A Fast Bandit Algorithm for Recommendations to Users with Heterogeneous Tastes

Pushmeet Kohli and **Mahyar Salek**
Microsoft Research
Cambridge, United Kingdom
{pkohli, mahyar}@microsoft.com

Greg Stoddard
Northwestern University
Evanston, Illinois, USA
gregs@u.northwestern.edu

Abstract

We study recommendation in scenarios where there's no prior information about the quality of content in the system. We present an online algorithm that continually optimizes recommendation relevance based on behavior of past users. Our method trades weaker theoretical guarantees in asymptotic performance than the state-of-the-art for stronger theoretical guarantees in the online setting. We test our algorithm on real-world data collected from previous recommender systems and show that our algorithm learns faster than existing methods and performs equally well in the long-run.

1 Introduction

The market for online content consumption and the rate at which content is produced has experienced immense growth over the past few years. New content is generated on a daily or even hourly basis, creating an incredibly fast turn-over time for relevant content. While traditional search and recommendation engines have the ability to discover quality content in an offline manner, services such as news aggregators need to constantly adjust their recommendations to cater to current hot topics. For example, articles about the U.S. presidential inauguration may be quite popular on January 21$^{\text{st}}$, the day of the inauguration, but they're likely to fall out of favor on the morning of the 22$^{\text{nd}}$. In the face of such rapid changes in relevance, online algorithms which continually optimize recommendations based on user usage data provide an attractive solution.

We propose a simple online recommendation algorithm which learns quickly from user click data to minimize abandonment, the event that a user does not click on any articles in the recommended set (also known as *%no* in the information retrieval community). Our algorithm operates with minimal assumptions and no knowledge of features of users or articles, and thus is well-suited to address changing environments induced by frequent turn-over in the set of potential articles and shifts in user preferences. We focus on content such as news articles, jokes, or movies, where users have varying *tastes* but there's no notion of a single "correct" recommendation.

Recommending relevant content is a key challenge for search engines and recommendation systems and has been extensively studied in the information retrieval community. The early guiding principle in the IR literature was the *probability ranking principle* (PRP) (Robertson 1977), stating that articles should be ranked in decreasing order of relevance probability. (Chen and Karger 2006) noted that optimizing with PRP in mind may yield sub-optimal outcomes, particularly when the objective is minimizing abandonment. In recent years, the concept of "diversity" in recommended sets of content has emerged as a guiding principle which better serves in addressing goals such as abandonment minimization. The intuitive goal behind a diverse set of content is to use each article in the set to satisfy a different type of user. This approach is particularly applicable to the canonical problem of handling a variety of user *intents*; when a user searches for a term such as "jaguar" their intended meaning could be the car, the animal, the American football team, or a number of different meanings.

This paper compares the PRP and diversity principle from an online algorithm perspective. We compare our online algorithm, which is implicitly based on the PRP, with the Ranked Bandit Algorithm (RBA) of (Radlinski, Kleinberg, and Joachims 2008), which is based on the diversity principle. While the diversity principle yields superior offline performance, our approach has stronger theoretical guarantees in the online case. Our empirical work focuses on a fundamentally different sort of user preference than the previous diversity work. Instead of intent, we cater to a heterogeneity of users *tastes*, i.e. does the user find this joke funny or will the user like this news article. Surprisingly, we find that explicitly incorporating diversity in this setting doesn't yield a large gain; the offline PRP-based solution gives nearly the same performance as the offline diversity-based solution.

At the heart of our method is the use of a stochastic multi-armed bandit algorithm to control the trade-off between exploration and exploitation of articles. A multi-armed bandit problem is an abstract game where a player is in a room with many different slot machines (slot machines are sometimes called one-armed bandits), with no prior knowledge of the payoffs of any machines. His goal is to maximize his total reward from the slot machines and in doing so, he must explore machines to test which machine has the highest average payoff but also exploit those he knows to have high rewards. Similar to (Radlinski, Kleinberg, and Joachims 2008), the primary contribution of our algorithm is the method by which we combine instances of several MAB algorithms to efficiently approximate this combinatorial set recommendation problem.

1.1 Our Contributions

We present an online algorithm for the minimization of abandonment. Our method uses several instances of a multi-armed bandit algorithm working (almost) independently to recommend a set of articles. Although the independence between bandit instances carries all the drawbacks of the PRP, we use a stochastic optimization concept known as the *correlation gap* (Agrawal et al. 2010) to prove that our algorithm has near-optimal performance in the online setting. Furthermore, the independence between bandit instances allows for a faster learning rate than online algorithms based on the diversity principle. Our second contribution is an empirical study of bandit-based recommendation algorithms on real-world datasets collected from previous recommendation algorithm research. We find that while in theory the diversity-based solutions yield superior offline solutions, in practice there are only small differences between the offline diversity-based solution and the offline PRP-based solution. We also empirically verify that the learning rate of our method is faster than that of existing methods.

2 Previous Work

Previous work in information retrieval and machine learning has addressed recommendation to heterogenous populations via the goal of maximizing diversity in search results but the literature varies widely in modeling assumptions. In some work, diversity refers to increasing the set of topics that a recommended set of articles or search results may cover (Agrawal et al. 2009) and (Panigrahi et al. 2012). Other works assume users intrinsically value diversity; (Raman, Shivaswamy, and Joachims 2012) and (Yue and Guestrin 2011) both assume a rich feature model and use online learning techniques to learn user utility functions. (Li et al. 2010) give an online approach for news recommendation using a user's profile as a feature vector. (Chen and Karger 2006) prove in a general sense that the standard submodular greedy algorithm is the optimal way to incorporate diversity into search result rankings.

By contrast, our work carries little assumptions. In this sense, our work is closer to the literature on online stochastic submodular maximization, particularly in the bandit setting. (Calinescu et al. 2011) prove that a continuous version of the standard submodular greedy algorithm yields an optimal approximations for all matroid constraints and (Streeter, Golovin, and Krause 2009) give a similar method (though less general) which can be extended to the online setting.

The work most closely related to ours is (Radlinski, Kleinberg, and Joachims 2008) and (Slivkins, Radlinski, and Gollapudi 2010), although the latter work makes strong use of a similarity measure between documents whereas we assume no such construct. Their "ranked bandit algorithm" serves as our baseline in this paper and we discuss the relationships between our methods in later sections.

3 Problem Formalization

We consider the problem of minimizing abandonment for an article recommendation system. At the beginning of the day, n articles are submitted to the system. When a user visits our site, they're presented with a set of k articles; if the user finds any of the articles relevant, he clicks on it and we receive a payoff of 1. If the user finds no articles relevant, we receive a payoff of 0. We receive no additional payoffs if the user clicks on more than one article. Each user j can be represented by a $\{0,1\}^n$-vector X^j, where a $X_i^j = 1$ indicates that the user j finds article i relevant. These relevance vectors X^j are distributed according to some unknown distribution D. These relevance vectors can be thought to represent the *type* of a user. This type structure allows for a large degree of correlation between article relevances.

At each time period t, a random user arrives, corresponding to choosing a vector X^t i.i.d. from D, and we present a set of k articles S^t. Let $F(S^t, X^t)$ denote the payoff for showing set S^t to a user with relevance vector X^t. We'll refer F as the *set relevance* function and it has the following form

$$F(S^t, X^t) = \begin{cases} 1 & \text{if } X_i^t = 1 \text{ for some } i \in S^t \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The user's relevance vector X^t is not observed before the algorithm the set is chosen. Thus the value of displaying a set S^t is the expected value $E[F(S^t, X)]$ where the expectation is taken over the realization of the relevance vector X from the distribution D. When it is clear, we will write $E[F(S)]$ as shorthand for $E[F(S, X)]$. In words, $E[F(S)]$ is the fraction of users who will be satisfied by at least one article in S. The problem of minimizing abandonment is equivalent to the problem of maximizing $E[F(S)]$ subject to $|S| \leq k$. For the remainder of this paper, we'll focus on maximizing expected set relevance $E[F(S)]$.

Before turning to the online version of this problem, we consider optimization in the offline setting. In the offline setting, an algorithm would have access to the distribution but even with such assumptions the problem is NP-hard[1] Despite this intractability, we can take advantage of the structure of $F(S)$, namely that it is *submodular*, and use the greedy algorithm of (Nemhauser, Wolsey, and Fisher 1978). This yields a $(1 - \frac{1}{e})$ approximation, which is the best possible approximation under complexity theoretic assumptions. (Chen and Karger 2006) argue this greedy approach yields an optimally diverse set of articles.

A set function G is said to be *submodular* if, for all elements a and sets S, T such that $S \subseteq T$, $G(S \cup a) - G(S) \geq G(T \cup a) - G(T)$. The set relevance function $F(S, X)$, as defined in equation 1, is submodular and this property forms the theoretical basis for the online approaches given in the next section.

[1] This can be shown by a standard reduction from the max coverage problem. See (Radlinski, Kleinberg, and Joachims 2008) for details.

4 The Online Problem

We now turn to the online version of this problem, which presents a classic explore-exploit tradeoff: we must balance the need to learn the average relevance of articles with no feedback against the need to exploit the good articles that we've already discovered. We solve this problem using theoretical results from the multi-armed bandit (MAB) literature, a class of algorithms which solve exploration-exploitations problems. Bandit problems can be distinguished by the assumptions made on the rewards. In the *stochastic* bandit problem, rewards for each option are drawn from a stationary distribution while in the *adversarial* setting, payoffs for each option are determined by an adversary who has knowledge of past play, history of rewards, and the strategy that the player is using.

The objective of an online algorithm is the minimization of *regret*, where the regret of an algorithm is defined as the expected difference between the accumulated rewards of the single best option and the rewards accumulated by that algorithm. In our context, this is the difference between the fraction of users satisfied by the optimal set of articles and the fraction of users satisfied by the recommendation algorithm. However, as we noted in the previous section, maximization of $E[F(S)]$ is intractable, so we follow the approach of (Streeter, Golovin, and Krause 2009) and (Radlinski, Kleinberg, and Joachims 2008) and use $(1-\frac{1}{e})OPT$ as the offline benchmark. The regret after time t is defined as

$$R(T) = (1-\frac{1}{e})\sum_{t=0}^{T} E[F(S^*)] - \sum_{t=0}^{T} E[F(S^t)]$$

There are known bandit algorithms which achieve provably-minimal regret (up to constant factors), but direct application of these bandit algorithms requires exploring all possible options at least once. In our setting, each subset of articles is a potential option and hence there are exponentially many options, making standard MAB algorithms impractical. In the next section we present two approaches, one from previous work and our algorithm, for combining several instances of a bandit algorithm to yield a low-regret and computationally efficient solution to this recommendation problem.

4.1 Ranked Bandit Approach

The work of (Radlinski, Kleinberg, and Joachims 2008) and (Streeter, Golovin, and Krause 2009) introduced the "ranked bandit" algorithm to solve the problem of minimizing abandonment. The pseudocode is given in algorithm 1. The idea behind the ranked bandit algorithm is to use k instances of a MAB algorithm to learn the greedy-optimal solution (which is also the diversity-optimal solution). Specifically, k instances of a bandit algorithm are created, where bandit i is responsible for selecting the article to be displayed in slot i. The algorithm is designed such that the bandit in slot i attempts to maximize the marginal gain of the article in slot i. In the context of minimizing abandonment, bandit i attempts to maximize the click-through-rate of the article in slot i given that the user has not clicked on any earlier articles.

Algorithm 1 Ranked Bandit

```
MAB_i : Bandit algorithm for slot $i$
for $t = 1...T$ do
    $s_i \leftarrow$ selectArticle(MAB_i, N)
    $S^t \leftarrow \cup_i s_i$
    Display $S^t$ to user, receive feedback vector $X^t$
    Feedback:
        $z_i = \begin{cases} 1 & \text{if article } s_i \text{ was the first click} \\ 0 & \text{otherwise} \end{cases}$
    update(MAB_i, $z_i$)
end for
```

While RBA works with any bandit algorithm, the regret of RBA depends on the choice of bandit algorithm. (Radlinski, Kleinberg, and Joachims 2008) use an adversarial bandit algorithm known as EXP3 in their work and show that RBA inherits the regret bounds guaranteed by EXP3. However the adversarial assumption is overly pessimistic in this problem and ideally we could make use of the stochastic nature of user behavior. Stochastic bandit algorithms such as UCB1 have better theoretical and practical performance but the dependence between slots in RBA violates the necessary independence assumptions for the stochastic setting. In their work, (Radlinski, Kleinberg, and Joachims 2008) show RBA to have regret on the order of $O(k\sqrt{Tn\lg(n)})$. Our approach, discussed in the next section, is able to leverage the stochastic nature of the problem without complication and thus achieves a provable regret of $O(kn\lg(T))$.

In addition to the lack of theoretical guarantees, the learning rate of RBA can be quite slow because of "wrong" feedback. The "correct" value of an article in slot $i+1$ is the marginal value of that article given that slots 1 to i are displaying the "correct" articles, that is the first i articles in the greedy solution. In any time period where those articles aren't displayed, the marginal value of any article in slot $i+1$ will not necessarily be correct. Although early slots should display the correct articles most of the time, later slots can't begin learning correctly until the earlier slots converge. This effectively induces sequential learning across slots and back of the envelope calculations suggest that "correct" learning will only begin in slot $k+1$ after $\Omega(n^k)$, time steps have past.

4.2 Independent Bandit Approach

In this section we describe our method which we call the *independent bandit* algorithm (IBA) which is implicitly based on the probability ranking principle. Rather than learning the marginal values as in the ranked bandit algorithm, the independent bandit algorithm optimizes the click-through-rate of each slot independently of the other slots. Using tools from stochastic optimization theory, we prove that the independent bandit algorithm has near-optimal regret and our simulations demonstrate that IBA converges to its offline-optimal solution much quicker than RBA.

The pseudocode for the independent bandit algorithm is given in algorithm 2. Line 5 ensures that the bandit algorithms don't select the same articles by temporarily removing articles already displayed from the set of potential articles for bandits in later slots. The main difference between the independent and the ranked bandit algorithm is the feedback; IBA gives a reward of 1 to any article that was clicked on while RBA only gives a reward of 1 to the first article that was clicked on. This independence between bandit instances in IBA allows for learning to happen in parallel, enabling a faster rate of learning for IBA.

To analyze the regret of IBA, we must first derive an approximation guarantee for what the offline version of the independent algorithm would compute. The independent-optimal solution consists of the k articles with the highest click-through-rates. If article relevances were all independent then the independent-optimal solution is the optimal solution, however the independent-optimal solution will be sub-optimal when there are correlations between article relevances. We use the *correlation gap* result of (Agrawal et al. 2010) to show that the independent-optimal solution yields a $(1 - \frac{1}{e})$ approximation to the optimal solution for any distribution over user relevance vectors. The correlation gap is a concept in stochastic optimization which quantifies the loss incurred by optimizing under the assumption that all random variables are independent. Formally let $G(S, X)$ be some function where S is the decision variable and X is a vector of $\{0, 1\}$ random variables, where X is drawn from some arbitrary distribution D. Let $D^{\mathcal{I}}$ be the product distribution if each X_i were an independent bernoulli variable with probability equal to its marginal probability under D. When G is a nondecreasing, submodular function the correlation gap is quite small.

Theorem (Agrawal et al. 2010) 1. *Let G be a nondecreasing, submodular function. Let S^* and $S^*_{\mathcal{I}}$ be the optimizers for $E_D[G(S, X)]$ and $E_{D^{\mathcal{I}}}[G(S, X)]$ respectively. Then* $E_D[G(S^*_I, X)] \geq (1 - \frac{1}{e})E_D[G(S^*, X)]$.

Now we consider the independent bandit algorithm. The key property of IBA is that individual bandit instances do not affect each other and this allows us to prove that IBA inherits the low regret of the underlying stochastic bandit algorithms, yielding better regret bounds than RBA. For the purposes of the next theorem, we use the UCB1 algorithm (details are given in section 5), which has regret $O(n \lg(T))$.

Theorem 1. *When UCB1 is used as the bandit algorithm for IBA, the accumulated rewards satisfy*

$$E[\sum_t^T F(S^t, X)] \geq (1 - \frac{1}{e})OPT - O(kn \lg(T))$$

Proof. The high level is to first show that IBA has low regret when compared with the independent-optimal set. We then apply the correlation gap of (Agrawal et al. 2010) to conclude the regret is close to $(1 - \frac{1}{e})OPT$.

Algorithm 2 Independent Bandit

```
MAB$_i$ : Bandit algorithm for slot $i$
for $t = 1...T$ do
    $S_0^t = \emptyset$
    for $i = 1...k$ do
        $S_i^t \leftarrow$ selectArticle(MAB$_i$, $N \setminus S_{i-1}^t$)
    end for
    Display $S^t$ to user, receive feedback vector $X^t$
    Feedback:
        $z_i = \begin{cases} 1 & \text{if article } s_i \text{ was clicked on} \\ 0 & \text{otherwise} \end{cases}$
    update(MAB$_i$, $z_i$)
end for
```

For a given document displayed in slot i, let p_i denote the marginal probability of relevance, that is $p_i = E_X[X_i]$. Assume for now that all X_i are independent. Using this independence assumption, for a given set S^t we can write the expected valued of $F(S^t, X)$ as follows

$$E[F(S^t)] = \sum_{i=1}^{k} \prod_{j=1}^{i-1} (1 - p_j)p_i \tag{2}$$

(note, this equation gives the same value for any permutation of S^t). Let $S^*_{\mathcal{I}}$ denote the set which maximizes the above function under the assumption that all X_i are independent. Trivially, this set consists of the k articles with the largest p_i. Label these elements p^*_i for $i = 1...k$. At a given time t let S^t denote the set played and let S^t_i represent the i$^{\text{th}}$ element of this set. Define $\delta_i = p^*_i - p_i$, that is the difference between the relevance probability of the best article and the relevance probability of the article actually played at time t.

$$\begin{aligned} E[F(S^t)] &= \sum_{i=1}^{k} \prod_{j=1}^{i-1} (1 - (p^*_j - \delta_j))(p^*_i - \delta_i) \\ &\geq \sum_{i=1}^{k} \prod_{j=1}^{i-1} (1 - p^*_j)(p^*_i) - \delta_i \\ &= E[F(S^*_{\mathcal{I}})] - \sum_i \delta_i \end{aligned}$$

Now taking the sum of the $f(S^t, X)$ over time yields

$$E[\sum_t F(S^t, X)] \geq \sum_t F(S^*_{\mathcal{I}}) - \sum_i \sum_t \delta^t_i$$

The term $\sum_t \delta^t_i$ is the regret incurred in slot i. (Auer, Cesa-Bianchi, and Fischer 2002) proves that the regret of UCB1 is bounded by $O(n \lg(T))$, so $\sum_t \delta^t_i \approx O(n \lg(T))$ for each slot.

In the above analysis, we assume that the probability of an article being relevant was independent of each other X_i, which is usually a faulty assumption. However, the work of (Agrawal et al. 2010) shows that optimizing under the independence assumption yields a provable approximation. Let S^* denote the set which maximizes $E[f(S, X)]$. Then the correlation gap implies $E[f(S^*_{\mathcal{I}})] \geq (1 - \frac{1}{e})E[f(S^*)]$. Combining this with the above regret bound yields the result

$$E[\sum_t F(S^t, X)] \geq (1 - \frac{1}{e})OPT - O(kn \lg(T))$$

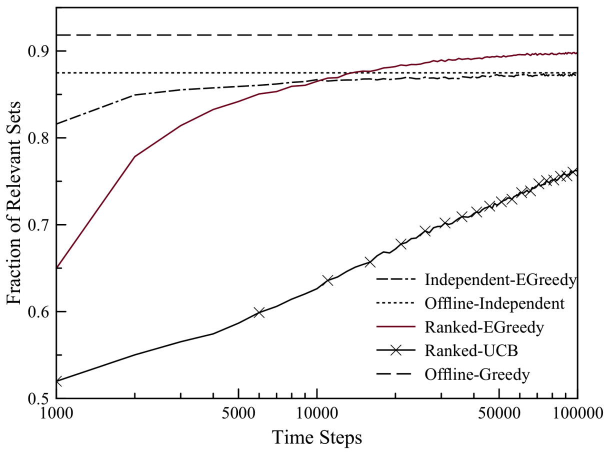

Figure 1: *Movie-Lens-100* dataset with relevance threshold $\theta = 2$, the low threshold. The Ranked-ϵGreedy method starts performing better after $t = 10000$ but fails to achieve the theoretical optimum performance within 100000 time steps. The Independent-ϵGreedy algorithm achieves its offline optimum after 50000 time steps.

□

It is worth noting that the independent-optimal solution is (weakly) worse than the greedy-optimal solution, so RBA will asymptotically outperform IBA. However, the previous theorem shows that IBA has the same worst-case guarantee along with a better regret bound that holds uniformly throughout time. In the next section, we simulate both algorithms using real-world datasets and show that the asymptotic performances of the two methods are essentially equal but IBA performs better in the short term.

5 Experimental Results

In this section, we give the results of experiments we used to test the empirical difference in performance between the ranked bandit algorithm and the independent bandit algorithm.

Datasets. We used two publicly available datasets as our input for user preferences. Our first dataset is from the Jester project (Goldberg et al. 2001) and is a collection of user ratings on jokes, ranging from -10.0 (very not funny) to 10 (very funny). Our second dataset comes from the MovieLens project (movieslens.umn.edu) and consists of user ratings assigned to movies, where each rating is from 1 (bad) to 5 (very good). Each dataset consists of a collection of $< userID, articleID, rating >$-tuples denoting the rating that the user gave to this article (either a joke or a movie). With the Jester dataset, we used two separate datasets. *Jester-Small* consist of 25000 users' ratings on 10 articles where each user had rated most articles in the set. *Jester-large* consists of 25000 users' ratings on 100 articles but there many unrated articles for each user. In the case where a user didn't rate an article, we assign that article the lowest score. *Movie-Lens-100* consists of ratings by 943 users on a sub-sampled set of 100 articles from the MovieLens dataset. For all datasets, we convert real-valued ratings to binary relevant-or-not scores by using a threshold rule; if the rating assigned by a user to an article exceeds a threshold θ, then that article is deemed relevant to that user. For each dataset, we tested a high and a low threshold for relevance.[2]

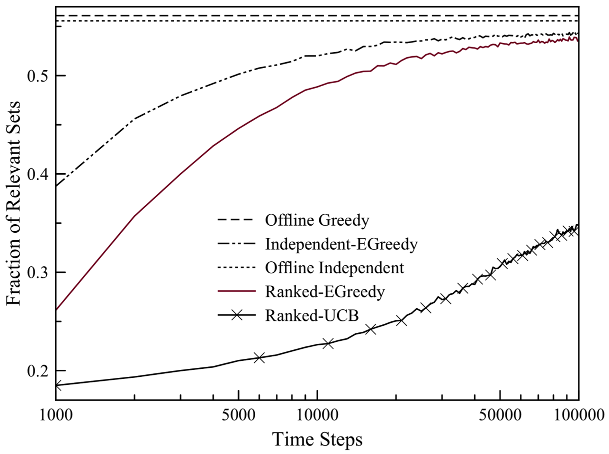

Figure 2: *Movie-Lens-100* dataset with relevance threshold $\theta = 4$, the high threshold. The Independent-ϵGreedy method performs the best out of all four methods.

The data we use is of a fundamentally different nature than the generated by (Radlinski, Kleinberg, and Joachims 2008). In that work, they model user *intent*, i.e. is a user that searches for the term "jaguar" talking about the car, the animal, or some other meaning? In our work, we care about user *taste*, i.e. which joke or movie will a user like? In the case of intent, there's generally a correct answer and a single article rarely satisfies multiple types of users. For the case of taste, there is rarely a single "correct" answer and a single article may satisfy many different types of users.

Baselines.In our experiments, we used two well-known stochastic multi-armed bandit algorithms to test the Ranked Bandit Algorithm and the Independent Bandit Algorithm. Both algorithms, UCB1 and ϵ-Greedy, are examined in detail in (Auer, Cesa-Bianchi, and Fischer 2002) but we briefly review them here. In each time step t, UCB1 plays the option which maximizes $\bar{x}_i + \sqrt{\frac{2\lg(t)}{t_i}}$ where $\bar{x}_i$ denotes the current average reward of option i and t_i denotes the number of times that option i has been played so far. The second term in this equation naturally induces exploration since this term grows for options that have not been played in a while.

The second MAB algorithm is the ϵ-Greedy algorithm. At each time t, with probability ϵ a uniformly random arm is played, and with probability $1-\epsilon$ the option with the current highest average reward is played. Note that this algorithm requires the ϵ parameter to be tuned; for these experiments, we set $\epsilon = .05$, which proved to give the best average performance during initial tests.

[2] We only show the results for a few different datasets due to space constraints. These datasets are representative of the qualitative results from the entire set of experiments.

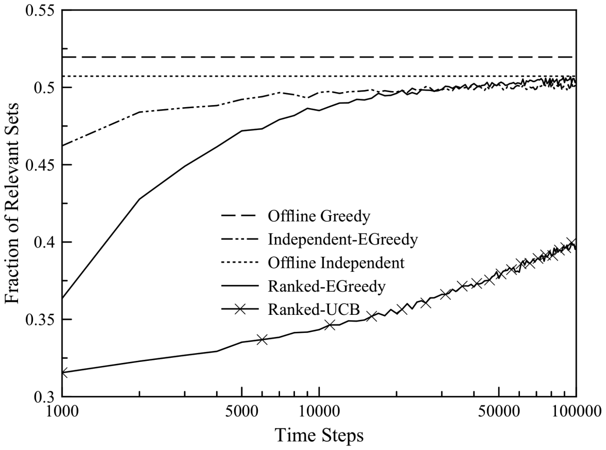

Figure 3: *Jester-Large* dataset with relevance threshold $\theta = 7$, the high threshold. Ranked-ϵGreedy and Independent-ϵGreedy perform similarly;Independent-ϵGreedy performs better until t=20000 but both remain very close, and well below the offline greedy optimum, for all 100000 time steps.

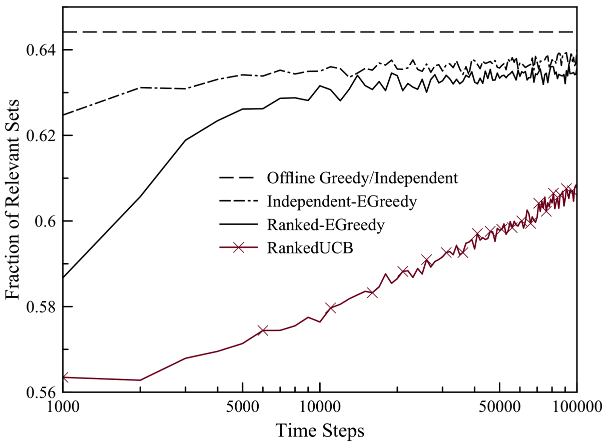

Figure 4: *Jester-Small* dataset with relevance threshold $\theta = 3.5$, the low threshold. In this case, the offline greedy and offline independent solution were the exact same set.

Our experiment consists of the following steps: at each time t, we draw a random user from the dataset and the algorithm recommends a set of $k = 5$ articles to display. We assume that the user "clicks" on any relevant articles displayed. If the user clicks on any articles, we get a payoff of 1 and a payoff of 0 otherwise. Each experiments consists of $T = 100000$ time steps and we average our results over 200 repetitions of each experiment. Performance of each algorithm was measured by the percent of sets that contained at least one relevant article to the user. We show datapoints at 1000 time step increments and each datapoint shown is the average set relevance over the last 1000 time steps.

Key Results. The results of our experiments are displayed in figures 1, 2, 3, and 4. Each plot shows the performance of the online algorithms as well as the offline benchmarks. The performance of Independent-ϵGreedy and Independent-UCB were roughly the same in all cases, so we omit the results for Independent-UCB for clarity. Our most surprising finding is the closeness of the greedy-optimal and the independent-optimal solutions. The largest difference between the two solutions, shown in figure 1, is 4%; if we displayed the greedy-optimal set of articles, approximately 92% of users will find at least one relevant article while if we displayed the independent-optimal set, then 88% of users will find at least one relevant articles. This finding suggests that in settings where a recommendation algorithm is catering to the tastes (as opposed to intents), explicit consideration of diversity may not be necessary since the independent-optimal solution yields similar results to the greedy-optimal solution.

Our second finding, which goes hand in hand with the previous one, is the favorable performance of the Independent Bandit Algorithm versus the performance of the Ranked Bandit Algorithm. In half of our experiments, either Independent-ϵGreedy or Independent-UCB perform strictly better than Ranked-ϵGreedy. In the experiments shown in figures 1 and 3, Ranked-ϵGreedy performs better than the independent solutions but only begins to perform better after 10000 or 20000 time steps. The faster learning rates of IBA compared to RBA demonstrates a key feature of IBA; the independence between bandit instances in different slots allows learning to happen in parallel as opposed to the de facto sequential learning in RBA. This parallel learning allows for a quicker convergence to the independent-optimal solution. In all cases, the Ranked-ϵGreedy algorithm doesn't converge to the value of the greedy-optimal solution within 100000 time steps.

Lastly, our experiments demonstrate a stark difference between the performance of Ranked-ϵGreedy and Ranked-UCB. As we noted at the end of section 4.1, learning for later slots in RBA is hindered by exploration in early slots. This effect is especially pronounced in the UCB1 algorithm when there are multiple articles that have high average rewards. The relatively low exploration rate of the ϵ-Greedy algorithm allows for faster convergence in earlier slots and hence a faster learning rate for later slots. In RBA, low exploration raises the risk of playing a sub-optimal article in an earlier slot but the gain from the faster learning rate outweighs that potential loss.

6 Conclusion

We've presented a simple online algorithm for the problem of abandonment minimization in recommendation systems which has near-optimal performance in the online problem. We have demonstrated, theoretically and empirically, that our approach trades off a small loss in offline performance for a faster learning rate and stronger performance in the online setting.

In the future, we would like to investigate the extension of these MAB techniques to general submodular utility functions. Additionally, we would like to investigate how to run algorithms such as IBA or RBA when it is only possible to observe user feedback on the set of articles but not on the individual articles within the set.

References

Agrawal, R.; Gollapudi, S.; Halverson, A.; and Ieong, S. 2009. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, 5–14. ACM.

Agrawal, S.; Ding, Y.; Saberi, A.; and Ye, Y. 2010. Correlation robust stochastic optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, 1087–1096. Society for Industrial and Applied Mathematics.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2):235–256.

Calinescu, G.; Chekuri, C.; Pál, M.; and Vondrák, J. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing* 40(6):1740–1766.

Chen, H., and Karger, D. 2006. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 429–436. ACM.

Goldberg, K.; Roeder, T.; Gupta, D.; and Perkins, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval* 4(2):133–151.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670. ACM.

Mahajan, D. K.; Rastogi, R.; Tiwari, C.; and Mitra, A. 2012. Logucb: an explore-exploit algorithm for comments recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 6–15. ACM.

Nemhauser, G.; Wolsey, L.; and Fisher, M. 1978. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming* 14(1):265–294.

Panigrahi, D.; Das Sarma, A.; Aggarwal, G.; and Tomkins, A. 2012. Online selection of diverse results. In *Proceedings of the fifth ACM international conference on Web search and data mining*, 263–272. ACM.

Radlinski, F.; Kleinberg, R.; and Joachims, T. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, 784–791. ACM.

Raman, K.; Shivaswamy, P.; and Joachims, T. 2012. Online learning to diversify from implicit feedback. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, 705–713. New York, NY, USA: ACM.

Robertson, S. E. 1977. The probability ranking principle in ir. *Journal of documentation* 33(4):294–304.

Slivkins, A.; Radlinski, F.; and Gollapudi, S. 2010. Learning optimally diverse rankings over large document collections. *arXiv preprint arXiv:1005.5197*.

Streeter, M.; Golovin, D.; and Krause, A. 2009. Online learning of assignments. In *Neural Information Processing Systems (NIPS)*.

Yue, Y., and Guestrin, C. 2011. Linear submodular bandits and their application to diversified retrieval. In *Neural Information Processing Systems (NIPS)*.