CS 6780 Research Proposal: Top N Recommender Systems

1. Motivation

2. Problem Formulation

In this problem, we are given a collection of items (such as movies, books, articles, etc), denote them as $I = \{I_1, I_2, \cdots, I_M\}$. For each item I_n, we use a k dimension vector $I_n = (i_{1,n}, \cdots, i_{k,n})$ to represent its score on k different features. For any time $t = 1, 2, \cdots, T$, we are asked to pick N items $S_t = \{I_{t,1}, I_{t,2}, \cdots, I_{t,N}\} \subset I$, order them and forward these N items to the user.

We assume there is a known probability vector $(p_1, \cdots, p_N)$ associated with the N different positions. Since the top positions are more likely to be seen by the user, we further assume $p_1 \geq p_2, \cdots \geq p_N$. The user has a user preference vector $\theta = \{\theta_1, \cdots, \theta_k\}$ which is unknown but remain static over the time. Every time the user sees N items, he/she will click the item $I_{t,i}$ with probability $p_i \theta \cdot I_{t,i}$. If the user clicks the item, then we receive a reward 1 and 0 otherwise. Thus, the feedback from the user in this case is a N dimensional vector $(Y_{1,n}, \cdots, Y_{N,n})$ and $Y_{i,n} = 1$ if the users clicks the item on the i_{th} position and 0 otherwise. For each $Y_{i,n}$, we know

$$Y_{i,n} | \theta, I_{t,i}, p_i \sim Bernoulli(p_i \theta \cdot I_{t,i}). \tag{1}$$

Denote the total reward received at time n as $Y_n = \sum_{i=1}^{N} Y_{i,n}$.The goal of this problem is to find a strategy π to maximize the following expression

$$E^{\pi}\left[\sum_{t=1}^{T} Y_t\right] \tag{2}$$

The questions we are going to solve is the following:

(1) Suppose there are many historical users in our database and we know their user preference vectors. Then we can calculate the empirical distribution for those user preference vectors and treat it as the prior distribution of the current θ. Every time we observe the user's feedback $(Y_{1,n}, \cdots, Y_{N,n})$, we can use this feedback and maximum likelihood method to find a new estimate for θ, denote as $\hat{\theta}_n$. Then we make our next selection based on this $\hat{\theta}_n$. How good is this greedy method compared to other existing methods?

(2) Further, when we make the selection based on $\hat{\theta}_n$, what's the best trade off between exploitation and exploration?

(3) When k is large, this problem is computational intractable. How could we solve this problem efficiently? Can we prove any structural results for this problem?

3. Possible Applications