Adaptive Submodular Maximization in Bandit Setting

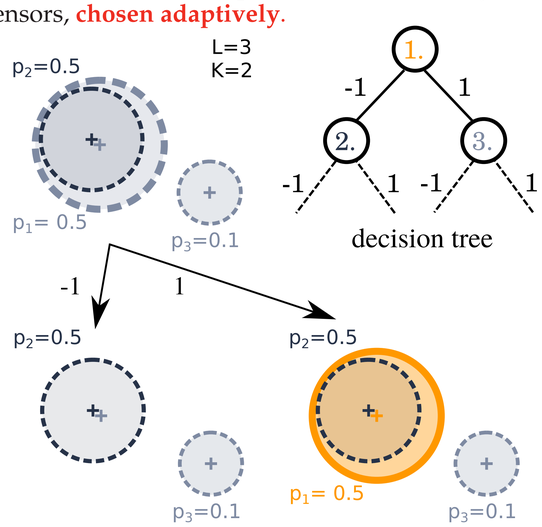
Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, S. Muthukrishnan

Abstract

- Maximization of submodular functions (SM) has wide applications in machine learning.
- Adaptive SM has been traditionally studied in the setting where the model of the world is known.
- We study the setting where the model is initially unknown, and it is learned by interacting repeatedly with the environment.
- Our work brings together the concepts of adaptive submodular maximization and bandits.

Adaptive Submodular Maximization

- Three sensors ($L = 3$) with known placements.
- The area covered by sensors is known.
- Upon choosing, each sensor i covers its area with probability p_i and the state of the sensor (active (1) or inactive (-1)) is **observed**.
- **Goal:** Maximize the **expected** area covered by two sensors, **chosen adaptively**.



Maximize a **real** function of the form:

$$f(\underbrace{A}_{\text{Set of chosen items}}, \underbrace{\phi}_{\text{State of all items}}) \to \mathbb{R}$$

The **state** $\phi \in \{-1, 1\}^L$ is drawn i.i.d. from a probability distribution $P(\Phi)$. The i-th entry of the state ϕ, $\phi[i]$, is the state of item i.
An **observation** is a vector $\mathbf{y} \in \{-1, 0, 1\}^L$.

$$\begin{aligned} \phi &= (-1, \ -1, \ 1, \ -1, \ 1) \leftarrow \text{State} \\ \mathbf{y} &= (0, \ -1, \ 0, \ -1, \ 1) \leftarrow \text{Observation} \\ \mathbf{y}' &= (0, \ 0, \ 0, \ -1, \ 0) \leftarrow \text{Observation} \end{aligned}$$

We say that $\mathbf{y} \sqsupseteq \mathbf{y}'$, $\phi \sim \mathbf{y}$, and $\phi \sim \mathbf{y}'$.
The observed items in $\mathbf{y}$ are $\text{dom}(\mathbf{y}) = \{2, 4, 5\}$.

- $\pi_k(\phi)$ are the first k items chosen by π in state ϕ.
- The optimal $\boldsymbol{K}$-step policy is defined as:

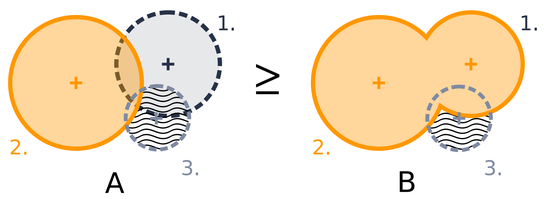
$$\pi^* = \arg\max_\pi \mathbb{E}_\phi[f(\pi_K(\phi), \phi)] \, .$$

- Computing π^* is NP-hard **BUT** ...

Function f is **adaptive submodular** if:

$$\mathbb{E}_\phi[\, f(A \cup \{i\}, \phi) - f(A, \phi) \,|\, \phi \sim \mathbf{y}_A \,] \geq \mathbb{E}_\phi[\, f(B \cup \{i\}, \phi) - f(B, \phi) \,|\, \phi \sim \mathbf{y}_B \,]$$

for all i, $\mathbf{y}_B \sqsupseteq \mathbf{y}_A$, $A = \text{dom}(\mathbf{y}_A)$, $B = \text{dom}(\mathbf{y}_B)$.



Function f is **adaptive monotonic** if

$$\mathbb{E}_\phi[\, f(A \cup \{i\}, \phi) - f(A, \phi) \,|\, \phi \sim \mathbf{y}_A \,] \geq 0$$

for all i and $A = \text{dom}(\mathbf{y}_A)$.

- Let π^g be the **greedy policy** for maximizing f, the policy that chooses the item with the highest expected gain:

$$\pi^g(\mathbf{y}) = \arg\max_{i \in I \setminus \text{dom}(\mathbf{y})} g_i(\mathbf{y})$$

where $g_i(\mathbf{y}) =$

$$\mathbb{E}_\phi[\, f(\text{dom}(\mathbf{y}) \cup \{i\}, \phi) - f(\text{dom}(\mathbf{y}), \phi) \,|\, \phi \sim \mathbf{y} \,]$$

is the *expected gain* of item i after observing $\mathbf{y}$.

- If f is adaptive submodular and monotonic, then π^g **is a $(1 - 1/e)$-approximation to π^*.**

The greedy policy cannot be computed when $P(\phi)$ is unknown!

Optimistic Adaptive Submodular Maximization (OASM)

Our approach: Mimic the greedy policy π^g while **learning** $P(\Phi)$.

Assumption: The state of each item is distributed independently of the other states:

$$P(\Phi = \phi) = \prod_{i=1}^{L} p_i^{\mathbb{1}\{\phi[i]=1\}} (1 - p_i)^{1 - \mathbb{1}\{\phi[i]=1\}}$$

- Learning L parameters instead of 2^K.

The expected gain can be written as $g_i(\mathbf{y}) = p_i \bar{g}_i(\mathbf{y})$, where $\bar{g}_i(\mathbf{y})$ is the expected gain of choosing item i in state 1 (area covered when sensor i is active).

Assumption: The gain $\bar{g}_i(\mathbf{y})$ is known and can be computed without knowing $P(\Phi)$ (quite common).

OASM

Play n episodes of a $\boldsymbol{K}$-step game.

for $t = 1, 2, \ldots, n$ **do**

Maximize $\boldsymbol{f}$ in $\boldsymbol{K}$ steps using the policy:

$$\pi^t(\mathbf{y}) = \arg\max_i \left(\hat{p}_{i, T_i(t-1)} + \sqrt{\frac{2 \log(t)}{T_i(t-1)}} \right) \bar{g}_i(\mathbf{y})$$

Update all statistics of the model

end for

Theoretical Results

Goal: Minimize the **expected cumulative regret**:

$$R(n) = \mathbb{E}_{\phi_1, \ldots, \phi_n} \left[\sum_{t=1}^{n} f(\pi^g_{\boldsymbol{K}}(\phi_t), \phi_t) - f(\pi^t_{\boldsymbol{K}}(\phi_t), \phi_t) \right].$$

Regret bound:

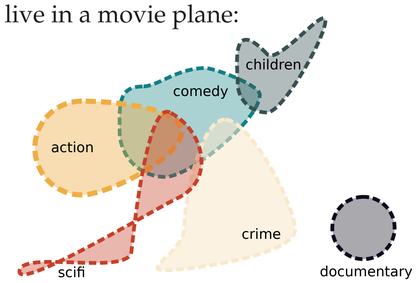
$$R(n) \leq \underbrace{\sum_{i=1}^{L} \ell_i \sum_{k=1}^{K} G_k \alpha_{i,k}}_{O(\log n)} + O(1),$$

where ℓ_i is the number of pulls after which arm i is unlikely to be pulled suboptimally, G_k is an upper bound on the expected gain of the greedy policy π^g from step k forward, and $\alpha_{i,k}$ is a weight that associates the regret to step k such that $\sum_{k=1}^{K} \alpha_{i,k} = 1$.

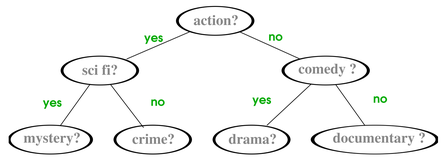
- Polynomial in both L and K.
- Captures the shape of the optimized function f.

Experiments on a Preference Elicitation Problem

Goal: Identify the largest number of movies of interest in $\boldsymbol{K}$ questions.
Movies live in a movie plane:



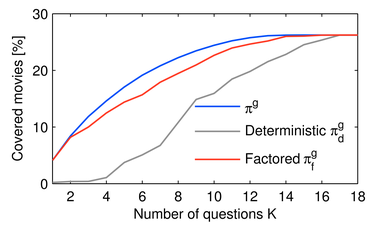
We want to cover this space as much as possible by asking adaptively K questions:



- The preferences of users are estimated from 500 most rated movies in the *MovieLens* dataset.
- The reward for asking user ϕ questions A is the percentage of movies that belong to at least one genre that is preferred by the user and queried in A.

Offline case: Our independence assumption on $P(\Phi)$ is not very restrictive in our domain.

- π^g makes no assumption on $P(\Phi)$.
- π^g_{f} assumes that the distribution $P(\Phi)$ is factored.
- π^g_{d} computes the gain as $\bar{g}_i(\mathbf{y})$, ignores the stochasticity of our problem.



Online case:

- The OASM policy outperforms the baseline π^g_{d}.
- The expected return of the OASM policy approaches that of π^g_{f}.