News Personalization

Li Zhou, Saloni Potdar, Emma Brunskill (advisor)
AGENDA

- Motivation
- Background
- Acting when a set of contextual MABs is known
- Learning from scratch
- Experiments Results
- Conclusion and Future Work
What is news personalization?

- **Customize** news feed based on users’ interests
- Particularly, we are interested in the **Cold Start** problem: How to personalize news for a new user?
- **Goal**: Maximize user engagement
Existing Work

- LinUCB (Li, Lihong et al. 2010), a contextual-bandit approach to news personalization
- Thompson sampling (Chapelle, Olivier et al. 2011), An Empirical Evaluation of Thompson Sampling
- tUCB (Lazaric, Alessandro et al. 2013) Sequential transfer in multi-armed bandit with finite set of models
Key Contributions

- Proposed a latent class contextual Multi-armed bandit algorithm
- Developed a web framework for Yahoo! InMind News Personalization
- Performed simulation results with Yahoo! frontpage datasets
- Conducted user studies
Multi-armed Bandit 101

- Take news personalization as an example
  - There are a bunch of articles in our news pool
  - Users come sequentially and ready to be entertained.
Multi-armed Bandit 101

- At each time we want to select one article for a user
Multi-armed Bandit 101

- The goal is to maximize user engagement
- In our case, engagement is defined by click through rate and potential time on news articles
Multi-armed Bandit 101

- We update the model once we get the feedback from the user.
Multi-armed Bandit 101

- We update the model once we get the feedback from the user.
Multi-armed Bandit 101

- Different from supervised learning because we do not have feedback of articles we didn’t pick
Modeling News Personalization as Contextual Multi-armed Bandit Problem

- Select news articles based on current context such as users’ profile and articles content
- **Pros:** Trade-off between acquiring new information (exploration) and capitalizing on the information available so far (exploitation). Able to handle cold start problem.
LinUCB (Li, Lihong 2010) for News Personalization

- Expectation of reward of each arm is modeled as a linear function of the context
  \[ \mathbb{E}[r_{t,a} \mid x_{t,a}] = x_{t,a} \theta_a^* \]

- The goal is to minimize regret, defined as the difference between the expectation of the reward of best arms and the expectation of the reward of selected arms.
  \[ R_A(T) \overset{\text{def}}{=} \mathbb{E} \left[ \sum_{t=1}^{T} r_{t,a_t^*} \right] - \mathbb{E} \left[ \sum_{t=1}^{T} r_{t,a_t} \right] \]
LinUCB (Li, Lihong 2010) for News Personalization

- For a given context, we estimate the reward and the confidence interval

\[ a_t^\text{def} = \arg \max_{a \in A_t} \left( x_{t,a} \hat{\theta}_a + \alpha \sqrt{x_{t,a}^T A_a^{-1} x_{t,a}} \right) \]

- Select arm based on the upper bound of confidence interval
Thompson Sampling (Chapelle, Olivier 2011) for Contextual Bandits

- Once used in Yahoo! news product
- Idea: randomly select each article according to its probability of being optimal

- Assume the distribution of reward depends on the model parameter and context: $P(r|\theta, c)$
- Given a prior $P(\theta)$, we can get the posterior based on the above likelihood. Normally in linear regression we use a Gaussian prior with a Gaussian posterior.
Thompson Sampling (Chapelle, Olivier 2011) for Contextual Bandits

- Each time we sample a $\theta$ from current posterior, and select the best news article based on that $\theta$, after receiving feedback from a user, we update the posterior distribution.

```
Algorithm 1 Thompson sampling

$D = \emptyset$

for $t = 1, \ldots, T$ do
    Receive context $x_t$
    Draw $\theta^t$ according to $P(\theta|D)$
    Select $a_t = \arg \max_a E_r(r|x_t, a, \theta^t)$
    Observe reward $r_t$
    $D = D \cup (x_t, a_t, r_t)$
end for
```
Our proposed method

A latent class contextual multi-armed bandit algorithm
Users in different groups behaves differently

- Teenagers’ like sport news, mid-aged care political news; teachers want education news, engineers love technology news. Each group behavior differently
- Can use one Multi-armed bandit model for each group
  - How to get these Multi-armed bandit models?
  - How to identify which model a user belongs to?
  - Is there any latent knowledge between different we can utilize?
Two different settings

- Already have a set of contextual Multi-armed bandit models (MABs)
  - This can be done by offline training using history data
  - Each contextual MAB represents a certain type of user
  - Need to identify which model a user belongs to

- No ground truth models are given, pure cold start problem
  - Need to learn these contextual MABs from scratch
Acting when a set of MABs is known

**Goal:** identify which model a user belongs to

**Method 1:**
- Estimate reward of current user by LinUCB
- Maintain a subset of models that is within the confidence interval of the estimated model
- The model that user belongs to is in the subset
Acting when a set of MABs is known

Method 1:

[Diagrams showing empirical estimates for Article 1 and Article 2]
Acting when a set of MABs is known

Method 1:

article 1

article 2
Acting when a set of MABs is known

Method 1:

article 1

article 2
Acting when a set of MABs is known

Method 1:

- **article 1**
  - Empirical estimate
  - Article 1

- **article 2**
  - Empirical estimate
  - Article 2
Acting when a set of MABs is known

Method 1:

article 1

article 2
Acting when a set of MABs is known

Method 2 (Based on Generalized Thompson Sampling (Li, Lihong 2013)):

- Treat each known contextual MABs as prior models
- Update the probability of a user belongs to each model based on the click feedback
  - Penalize each contextual MABs based on the mismatch in its prediction and the observed reward
Acting when a set of MABs is known

Method 2 (Based on Generalized Thompson Sampling (Li, Lihong 2013)):

- Let \( W_t = \{w_{i,t}\} \) be the posterior weight vector, representing the probability of the user belongs to each model.
- Probability of selecting arm \( a \):
  \[
  P(a) = \sum_{i=1}^{N} \frac{w_{i,t}1(\varepsilon_i(x_t) = a)}{W_t}
  \]
- Let \( f_i(x_t, a_t) \) be the prediction function of model \( i \), and \( \ell(f_i(x_t, a_t), r_t) \) be the loss of model \( i \) compared to the reward, then the update rule is
  \[
  \forall i : w_{i,t+1} \leftarrow w_{i,t} \cdot \exp(-\eta \cdot \ell(f_i(x_t, a_t), r_t))
  \]
  \[
  W_{i+1} \leftarrow \sum_i w_{i,t+1}
  \]
Can we learn from scratch?
A purely cold start
Learning a finite set of contextual MABs

- We have click feedbacks and contexts from different users, how can we learn a set of latent models (contextual MABs)?
- Possible solution
  - Spectral methods (Anandkumar, Anima 2012)
    - match moments (up to third order) to get consistent estimator
  - Better theoretical guarantee
  - Global optima
  - Our future work
- EM style algorithm
  - Mixture of linear regression
  - Local optima
A Mixture of Linear Regression approach

- Definition of mixture of linear regression
  - A mixture of linear regression model is given by
    \[
    y_i = \begin{cases} 
    x_i^T \beta_1 + \epsilon_{i1} & \text{with probability } \pi_1, \\
    x_i^T \beta_2 + \epsilon_{i2} & \text{with probability } \pi_2, \\
    \vdots & \\
    x_i^T \beta_J + \epsilon_{ij} & \text{with probability } \pi_J
    \end{cases}
    \]
  - \(\epsilon_{ij}\) are random errors, and we assume it is from a Gaussian distribution \(\epsilon_{ij} \sim N(0, \sigma_j^2)\)
  - Solved by EM algorithm
Learning from Scratch

- learn a mixture of linear regression model from the click feedback and context before interacting with the next user
- Treat the learned model as prior for the next user

Input: number of latent model $M$, number of steps $T$
Initialization: $\Theta_{-1} = \text{Initial Theta}$, Samples $S = []$
For each user $u = 1, \ldots, U$:
  - Run Generalized Thomson Sampling with $\Theta_{\{u-1\}}$, get samples $S_u$
  - $S = S_{\{1,\ldots,u\}}$
  - Run mixture of linear regression with $S$ to get $\Theta_{\{u\}}$
Experiments Results
Results on Synthetic Dataset

Assuming a set of MABs are known

- Ground truth model
Results on Synthetic dataset

Assuming a set of MABs are known

- When each user interact 100 times
Results on Synthetic dataset

Assuming a set of MABs are known

- When each user interact 200 times
Yahoo! Frontpage Dataset

- Yahoo! Front Page Today Module User Click Log Dataset, version 1.0
- Each user and each article has 6 features
- Each data instance contains 1 user and about 20 news articles, which article is showed and whether the user clicked or not
- Articles are associated with an id, users are not
- Number of total instance: about 40 million
- Average CTR: ~4%
Experimental Setting

- User id: If two users have the same feature vector, treat them as the same user.
- Feature space: the outer product of user features and article features.
- Interact with a user 100~500 times, then switch to next user.
- Cluster user into 2 ~ 6 groups, train a supervised model and treat them as ground truth model.
Results on Yahoo! frontpage dataset
Assuming a set of MABs are known

- Performance for different number of clusters

![Bar chart showing performance for different number of clusters](chart.png)
Results on Yahoo! frontpage dataset

Assuming a set of MABs are known

- Performance for different number of training users
Results on Yahoo! frontpage dataset

Assuming a set of MABs are known

- Incrementally increase training users and predict the next 10 users

![Graph showing CTR vs. #Training Users with #cluster=4, comparing Generalized TS, LinUCB, and Baseline.](image-url)
Future Work

- Conducting the user study and evaluating performance
- Derive theoretical bound of our algorithm
- Try spectral methods instead of EM
- Handle duplicate articles from different sources
- Feature engineering - NER from articles to identify more specific subtopic user is interested in.
- Problem of “Invasive Valley” - similar to uncanny valley
Questions?
STOP
BACKUP SLIDES
Acting when a set of MABs is known

Method 1:

Input: a set of MAB models $W=\{w_1, w_2, ..., w_m\}$, number of steps $T$, number of arms $K$

For each user:
  Initialize LinUCB model
  For $t=1, ..., T$
    Build $\Theta_t = \{\theta : \forall i, |w^k_m \cdot x - w^k_c| \leq \epsilon_{k,t}\}$
    Select $\theta_t = \arg \max_{\theta_t} \{w^k_m \cdot x\}$
    Pull the best arm in $\theta_t$
    Update LinUCB model
Acting when a set of MABs is known

Method 2 (Generalized Thompson Sampling):

Input: a set of MAB models $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N$, learning rate $\eta$, prior $p$

For each user:
- Initialize posterior: $w_1 \leftarrow p; W_1 \leftarrow ||w_1||_1$
- For $t=1,\ldots,T$
  - Receive context $x_t$
  - Select arm according to the mixture probabilities $P(a)$
  - Observe reward and update weights
A Mixture of Linear Regression
approach

- EM algorithm for solving mixture of linear regression
  - Log-likelihood
    \[
    L(\theta|x_1, \ldots, x_n, y_1, \ldots, y_n) = \sum_{i=1}^{n} \log \left( \sum_{j=1}^{J} \pi_j \phi_j(y_i | x_i) \right)
    \]
  - \(\phi_j(y_i | x_i)\) denotes the density of an univariate Gaussian with mean \(x_i^T \beta_j\) and error \(\sigma_j^2\)
  - E-step
    \[
    w_{ij}^{(r)} = \frac{\pi_j^{(r)} \phi_j(y_i | x_i)}{\sum_{j=1}^{J} \pi_j^{(r)} \phi_j(y_i | x_i)} \quad (i = 1, \ldots, n; j = 1, \ldots, J)
    \]
  - M-step
    \[
    \hat{\pi}_j^{(r+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(r)}}{n} \quad (j = 1, \ldots, J)
    \]
    \[
    \hat{\beta}_j^{(r+1)} = (X^T W_j X)^{-1} X^T W_j Y \quad (j = 1, \ldots, J)
    \]