Slate Recommendation Systems: A Reinforcement Learning Problem

Introduction to Problem Statement:

The PS here deals with building a Recommender System that optimally chooses and places C=9 items on a grid out of a total dataset of T=350 items. This should consider some basic features of the users, expected to be single-use, as well as interaction and correlation amongst the chosen items. We also wish to incorporate features such as seasonality and trends.

What a Multi Arm Bandit Approach would look like:

A simple bandit approach would treat every slate as one item, i.e. one action set. This would create 350P9, i.e. 7*10^22 items. Finding enough data to exploit the best actions from this set would be near impossible. This makes it intractable to use on such a dataset. This leads to the need to look at alternate methods of solving this problem

SlateQ:

A recent approach to the slate recommendation problem is an MDP model called SlateQ. SlateQ looks at building a two-part solution to this problem as follows:

1. A User Choice Model:
   a. Input:
      Interaction Data from users
   b. Output:
      For every grid, the user choice model outputs a relative probability of selecting each item. For example, this probability would be 1/10 for each item before training. (The 10th item being a null item)
      After that it would then train to calculate relative probability.
   c. How:
      SlateQ does not explain this model. Here we are looking to ideate and create an alternate model.

2. A Slate Building Model:
   a. Input:
      The relative probabilities of selection, and the rewards associated with each item
   b. Output:
      The best slate out of the 7*10^22 items.
   c. How:
      SlateQ uses mixed integer programming to solve a combinatorial problem in linear time.

Approach to the problem:

This problem can be approached at two stages:

1. Stage 1: Looking for alternate slate recommendation solutions
    a. This includes looking at SlateQ, but also exploring other alternatives such as off-policy evaluation, SEO models, CMAB models etc.
2. Stage 2: Proceeding with slateQ and looking at possible User Choice models
    a. If we do proceed with SlateQ, it still leaves uncertainty as to how the user choice will be modeled.
    b. This could include looking at Multinomial Logit models, CTR models or building a model from scratch.

Stages of solution:

1. Literature Review:
    a. This is the current step of the process that I am at which involves reading both about possible alternate models as well as possible options for user choice and understanding their pros and cons
    b. As of now, I have gone the papers listed below and am still to find any evidence that interactions data is modeled at any stage of this process.
    c. I have found papers that take into account the positioning of the items
2. Proposed Alternative Model:
    a. As of now my proposal is to consider 350C2 interactions.
    b. I'll explain this model further as we build it.

References:

1. SlateQ:
    a. Paper: https://www.ijcai.org/proceedings/2019/0360.pdf
    b. Implementation Code: https://docs.ray.io/en/latest/rllib/rllib-algorithms.html
    c. Source Code: https://docs.ray.io/en/latest/_modules/ray/rllib/algorithms/slateq/slateq.html#SlateQConfig
    d. Implementations: https://github.com/collinprather/SlateQ/blob/master/notebooks
2. RecSim platform: https://arxiv.org/pdf/1909.04847.pdf
3. Off-policy evaluation: https://proceedings.neurips.cc/paper/2017/file/5352696a9ca3397beb79f116f3a33991-Paper.pdf
4. Combinatorial Multi Arm Bandit: http://proceedings.mlr.press/v28/chen13a.pdf