Group Recommendations Via Multi-armed Bandits*

José Bento* Stratis Ioannidis† S. Muthukrishnan‡ Jinyun Yan‡
*Stanford University †Technicolor ‡Rutgers University
*jbento@stanford.edu, †stratis.ioannidis@technicolor.com, ‡{muthu,jinyuny}@cs.rutgers.edu

ABSTRACT

We study recommendations for persistent groups that repeatedly engage in a joint activity. We approach this as a multi-arm bandit problem. We design a recommendation policy and show it has logarithmic regret. Our analysis also shows that regret depends linearly on d, the size of the underlying persistent group. We evaluate our policy on movie recommendations over the *MovieLens* and *MoviePilot* datasets.

Categories and Subject Descriptors

H3.3 [**Information Search**]: Recommendation—*Information Storage Retrieval*

General Terms

Algorithm, Design, Experimentation

Keywords

Group Recommendation, Multi-armed Bandits

1 Introduction

Humans are social beings and no person is an island unto themselves. People engage in activities as part of one or more groups. At home, families watch TV or plan movie nights, restaurant evenings, day trips, jointly as a group. At work, colleagues plan lunches, weekly socials, off-site retreats, and so on, also as a group. In fact, seldom is a person alone in many of their daily activities.

While the problem of providing recommendations to individuals is popular both in research and in applications, providing recommendations to groups of individuals has been much less explored. One important aspect one needs to understand is how group membership manifests over time. Our focus is on *persistent* groups, that is, those whose members are bound together due to some purpose and meet regularly. Examples include families, bands of friends, teams at work and so on. Neverthless, not all members of even a persistent group will be present for each of the group activities. People have conflicting appointments, medical time off, *etc.*

Our contributions are as follows. First, we formalize the problem of recommendation to persistent groups as a suitable Multi-armed bandit (MAB) problem and extend the theory of *minimum regret* MAB policies. In particular, we design a MAB policy for this problem, *Group-UCB*, and prove a logarithmic upper bound on the regret under suitable conditions. Second, we adapt Group-UCB to a concrete setting of recommending movies. We study a real data set from MovieLens which provides users' movie ratings and movie genres.

2 Group Multi-armed Bandit

2.1 Problem Formulation

We formulate group recommendation as a MAB problem as follows. Let $t = 1, 2, ...$ denote the series of times when recommendations are made. Let G be the persistent group of $d = |G|$ users. At any time t, $S(t) \subset G$ is the set of users that are present. Further, each user u has a weight $x_u(t)$ at time t. The vector $x(t)$ takes values in a finite set $\mathcal{X} \subset \mathbb{R}_+^d$ and models users who are not present (set $x_u(t) = 0$) as well as the relative influence of a user on the overall group's satisfaction. Our focus is on designing an algorithm to recommend one of the objects at each time t. Each user $u \in S(t)$ will provide an individual rating $r_u(t) \in [0, 1]$ for the recommended object; $r_u(t)$ is a random variable. We seek a recommendation algorithm that maximizes the expectation of $\sum_u x_u(t)r_u(t)$, over t.

Denote by A, with $|A| = K$, the set of all possible genres a movie can belong to, for instance, "horror","comedy", *etc*. We assume that, conditioned on a movie's genre being $a \in A$, the expected value of the rating given user $u \in G$ is $\theta_{u,a} \in [0, 1]$. These conditional expectations $\theta_{u,a}$ consitute a vector $\theta_a = [\theta_{u,a}]_{u\in G} \in [0, 1]^d$ determining every user's expected reaction towards genre a. Under this notation, if a movie from genre $a \in A$ is suggested and viewed by the users, the expected group rating is given by the inner product $\langle x(t), \theta_a\rangle = \sum_{u\in G} x_u(t)\theta_{u,a}$.

2.2 Group-UCB

Algorithm 1 Group-UCB

$n_u \leftarrow 0$; $n_{u,a} \leftarrow 0$; $\bar\theta_{u,a} \leftarrow 0$
for $t = 1$ to T **do**
  Observe present users $S(t)$ and weight vector $x(t)$
  for a=1 to K **do**
    for u =1 to d **do**
      if $n_{u,a} = 0$ then $p_{u,a} \leftarrow \infty$
      else $p_{u,a} \leftarrow \bar\theta_{u,a} + \sqrt{2 \ln n_u/n_{u,a}}$
    end for
    $p_a \leftarrow \sum_u x_u(t) \cdot p_{u,a}$
  end for
  choose arm $a \leftarrow \arg\max_a p_a$ (break ties arbitrarily)
  observe rating r_u from each user $u \in S(t)$
  $n_{u,a} \leftarrow n_{u,a} + 1$ for all $u \in S(t)$
  $n_u \leftarrow n_u + 1$ for all $u \in S(t)$
  $\bar\theta_{u,a} \leftarrow \bar\theta_{u,a} \cdot (n_{u,a} - 1)/n_{u,a} + r_u \cdot 1/n_{u,a}$
end for

A detailed description of the policy we propose can be found in Algorithm 1. In short, the recommender maintains estimates of the quantities $\theta_{u,a}$, for all $u \in G$ and $a \in A$. If u has rated movies from genre a for s times so far, the estimate of $\theta_{u,a}$ is the empirical average $\bar\theta_{u,a}(s) = \sum_{\tau=1}^{s} r_u(\tau)/s$ where r_u is the rating user u gave to τ-th movie from genre a. Moreover, the recommender keeps track of how many times a user has participated in the activity and a particular genre has been displayed: it keeps track of $n_{u,a}(T)$, the number of times that u has been present *and* a movie from genre a has been suggested upto session T, as well as $n_u(T)$, the number of times that u has been present upto session T. Using the above quantities, the recommender selects a genre as follows. At the t-th session, the recommender first observes the present composition of group $S(t)$ and the present weight vector $x(t)$. The genre a selected is the one maximizing $\sum_{u\in G} x_u(t)\big(\bar\theta_{u,a} + \sqrt{2 \ln n_u/n_{u,a}}\big)$. Subsequently, the recommender suggests a movie from that genre to the users in $S(t)$; the latter react by providing the recommender with ratings, which are

*This material is based upon work supported by the National Science Foundation under Grant No. 0916782

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

then used to update the estimates $\bar{\theta}_{u,a}$ for the arm $a = a(t)$ and for users $u \in S(t)$.

2.3 An Upper Bound

Given a vector $x \in \mathcal{X}$, an *optimal genre* $a^*(x)$ is a genre that maximizes the expected group rating, *i.e.*, $a^*(x) = \arg\max_{a \in A} \langle x, \theta_a \rangle$. Given a policy $\{a(t)\}_{t=1}^{T}$, we define the *regret* of the recommender after T sessions to be $R(T) = \sum_{t=1}^{T} \langle x(t), \theta_{a^*(x(t))} \rangle - \sum_{t=1}^{T} \langle x(t), \theta_{a(t)} \rangle$, where $a^*(x(t))$ is a genre that is optimal at time t. The regret under Group-UCB can be bounded according to the following theorem.

THEOREM 1. *Given $x \in \mathcal{X}$, denote by $B_x \subset A$ the set of suboptimal genres under x,* i.e., $B_x = \{a \in A : \langle x, \theta_{a^*(x)} \rangle > \langle x, \theta_a \rangle\}$ *Moreover, let $\Delta_{\min}^a = \inf_{x \in \mathcal{X}: a \in B_x} \langle x, \theta_{a^*(x)} \rangle - \langle x, \theta_a \rangle$. Then, under Group-UCB,*

$$R(T) \le \sum_{a \in A} \frac{8M_1^3 d}{(\Delta_{\min}^a)^2} \ln T + 4KdM_1.$$

The bound in Theorem 1 holds *for arbitrary sequences $x(t) \in \mathcal{X}$*: irrespectively of which subsets of users show up, and their ratings are weighted, the regret is logarithmic. Compared to the bound of the regret of UCB for the classic bandit problem appearing in [1], Theorem 1 differs by a multiplicative factor of $dM_1^3/\Delta_{\min}^a$. It can be shown that the leading coefficient must be $\Omega(d)$ and, in this sense, the bound is tight in d. The constant M_1 is a bound on the sum of weights of all users. For practical purposes, this ought to be small. Moreover, the constants $\Delta_{\min}^a$ capture the gap in expected group rewards between optimal and suboptimal arms. Similar quantities also appear in the bound of [1].

3 Experimental Evaluation

We evaluated Group-UCB on two datasets, *MovieLens* [2] and *MoviePilot* [3]. For brevity, we only present results for *MovieLens*.

MovieLens Dataset. The *MovieLens* dataset consists of 1 000 209 ratings, given by 6 040 users to 3 883 movies. Ratings of a movie range from 1 to 5. Movies in the dataset are labeled by genres, such as "Animation", "Children's", *etc*. Before simulating Group-UCB, we construct a low-rank approximation of the MovieLens dataset; we subsequently use it to predict movie ratings for arbitrary movie-user pairs—*i.e.,* perform matrix completion.

Simulation Setup and Evaluation. We simulate Group-UCB on two types of groups in MovieLens dataset: (a) random groups with size up to 10, (b) a special group of users with shared profile. At each iteration t, the subgroup $S(t)$ is selected uniformly at random from non-empty subset of G. At each session, if Group-UCB selects genre a, a movie selected u.a.r. among movies tagged with label a is displayed to the present group. The reaction of a user is then the rating that she provided, if the latter is in the dataset; otherwise, the rating predicted by our low-rank approximation model is used.

To form the group, we randomly pick 10 users from the dataset, then build 10 groups by adding one more user to the group each time. Figure 1(a) plots the regret in semi-log scale. It shows that regrets indeed grow logarithmically. We estimate its regret slope as the slope of the final portion of the curve, which is a straight line. Sampling 6 times, we compute average slopes with regards to different group sizes. Figure 1(b) illustrates the average slope has positive relationship with the group size. We also group users by location: picking users in a same zip code. Profiles of members in the selected group are shown in Table 1. In Figure 1(a), each line represents the regret given a group with random users. We can see that lines can be distinguishable for group size 2 to 6. However,

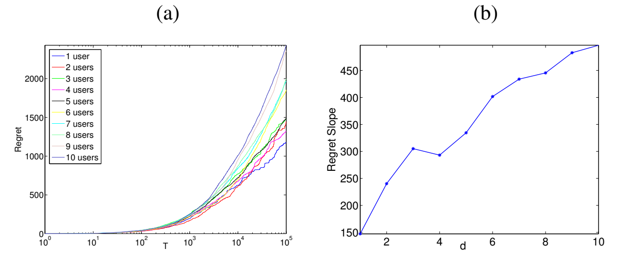

Figure 1: (a) Random Groups up to 10 Users. (b) Average slope of the regret VS Group Size.

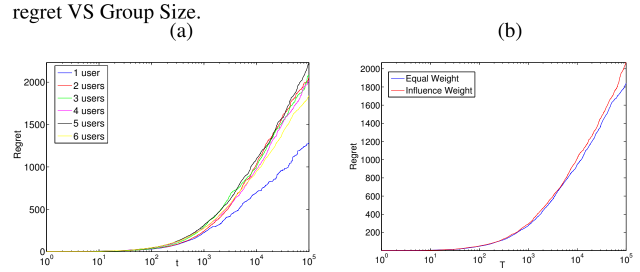

Figure 2: (a) Regret for Groups that people share zip code. (b)Regret for equal weight vector and influence weight vector.

in Figure 2(a), which depicts the regret lines for a group of users sharing a zip code, lines are indistinguishable from size 2 to 6.

User ID	Gender	Age	Occupation
1	Male	18-24	college/grad student
2	Male	18-24	programmer
3	Male	18-24	college/grad student
4	Female	25-34	sales/marketing
5	Female	18-24	other
6	Male	35-44	academic/educator

Table 1: member information in the group

Next, we now investigate the scenario in which users have different influence in the group reward. As it is not easy to determine the influence power of each member, we use a simple heuristic: older people and females gain higher weight. Take our 6-user group in Table 1 as an example. Initially, every user has $w_u = 1$. User 4 gets 1 more weight unit and User 6 gains 2 more units because of their age. User 4 and User 5 will get 1 more unit because of their gender. The final weight vector w for the group is (1,1,1,3,2,3). This reflects how each individual affects the group reward.We then apply Group-UCB on this group, with other settings not changed. Figure 2(b) displays the regrets for *Equal Weight* and our heuristic *Influence Weight*, on semi-log scale. We can see that Group-UCB performs well for both types of weight vectors. The final leading constant of regret is 348 for *Equal Weight* and 380 for *Influence Weight*. Hence the regret does not vary significantly, indicating that our Group-UCB policy works for varying values of $x(t)$.

4 Conclusion

Our work has initiated the MAB approach to group recommendations. Many extensions remain open. For example, can we work with only a group rating each time, rather than rating from each individual? How can we extend our policy when genres are correlated? Last but not least, Group-UCB may have other applications.

5 References

[1] P. Auer, N. Cesa-Bianchi, P. Fischer, and L. Informatik. Finite-time analysis of the multi-armed bandit problem. In *Machine Learning*, 2002.

[2] GroupLens. http://www.grouplens.org/node/73.

[3] MoviePilot. http://moviepilot.com/.