A Contextual Bandit Framework for Bite Acquisition in Robot-Assisted Feeding

Ethan K Gordon, Xiang Meng, Tapomayukh Bhattacharjee, Matt Barnes, Siddhartha S Srinivasa
Visual Features and Discrete Strategies

Image $\rightarrow$ Expected Success Rate for Each Strategy

"Robot-Assisted Feeding: Generalizing Skewering Strategies across Food Items on a Realistic Plate" (Feng et al., ISRR 2019)
The Incorrect Action Fails

- Discrete actions are dissimilar
- The wrong action is particularly likely to fail.
- Example: vertical skewering fails on banana
SPANet Has Difficulty With New Food Items

“Robot-Assisted Feeding: Generalizing Skewering Strategies across Food Items on a Realistic Plate” (Feng et al., ISRR 2019)
What about just collecting more data?
Collecting More Data Won’t Help
What about just collecting more *diverse* data?
Collecting More Diverse Data Won’t Help

![Graph showing SPANet Success Rate vs. Number of Classes in Training Set]

- **SPANet Success Rate**
- **Number of Classes in Training Set**
  - 8
  - 12
  - 15

(0 = Random Action)
How can the robot adapt to previously-unseen food items?
Why not use RL?

RL is Great for This...

And keep its balance while being pushed around

Because, in simulation, we can retry over and over
Food is Diverse and Hard to Model
Why not use RL?

And in the real world, if we fail...

It’s messy.
We Need Online Learning

- Observe State
- Choose Action
- Update Policy
- Observe Reward
We can model Bite Acquisition as a Contextual Bandit
What’s a Multi-Arm Bandit?

Select 1 Machine
Observe 1 Reward
Adding State: The Contextual Bandit

Observe $c_t$

Advance $t$

Update $\pi$

Policy: $\pi$; which could include:

- Function Approximating $c_t$
- Stochastic Behavior
- A Featurizer $\phi$
Our Goal is to Minimize Regret

Trade-off exploitation and exploration to minimize Cumulative Regret

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

Our Policy

Best Policy in Class:
Aside: The Contextual Bandit is RL

Full RL Problem

Contextual Bandit
One Solution, \( \varepsilon \)-Greedy

\[ \mathbb{P}(\text{explore}) = \varepsilon \]

\[ \mathbb{P}(\text{greedy}) = (1-\varepsilon) \]

- Start with \( c_t = f(x_t) \)
- Update with Regression
- **With Probability** \( \varepsilon \), choose a random action (explore).
- **With Probability** 1-\( \varepsilon \), choose the best action according to \( f \) (exploit).
- *Empirically competitive*

“A Contextual Bandit Bake-off” (Bietti et al., 2018)
One Solution, LinUCB

- **Start with Linear Map:**
  - $c_t = Ax_t$
- **Update with Linear Regression**
- **Choose the action with the highest upper confidence bound:**

\[ + \alpha \sqrt{\phi(x)^T A \phi(x)} \]

- **Sacrifices learning speed for improved regret.**

“A Contextual Bandit Bake-off” (Bietti et al., 2018)
What does this have to do with bite acquisition?
Bite Acquisition is a Contextual Bandit

Context: $x_t$

Observe $c_t$

Update $\pi$

Select **Single** Action

Policy (SPANet): $\pi$

Vertical Skewer
Bite Acquisition is a [Linear] Contextual Bandit

Context: $x_t$

Observe (Stochastic) $c_t$

Update $\pi$

Select Single Action

Vertical Skewer

Policy (SPANet): $\pi$
The Full System

Object Segmentation

Observe [Binary] Reward

Featurizer (SPANet)

Repeat

Linear Map

Update

Select Single Action

\[\mathbb{R}^d \rightarrow \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots \\ \vdots & \ddots & \vdots \\ a_{d,0} & a_{d,1} & \cdots \end{bmatrix} \rightarrow \mathbb{R}^K\]

Action Distribution

Success Rate

Action
That’s the theory, but how well does it do?
Simulating the System

Object Segmentation

Simulated Inputs

Featurizer (SPANet)

Doubly-Robust Estimated Rewards

Linear Map

Select Single Action

Repeat

Update

\[ \mathbb{R}^{d} \]

\[ \mathbb{R}^{K} \]

Observe [Binary] Reward

Reward

\[ a_{0,0}, a_{0,1}, \ldots \]

\[ : \ldots : \]

\[ a_{d,0}, a_{d,K} \]
All Algorithms Were Competitive

The graph shows the success rate estimate for different algorithms and conditions. The x-axis represents the number of simulated attempts, ranging from 0 to 300. The y-axis represents the success rate estimate, ranging from 0 to 1. The graph includes lines for LinUCB, $\epsilon = 0.5$, and greedy algorithms. The parameters $d = 2048$ and $\lambda = 100$ are specified.
LinUCB was the Most Stable

\[ d = 512, \lambda = 10 \]

![Graph showing success rate estimates with LinUCB, ε = 0.5, and Greedy strategies over the number of simulated attempts.](image)
Online learning learns that **TA** is the **optimal** action for previously-**unseen** ripe banana.
What did we do?

- Model Bite Acquisition as a Contextual Bandit
- Demonstrate the stability of LinUCB on this data
- Successfully adapt to a new food item on a real robot.
A Contextual Bandit Framework for Bite Acquisition in Robot-Assisted Feeding

Ethan K Gordon, Xiang Meng, Tapomayukh Bhattacharjee, Matt Barnes, Siddhartha S Srinivasa
Appendix: The Doubly-Robust Estimator

Start with an estimate: \( \hat{l}_a \)

\[
\hat{l}_{DR}(x_i, a) = \hat{l}_a + (l_i - \hat{l}_a) \frac{1(a_i = a)}{p(a_i | x_i)}
\]

“Doubly robust policy evaluation and learning” (Dudik et al., 2011)
Appendix: SPANet Full Data

“Robot-Assisted Feeding: Generalizing Skewering Strategies across Food Items on a Realistic Plate” (Feng et al, ISRR 2019)