Personalized news recommendation

Abstract

Personalized web services strive to adapt their services (advertisements, news articles, etc.) to individual users by making use of both content and user information. News recommendation is a field different from traditional recommendation fields. News articles are created and deleted continuously with a very short life cycle. Users' preference is also hard to model since they can easily be attracted by things happening around them. One of the major challenges is recommendations for new users, also known as the cold-start problem. Now, there are some studies which propose a bandit approach to this problem, but they fall short on the recommendations to users with a lot of behavioral history.
This study proposes a hybrid algorithm which adapts as the user interacts with the environment. Along with this, we also propose a framework for unregistered users for which we have no information about.

1. Introduction

The user base can be divided into two different categories the unregistered and registered users. For the unregistered users, no prior knowledge about the user is observed and in the case of registered users, some level of demographic, geographic or behavioral information is available to us.

The unregistered users have temporary click information for a particular session. Hence, the expected reward can only be predicted by features based on articles including URL categories, editor categories, and topic. Now if the article considered is new and thus has no user interactions, it becomes very hard for the MAB to find the expected reward. On the other hand for articles with large user interactions are easy to model the expected payoffs.

In the case of registered users, we face a different challenge of a cold start. We do not have any behavioral information for new users and thus are hard to model.

We solve the challenges by introducing a hybrid model for both cases:

1. Registered user: We have two algorithms contextual bandit for all users and linear bandit for each user. For a new user, contextual bandit algorithm will be used and as the behavior information increases, we shift to linear bandit for each user.
2. Unregistered user: Similar to above we have contextual bandit for all articles and linear bandit for new articles. Linear bandit will use features based on the topic of article and URL, editor categories to predict expected reward and as the user interactions for article increases, we can use MAB using the click rate of the article.

The framework discussed above can be summarised as:

Unregistered users		Registered users	
Articles without history	Linear bandit	High behavioral information	Linear bandit for each user
Articles with history	MAB of articles	Cold Start (new user)	Contextual bandit

2. Model Description

2.1 For the Unregistered user

Two algorithms defined and both run simultaneously. The detailed description of algorithms are given below

2.2.1 Linear bandit

This algorithm is effective for new articles where limited user interactions are observed. Hence, we rely on other articles to predict the expected reward of the article. The context of an article is based on the topic of article and URL, editor categories. Here, LinUCB algorithm can be used. Each article is represented by a d-dimensional vector $x_{t,a}$ and with some unknown coefficient vector Θ_a now, we assume expected payoff of an article is linear in its feature vector:

$$E[r_{t,a}|x_{t,a}] = (x_{t,a})^T * \Theta_a$$

Let D_a be a matrix of *m*X*d* with contexts for m training inputs and b_a be the response vector (corresponding m click/no-click feedback). Using ridge regression, we get the estimate of coefficient as:

$$\Theta^*_a = (D_a^T D_a + I_d)^{-1} D^T{}_a c_a \qquad I_d \text{ is d*d identity matrix}$$

Now, for better use of the data, we propose to cluster the articles into k clusters using unsupervised learning like k-means. Once we have a new article we place it in one cluster and use the parameters for the corresponding cluster to estimate the reward.
Now, it can be shown that the Upper Confidence Bound for the article is

$$\alpha\sqrt{x_{a,t}^T A_a^{-1} x_{t,a}} + x_{a,t}^T \theta \qquad \text{where } A_a = D_a^T D_a + I_d$$

While calculating θ_a, we feed the past m trials of articles from its cluster in the matrix D_a.
Observe that the tightness on the bound is $\alpha\sqrt{x_{a,t}^T A_a^{-1} x_{t,a}}$,where $A_a = D_a^T D_a + I_d$

2.1.2 Multiarm bandits

This algorithm will be useful for articles which have decent user interactions for us to estimate the reward based on history. For this, we use a simple UCB1 for multiarm bandits algorithm. Assume that the past reward of the article for a given time interval is given by $[X_{A,j}]$:

$$x_{a,j} = \text{reward at time j for article a.}$$

The expected reward is $E[x_a]$ and UCB is: $E[x_a] + \alpha \sqrt{\frac{2ln(n)}{n_j}}$

Where n = total number of articles shown; n_j is the number of times this article was shown. Observe that the tightness of bound is given by $\alpha \sqrt{\frac{2ln(n)}{n_j}}$

2.1.3 Putting it all together

Now we run both the algorithms on all articles. Hence, we have UCB and corresponding tightness of bound by both algorithms.

<u>Rule:</u> For each article, compare the tightness of bounds for both and choose UCB of the algorithm with more tight bound i.e. tightness is lower.
Now after this is done for the articles, we choose the article with the highest UCB. We also ensure that the same article is not shown twice in a single session.

2.1.4 Further Improvements

As mentioned before unregistered users have temporary click information for a particular session. Hence, we can use this for keeping an affinity vector of the user with a cluster of articles and utilize this information for choosing the article from the selected cluster.

2.2 Registered user

2.2.1 Contextual bandit

Here we have demographic and geographic features for each user. Same as the case of unregistered users we cluster users into groups.Users within the same latent class share similar interests and behaviors. Now, we can formulate in two possible ways:

2.2.1.1 EXP4 algorithm

At every time step, we provide expert advice of probability over arms. This can be modeled separately for each class. We use the clusters of articles made in linear bandit for unregistered users and then calculate the probability of the class choosing article in the article cluster.
Expert advice $E_{t,a}$ = Probability that user class read the article in the cluster of article a.
The algorithm goes as follows:

In round t=1,2,…, the following things happen: (Q_1 initialized to be a uniform distribution)

1. The advice E_t is received
2. Choose the action $A_t \sim P_{t,\cdot}$ at random, where $P_t=Q_tE_t$
3. The reward $X_t=x_{t,A_t}$ is received
4. The rewards of all the actions are estimated; say: $\widehat{X}_{ti} = 1-\frac{I\{A_t=i\}}{P_{ti}+\gamma}(1-X_{ti})$
5. Propagate the rewards to the experts: $\overline{X}_t = \widehat{X}_t E_t$
6. The distribution Q_t is updated using exponential weighting:
7. $Q_{t+1,i} = \frac{exp(\eta\widehat{X}_{ti})Q_{ti}}{\sum_j exp(\eta\widehat{X}_{ti})Q_{ti}}$

2.1.1.2 LinUCB algorithm

The algorithm runs similar to 2.1.1 with a simple alteration of the vector representation of the article. Along with the features relating to articles, we also include information on the latent class of the user and its affinity with the article cluster.

2.2.2 Linear bandit for each user

An algorithm similar to 2.1.1. With the exception that the weight parameters are learned separately for all the users and thus the Design matrix used to train the weight consists of the past interactions of the user with different articles. The differences are summarised as:
θ^u_t : Separate weight for each user.
D^u_a: Past interactions of the user with different articles

2.2.3 Putting it all together

If LinUCB is used for both the algorithms then the tightness of bound logic will work here as well.
But if EXP4 algorithm is used for 2.2.1, we can come up with a threshold on the Linear bandit for each user and if the tightness of bound becomes lower than the threshold, we use Linear bandit otherwise EXP4 algorithm is used.

2.2.4 Further improvements

For the registered user, instead of deciding on rules for switching in between algorithms, a UCB-based Meta learner or Episodic Meta-learner (LSTM based) might be used to learn the policy of choosing any of the two algorithms.

3. Related work

Using a hybrid framework for recommendations has its own advantages, however, to the best of our knowledge, no research effort has been paid on exploring this important property. However, an ensemble recommendation is explored in paper by Liang Tang, Yexi Jiang, Lei Li, Tao Li of Florida university[1]. Their paper explores the possibility of stabilizing the CTR estimation of web objects by integrating the advantages of different bandit policies and using meta-learning.
Aside from these, the paper by Larisa Schwartz, Genady Ya. Grabarnik, NY, USA[2] proposes an online interactive collaborative filtering with clustering articles. A similar approach is seen in the paper by Li Zhou and Emma Brunskill of Carnegie Mellon University[3] on leveraging a set of learned latent user classes for new users to solve the cold start problem.

4. References

[1] Ensemble Contextual Bandits for Personalized Recommendation
[2] Online Interactive Collaborative Filtering Using Multi-Armed Bandit with Dependent Arms
[3] Latent Contextual Bandits and their Application to Personalized Recommendations for New Users