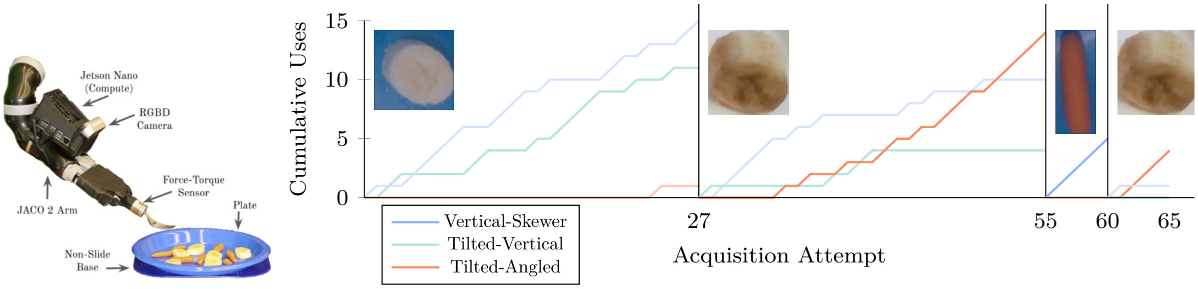

Fig. 2: Experimental validation of LinUCB *(Right)* using the Autonomous Dexterous Arm (ADA) *(Left)*. The robot acquired previously-unseen under-ripe bananas for the first 27 trials, which are firm enough to be picked up with vertical skewers. For the next trials with *ripe* bananas, the algorithm switched to the only viable action (tilted-angled) within 10 attempts. Attempts 50-55 were exclusively tilted-angled. Finally, we alternated between carrots (optimal action VS) and ripe banana (optimal action TA) to verify that the model did not over-fit.

IV. Contextual Bandit Algorithms

A. *ϵ-greedy*

This simplest approach to this problem is the ϵ-greedy algorithm. This algorithm opts for the optimal action based on previous observations with probability $(1-\epsilon)$ and explores all actions uniformly with probability ϵ. We consider both purely greedy ($\epsilon = 0$) and exploratory ($\epsilon > 0$) variants.

B. *LinUCB*

We propose the use of LinUCB [5], [9], due to (a) the linear regression form of the penultimate network layer and (b) the adversarial nature of the contexts. At each time step, we compute the reward upper confidence bound (UCB) for each action and choose the action with the highest UCB. This implicitly encourages exploration, as in a choice between two actions with similar expected cost, the algorithm will opt for the one with higher variance.

V. Experiments

a) Tuning in Simulation: We first tune the hyperparameters of our algorithms by constructing a simulated environment using the data from [2]. Since this data was collected with bandit feedback, the original work imputed the full loss vector of each context by averaging the success rate of a given action across all food items of the same type. While simple, this can introduce a herding bias into the simulation relative to the real world. We eliminate bias in our dataset using a doubly-robust [10] estimator. This estimator eliminates bias from our imputed values at the cost of added variance.

We found that LinUCB ($\alpha = 0.05$) was most robust to variation in hyper-parameters. Furthermore, the best result of the entire grid search was LinUCB with $d = 2048$, $\lambda = 0.1$. We therefore used this algorithm for our real robot experiment.

b) On-Robot Experimental Procedure: For each trial, we place a single food item in the center of the plate. ADA positions itself vertically above the plate and performs object segmentation, featurization, and action selection using a checkpoint of SPANet that has been trained on 15 food items, excluding banana. After performing the requested action, the binary loss is recorded manually and used to update the online learning algorithm. For consistency, regardless of success, we removed and replaced the food item after every attempt.

We conducted 27 trials on deliberately previously-unseen under-ripe bananas, 27 trials on *ripe* bananas, 5 trials on previously-seen carrots, and another 5 trials on ripe banana to test the online learning algorithm's ability to remember the optimal action for each type of food without over-fitting. The identity of the food item was *never* made available to the online learning algorithm.

VI. Results

The number of times the online learning algorithm selected an action, cumulative across each food item, is presented in Figure 2. The empirical success rate of VS and TV on the under-ripe banana was 33% and 27% respectively, not statistically significant from TA's *a priori* success rate of $\sim$ 30% based on the previously-seen 15 food items. Therefore, as expected, we see the online learning algorithm primarily stick with VS and TV.

With the ripe bananas, VS and TV both had a success rate of 0 while TA exhibited an 83% success rate. LinUCB began experimenting with TA after 7 trials, and after trial 20, it was almost exclusively choosing that action.

For the final 10 trials, the online learning algorithm demonstrated that it did not forget the optimal action for previously-seen food items. It performed the optimal action on carrot (VS, 90° rotation) 5 times in a row, and when returning to the ripe banana, it performed the optimal action (TA, any rotation) 4 out of 5 times.

VII. Discussion

One key takeaway from these results is that LinUCB is empirically robust in uncertain environments across a range of hyper-parameters, and its success suggests that a contextual bandit approach with discrete, dissimilar actions is a promising route to data-efficient adaptive bite acquisition. In future work, we intend to broaden the scope to multiple food items by considering the entire plate of food items as a single compound context, or switching food items if we expact to perform poorly on the current food item.

Finally, we only investigated discrete, solid food items. In order to generalize to a realistic average plate with continuous and mixed foods, we will need to expand to a richer action space. Since adding more action parameters will increase the size of the action space at a combinatorial rate, we could leverage similarities between actions by modeling each one as a coupled slate of actions [11].

References

[1] T. Bhattacharjee, G. Lee, H. Song, and S. S. Srinivasa, “Towards robotic feeding: Role of haptics in fork-based food manipulation,” *IEEE Robotics and Automation Letters*, 2019.

[2] R. Feng, Y. Kim, G. Lee, E. K. Gordon, M. Schmittle, S. Kumar, T. Bhattacharjee, and S. S. Srinivasa, “Robot-assisted feeding: Generalizing skewering strategies across food items on a realistic plate,” in *International Symposium on Robotics Research*, 2019.

[3] D. Gallenberger, T. Bhattacharjee, Y. Kim, and S. Srinivasa, “Transfer depends on acquisition: Analyzing manipulation strategies for robotic feeding,” ACM/IEEE International Conference on Human-Robot Interaction, 2019.

[4] A. Bietti, A. Agarwal, and J. Langford, “A contextual bandit bake-off,” Feb. 2018.

[5] C. Wei, L. Li, L. Reyzin, and R. E. Schapire, “Contextual bandits with linear payoff functions,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 208–214.

[6] D. Park, Y. K. Kim, Z. M. Erickson, and C. C. Kemp, “Towards assistive feeding with a General-Purpose mobile manipulator,” May 2016.

[7] L. V. Herlant, “Algorithms, Implementation, and Studies on Eating with a Shared Control Robot Arm,” Ph.D. dissertation, 2016.

[8] T. Lattimore and C. Szepesvari, *Bandit Algorithms*, 2019.

[9] Y. Abbasi-yadkori, D. Pál, and C. Szepesvári, “Improved algorithms for linear stochastic bandits,” in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2312–2320.

[10] M. Dudik, J. Langford, and L. Li, “Doubly robust policy evaluation and learning,” Mar. 2011.

[11] M. Dimakopoulou, N. Vlassis, and T. Jebara, “Marginal posterior sampling for slate bandits,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, pp. 2223–2229.