Combining Dynamic A/B Experimentation and Recommender Systems in MOOCs

Joseph Jay Williams
Harvard University
Cambridge, MA 02138
joseph_jay_williams@harvard.edu

Luong Hoang
Harvard University
Cambridge, MA 02138
lhoang@g.harvard.edu

Laurent Charlin
HEC Montreal
Montreal, Quebec H3T 2A7
laurent.charlin@hec.ca

ABSTRACT

We consider how dynamic A/B experiments – used to discover and deploy policies for personalizing items – can be interpreted from the perspective of a recommendation problem, where no prior data is available. We present an illustrative data set we collected, to evaluate algorithms for recommending emails (that vary along dimensions like subject lines) that will maximize response rate, from different subgroups of online learners. This problem is formalized as a contextual bandit, and we do an offline regret comparison of how standard bandit algorithms would perform in optimizing response rate. We report a system that provides an API for real–time data exchange and recommendation policy updates from algorithms from external machine learning researchers, and compare our best-performing offline algorithm – Thompson Sampling – against a randomized policy.

Keywords

AB Experimentation; Sequential Decision Making; Multi-armed Bandits; MOOCs; education

1. INTRODUCTION

Despite widespread applications of recommender systems, the majority of algorithms researchers develop are evaluated using offline data. There are few opportunities to test, in real-time, algorithms that choose item recommendations with the goal of balancing exploitation (maximizing user satisfaction) with exploration (testing out items in order to improve the underlying model in the long-run, potentially with short-term suboptimal performance). Even algorithms tackling these exploration-exploitation problems often have to use offline data [3]. If real-world applications were designed to enable algorithms to navigate this explore-exploit tradeoff through active experimentation, this could bring a range of novel computational problems to the forefront.

This poster aims to formulate sequential decision making in A/B experimentation in terms of a problem in personalized recommendation. The characteristics of this situation are atypical in most recommender systems, but arguably of interest in real–world applications. Optimal performance requires first discovering which features of items make for good recommendations on average, and then further gains in optimization are achieved by personalizing item recommendation.

In addition to this formulation, the other contributions are: 1. Collecting a data set for offline evaluation of algorithms for recommendation, that illustrates a problem at the intersection of dynamic A/B experimentation and recommendation. This involves recommending emails to MOOC participants that will maximize their response rate. 2. Our contextual bandit formulation of how to solve this email recommendation problem, and results from applying several standard bandit algorithms. 3. A system that allows *online* evaluation and comparison of algorithms for bandits/recommendation via an API that provides data and requests recommendations in real time 4. An online evaluation of the algorithm that performed best in our offline evaluation – Thompson Sampling – against a random policy, by dynamically changing the policy for experimentation/recommendation, as each email is sent and each participant's response observed.

2. RECOMMENDING EMAILS THAT MAXIMIZE RESPONSE RATE

The goal was to recommend emails that could be sent to participants, and maximize their response rate in providing feedback on an online course. In our email recommendation deployment, we chose three dimensions of the text of the email to change - subject line, introduction, body of email. Each of these three dimensions can be varied by writing different text. In this deployment, we chose to create three versions of each - three subject lines, three introduction messages, and three versions of the body of the email. This resulted in 27 unique email items, although any one dimension (with just three versions) could be analyzed independently (marginalizing over the others). This was because we designed the dimensions to be modular and randomized independently.

We consider just two user characteristics - user's age group (18-22, 23-26, 27-35, or 36 or older), and the number of days the user was active in the course (grouped as 0, 1, or 2 or more). While there are up to 50 other characteristics that are available about users, our preliminary analyses suggested these did not have substantial impact on which items would be recommended.

Following previous approaches for collecting a data set useful for testing a range of approaches and policies [2], we used a completely randomized policy, where all three item dimensions were independently randomized. Emails were sent to 3765 users.

3. CONTEXTUAL BANDIT FORMULATION

The problem of how to make person by person dynamic decisions about what email to recommend is formulated as a contextual multi–armed bandit problem [2].

Formally, each step in a sequence is indexed by t from $\{1, 2, .., T\}$, and furnishes a context vector x_t, with the choice of an action $a \in A$ producing an observed reward $r_{t,a}$.

The computational problem is to choose a sequence of actions $\{a_1, a_2, .., a_T\}$ that maximize the expected reward $E[\sum_{t=1}^{T} r_{t,a}]$.

4. MODELS,ALGORITHMS, RESULTS

We consider each email dimension (subject line, introduction, body) as a separate contextual bandit, so we solve three contextual bandit problems.

For each contextual bandit, our approach is twofold, following [2]. First, construct a model for the probability that a user responds to an email with a particular value on a dimension. We use logistic regression to predict the probability that a user – with specific context variables of age group and number of days active– will respond to each value of an email dimension, such as subject line 1, 2, or 3.

The reward $r_{t,a} \in \{0, 1\}$ depends on each x_t as well as the regression weights for action a, w_a. We use Bayesian logistic regression, since this gives us distributions over the weight parameters that we can use in Thompson Sampling.

Second, given these models, use an algorithm for selecting subsequent actions, which trades off maximizing response rate against collecting data that will informatively update the models. We compare Thompson Sampling [1], Upper Confidence Bound [2], epsilon–greedy, and a randomized policy. Given limited space, we describe only our logistic regression model with Thompson Sampling.

To do Thompson Sampling, we use the posterior distributions over each $\mathbf{w_a}$. At each time step t:

1. For each $a \in A$, sample $\mathbf{w}_a$.
2. Select $\underset{a}{argmax}\ r_{a,x_t}$
3. Observe reward r_{a,x_t} and update the posterior distribution on $\mathbf{w}_a$ (Bayesian logistic regression).

Figure 1 shows the regret for the algorithms we compared.

5. REAL-TIME POLICY CHANGE

To accomplish our goal of testing out these algorithms on *online* data, we built a system that updates the policy for recommending emails can be updated in real-time, after each email is sent and data is received. This system was constructed using the MOOClet framework, which enables any A/B experimentation infrastructure to adapt policies [4]. Specifically, an API (see MOOClet-provide-data, below) enables data about each user (or groups of users) to be provided on demand. A second key API call (see MOOClet-Request-Recommendation) provides a user's context variables to the system and requests from an algorithm which email item should be recommended.

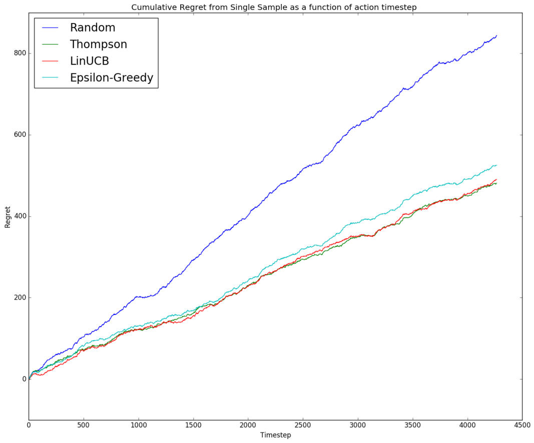

Figure 1: Regret comparison for Contextual Bandit algorithms using Regularized Logistic Regression.

To compare alternative algorithms, which algorithm is called to provide a recommendation can itself be randomized, interleaving the algorithms as users are selected. This will enable the algorithms analyzed for this offline data to be compared against each other in this real-time situation, as well as the evaluation of any other algorithm, from both the recommender systems and multi-armed bandits literature.

As a test deployment of our system, we compared the response rate of a new group of 1775 participants when recommended emails using a random policy (4.5%) against a heuristic that approximated Thompson Sampling (7.2%).

```
MOOClet-Provide-Data:
PROVIDES: For each of N participants, User Context
Variables (Age Group, Number Days Active), Item
Assigned (Email Subject Line, Introduction, Body),
Response (0 or 1)

MOOClet-Request-Recommendation:
PROVIDES: User Context Variables (Age Group,
Number Days Active)
RETURNS: Item assignment (Email Subject Line,
Introduction, Body)
```

6. REFERENCES

[1] O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

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

[3] H. P. Vanchinathan, I. Nikolic, F. De Bona, and A. Krause. Explore-exploit in top-n recommender systems via gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 225–232. ACM, 2014.

[4] J. J. Williams, N. Li, J. Kim, J. Whitehill, S. Maldonado, M. Pechenizkiy, L. Chu, and N. Heffernan. The mooclet framework: Improving online education through experimentation and personalization of modules. *Available at SSRN 2523265*, 2014.