INTRODUCTION

Briefly, we implement both LinUCB and Hybird LinUCB algorithms for online recommendation. When a user arrives, we recommend an article to user using LinUCB and if the recommended one is the same with the display one, we update the weights of the model. Our best result is given by LinUCB, reaching over 0.68 online. But the Hybrid model would have timeout error and the performance is worse than LinUCB.

IMPLEMENTATION

Specifically, LinUCB is simple but there are many things we should pay attention to. The first one is the hyper-parameters of LinUCB model, including alpha and rewards. The alpha is based on the delta and we set delta to 0.05, which give better result online instead of setting alpha=0.2 (better offline). As for the reward, we need to rescale them because the number of logs with reward r=1 is far less than the logs with r=0 so we set reward r1 to 20 when r=1 and r0 = -1 when r=0 to reduce the gap, which give the best result after many trials. The second one is to reduce the number of calculations. As we need to recommend for each user, but don’t need to update every time. Therefore, we put those computationally intensive terms (e.g., inverse matrix) in the update function that are needed to compute the UCB in recommend function, which is a way to optimize the running time. But besides vectorization, we have not found good ways to reduce computation in Hybrid model.