Adaptive Sequence Submodularity

Marko Mitrovic¹, Ehsan Kazemi¹, Moran Feldman², Andreas Krause³ and Amin Karbasi¹
¹Yale University, ²University of Haifa and ³ETH Zurich

1. Motivation

In order to properly solve many important real-world problems, our models must explicitly include notions of adaptivity and sequences.

- For example, most successful recommender systems adapt to user feedback and use it to improve future recommendations. Furthermore, in many cases, the order of the recommended items can be just as important as the items themselves.

2. Problem Set-up

In order to model sequential dependencies between items, we assume there is an underlying graph \( G = (V, E) \) where each item \( v \in V \) is represented as a vertex and the edges between the vertices encode the additional value of picking certain items in certain orders.

- For example, the directed edge from \( F \) to \( T \) in the graph below indicates that there is additional value in watching The Fellowship of the Ring before The Two Towers. However, since there is no edge from \( T \) to \( F \), there is no additional value in the reverse order.

![Graph Example](image)

3. Our Framework

To combine the graph-based model for sequences with the state-based model for adaptivity, we need to somehow assign states to the vertices and edges of our underlying graph \( G \).

- We start by defining the state of each vertex \( v \in V \) to be the state \( o \in O \) of the corresponding item. Next, each edge is assigned a state \( q \in Q \) according to some function of the states of its endpoints.

- Note that our framework works with any arbitrary function from vertex states to edge states, thus giving us great flexibility and modeling power.

- One sensible function is to define the state of each edge to be equal to the state of its start point. For example, in the graph below, vertex \( F \) is in state \( o = 1 \), which means that all edges originating from \( F \) will also be in state \( q = 1 \). Similarly, since the state of vertex \( R \) is unknown, the states of all edges originating from \( R \) are also unknown.

Another way to view this framework is that a sequence of items \( \sigma \) induces a set of edges \( E(\sigma) \) and a realization of item states \( \phi \) induces a realization \( \phi^E \) for the states of the edges.

- For example, in the graph, the sequence \( \sigma_1 = \{F, T\} \) would induce the set of edges \( E(\sigma_1) = \{(F, F), (F, T), (T, T)\} \). On the other hand, the sequence \( \sigma_2 = \{F, T\} \) would only induce the set of edges \( E(\sigma_2) = \{(F, F), (T, T)\} \).

4. Submodularity

- Explicitly modeling both adaptivity and sequences comes at a cost as the size of the search space increases exponentially.

- In order to bring tractability to this problem, we view it through the lens of submodularity.

  - Intuitively, a function is said to be submodular if it exhibits diminishing returns.

    Mathematically, a function \( h \) is said to be submodular if for all sets \( A \subseteq B \) and all elements \( v \in V \setminus B \):

    \[
    h(A \cup \{v\}) - h(A) \geq h(B \cup \{v\}) - h(B)
    \]

    This allows us to define \( f(\sigma, \phi) = h(E(\sigma), \phi^E) \), where \( h \) is a submodular function on the edges, \( \sigma \) induces \( E(\sigma) \), and \( \phi \) induces \( \phi^E \).

5. Policy and Theoretical Results

![Algorithm 1: Adaptive Sequence Greedy Policy](image)

- Our policy takes a greedy-based approach where, at every iteration, we select the valid edge \( e \in E \) with the highest expected marginal gain.

- After we select each edge, we append the corresponding vertices to our sequence \( \sigma \) and observe their states. This allows us to update \( \psi^E \), which essentially holds all the state information we have observed so far.

- We theoretically show that the expected value of our policy is within a factor of \( \frac{1}{1 - \gamma} \) of the expected value achieved by the true optimal policy, where \( \gamma \) is the weak submodularity parameter and \( \phi^E \) is the maximum in-degree.

6. Product Recommendation Application

- Using the Amazon Video Games dataset (McAuley et al., 2015), our goal is to recommend \( k \) products to each user based on their purchase history.

- Our primary observation is that our policy is competitive with sophisticated deep learning-based baselines while providing numerous other advantages such as robustness against data scarcity, robustness against changing inputs, theoretical guarantees, and interpretability.

7. Acknowledgements

This work is partially supported by NSF (IIS-1845032), ONR (N00014-19-1-2406), AFOSR (FA9550-18-1-0160), ISF (1357/16), and ERC StG SCADAPT.