Clustering of Conversational Bandits for User Preference Learning and Elicitation

Junda Wu[1] Canzhe Zhao [2] Tong Yu [3]
Jingyang Li [4] Shuai Li [2]

[1]New York University

[2]Shanghai Jiao Tong University

[3]Carnegie Mellon University

[4]Chinese Academy of Sciences

September 26, 2021

Motivation - Conversational Recommender Systems

Traditional recommender systems:

- sample inefficient
- cold-start problem

Conversational recommender systems:

- more sample efficient
- mitigate the cold-start problem

Motivation - Conversational Recommender Systems

Traditional recommender systems:

- sample inefficient
- cold-start problem

Conversational recommender systems:

- more sample efficient
- mitigate the cold-start problem

Motivation - Limitations of existing CRSs

- Conversational key-terms need to be carefully labeled by humans
- Granularity of the key-terms labeled by humans is usually fixed

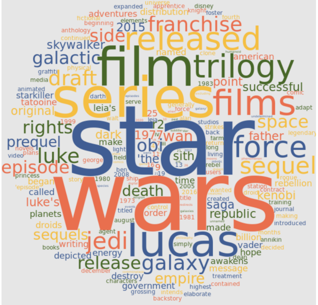

Figure 1. (a) Key-terms with fixed granularity labled by humans (in red box)[1]. (b) Finer granularity of key-terms is usually needed in CRSs[2].

[1] Star Wars: Episode IX - *The Rise of Skywalker* (https://www.imdb.com)
[2] https://wordcloudapi.com

Motivation - Limitations of existing CRSs

- Conversational key-terms need to be carefully labeled by humans
- Granularity of the key-terms labeled by humans is usually fixed

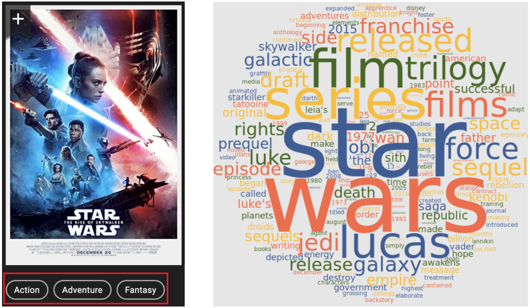

Figure 1. (a) Key-terms with fixed granularity labled by humans (in red box)[1]. (b) Finer granularity of key-terms is usually needed in CRSs[2].

[1] Star Wars: Episode IX - *The Rise of Skywalker* (`https://www.imdb.com`)

[2] `https://wordcloudapi.com`

Problem Formulation

- N users and M items
- Item i's feature vector: $x_i \in \mathbb{R}^d$
- Key-term k's feature vector: $\tilde{x}_k \in \mathbb{R}^d$
- $\theta_u \in \mathbb{R}^d$ and $\tilde{\theta}_u \in \mathbb{R}^d$ are user preference vectors on items and key-terms respectively
- Receive rewards $r_{i_t,t} = \theta_{u_t}^\top x_{i_t} + \epsilon_t$ and $\tilde{r}_{k_t,t} = \tilde{\theta}_{u_t}^\top \tilde{x}_{k_t} + \tilde{\epsilon}_t$ after recommending item i_t and conducting one conversation on key-term k_t respectively
- The conversation frequency: $g(t)$[1]

Problem Formulation

- N users and M items
- Item i's feature vector: $x_i \in \mathbb{R}^d$
- Key-term k's feature vector: $\tilde{x}_k \in \mathbb{R}^d$
- $\theta_u \in \mathbb{R}^d$ and $\tilde{\theta}_u \in \mathbb{R}^d$ are user preference vectors on items and key-terms respectively
- Receive rewards $r_{i_t,t} = \theta_{u_t}^\top x_{i_t} + \epsilon_t$ and $\tilde{r}_{k_t,t} = \tilde{\theta}_{u_t}^\top \tilde{x}_{k_t} + \tilde{\epsilon}_t$ after recommending item i_t and conducting one conversation on key-term k_t respectively
- The conversation frequency: $g(t)$[1]

Problem Formulation

- N users and M items
- Item i's feature vector: $x_i \in \mathbb{R}^d$
- Key-term k's feature vector: $\tilde{x}_k \in \mathbb{R}^d$
- $\theta_u \in \mathbb{R}^d$ and $\tilde{\theta}_u \in \mathbb{R}^d$ are user preference vectors on items and key-terms respectively
- Receive rewards $r_{i_t,t} = \theta_{u_t}^\top x_{i_t} + \epsilon_t$ and $\tilde{r}_{k_t,t} = \tilde{\theta}_{u_t}^\top \tilde{x}_{k_t} + \tilde{\epsilon}_t$ after recommending item i_t and conducting one conversation on key-term k_t respectively
- The conversation frequency: $g(t)$[1]

Problem Formulation

Learning objective: minimizing the expected cumulative regret[2]

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} \theta_{u_t}^\top x_{u_t}^* - \theta_{u_t}^\top x_{i_t}\right].$$

Challenges:

- No key-terms labeled by humans?
  - Cluster the items to generate meaningful key-terms!
- How to better elicit user preferences in conversations?
  - Conduct conversations in a coarse-to-fine grained manner!

Problem Formulation

Learning objective: minimizing the expected cumulative regret[2]

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} \theta_{u_t}^{\top} x_{u_t}^{*} - \theta_{u_t}^{\top} x_{i_t}\right].$$

Challenges:

- No key-terms labeled by humans?
  - Cluster the items to generate meaningful key-terms!
- How to better elicit user preferences in conversations?
  - Conduct conversations in a coarse-to-fine grained manner!

Model

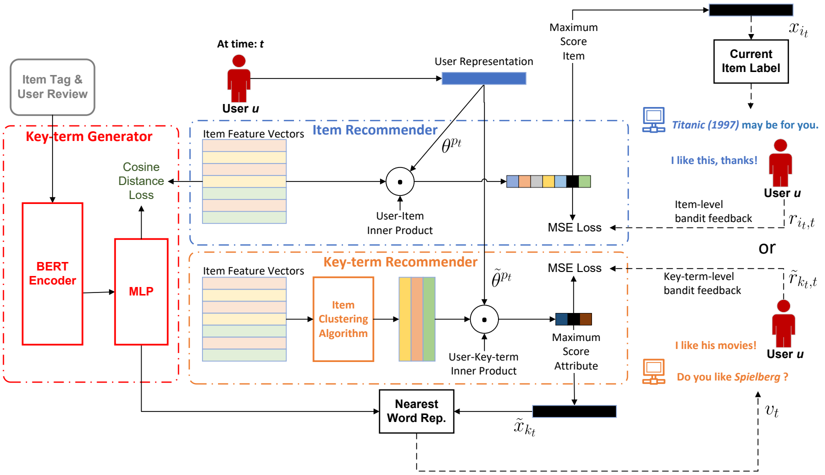

Figure 2. The model proposed by us.

Algorithm - Conversational Key-term Generation

No key-terms labeled by humans?

- Adopt *k*-means method to cluster the feature vectors of items to generate the feature vectors of key-terms
- Clustering number:

$$k(t) = \min\left(\left\lfloor \frac{M \log(1 + \delta t / T)}{\log(2)} \right\rfloor + 1, M\right).$$

- Entity names of key-terms are obtained by using a mapping constructed via a fine-tuned BERT and a three-layer MLP

Algorithm - Conversational Key-term Generation

No key-terms labeled by humans?

- Adopt *k*-means method to cluster the feature vectors of items to generate the feature vectors of key-terms
- Clustering number:

$$k(t) = \min\left(\left\lfloor \frac{M\log(1+\delta t/T)}{\log(2)} \right\rfloor + 1, M\right).$$

- Entity names of key-terms are obtained by using a mapping constructed via a fine-tuned BERT and a three-layer MLP

Algorithm - Conversational Key-term Generation

No key-terms labeled by humans?

- Adopt *k*-means method to cluster the feature vectors of items to generate the feature vectors of key-terms
- Clustering number:

$$k(t) = \min\left(\left\lfloor \frac{M\log(1+\delta t/T)}{\log(2)} \right\rfloor + 1, M\right).$$

- Entity names of key-terms are obtained by using a mapping constructed via a fine-tuned BERT and a three-layer MLP

Algorithm - Conduct Conversations on User Clusters

How to better elicit user preferences in conversations?

- Estimate user preference on key-terms:

$$\tilde{\theta}'_u = \arg\min_{\theta} \sum_{k=1}^{\tilde{T}_u} \left(\theta^\top \tilde{x}_k - \tilde{r}_k\right)^2 + \tilde{\lambda}\|\theta\|_2^2 \,.$$

- Estimate user preference on items:[3]

$$\theta'_u = \arg\min_{\theta} \; \lambda \sum_{k=1}^{T_u} \left(\theta^\top x_k - r_k\right)^2 + (1-\lambda)\|\theta - \tilde{\theta}'_u\|_2^2 \,.$$

- Cluster the users based on θ'_u

Algorithm - Conduct Conversations on User Clusters

How to better elicit user preferences in conversations?

- Estimate user preference on key-terms:

$$\tilde{\theta}'_u = \arg\min_{\theta} \sum_{k=1}^{\tilde{T}_u} \left(\theta^\top \tilde{x}_k - \tilde{r}_k\right)^2 + \tilde{\lambda}\|\theta\|_2^2 \,.$$

- Estimate user preference on items:[3]

$$\theta'_u = \arg\min_{\theta} \; \lambda \sum_{k=1}^{T_u} \left(\theta^\top x_k - r_k\right)^2 + (1-\lambda)\|\theta - \tilde{\theta}'_u\|_2^2 \,.$$

- Cluster the users based on θ'_u

Algorithm - Conduct Conversations on User Clusters

How to better elicit user preferences in conversations?

- Estimate user preference on key-terms:

$$\tilde{\theta}'_u = \arg\min_{\theta} \sum_{k=1}^{\tilde{T}_u} \left(\theta^\top \tilde{x}_k - \tilde{r}_k\right)^2 + \tilde{\lambda}\|\theta\|_2^2 \,.$$

- Estimate user preference on items:[3]

$$\theta'_u = \arg\min_{\theta} \; \lambda \sum_{k=1}^{T_u} \left(\theta^\top x_k - r_k\right)^2 + (1-\lambda)\|\theta - \tilde{\theta}'_u\|_2^2 \,.$$

- Cluster the users based on θ'_u

- Closed-form solutions of $\tilde{\theta}'_u$ and θ'_u:

$$\tilde{\theta}'_u = \tilde{S}_u^{-1}\tilde{b}_u, \ \theta'_u = S_u^{-1}(b_u + (1-\lambda)\tilde{\theta}'_u)\,,$$

where

$$S_u = (1-\lambda)\boldsymbol{I} + \lambda\sum_{k=1}^{T_u} x_k x_k^\top, \ b_u = \lambda\sum_{k=1}^{T_u} x_k r_k\,,$$

$$\tilde{S}_u = \tilde{\lambda}\boldsymbol{I} + \sum_{k=1}^{\tilde{T}_u} \tilde{x}_k\tilde{x}_k^\top, \ \tilde{b}_u = \sum_{k=1}^{\tilde{T}_u} \tilde{x}_k\tilde{r}_k\,.$$

Algorithm - Conduct Conversations on User Clusters

Interact with user clusters:

- Recommend to user cluster p using UCB-based strategy[4]

$$i_t = \arg\max_{i \in \mathcal{A}} x_{i_t}^{\top} \theta_t^{p'} + \lambda\alpha \|x_{i_t}\|_{(S^p)^{-1}} + (1-\lambda)\tilde{\alpha} \|(S^p)^{-1} x_{i_t}\|_{(\tilde{S}^p)^{-1}} .$$

- Conduct conversations to user cluster p to faster reduce the uncertainty coming key-terms

$$k_t = \arg\max_{k \in \mathcal{K}} \frac{\|X(S^p)^{-1}(\tilde{S}^p)^{-1}\tilde{x}_k\|_2^2}{1 + \|\tilde{x}_k\|^2_{(\tilde{S}^p)^{-1}}} .$$

Interact with user clusters:

- Recommend to user cluster p using UCB-based strategy[4]

$$i_t = \arg\max_{i \in \mathcal{A}} x_{i_t}^{\top} \theta_t^{p'} + \lambda\alpha \|x_{i_t}\|_{(S^p)^{-1}} + (1-\lambda)\tilde{\alpha} \|(S^p)^{-1} x_{i_t}\|_{(\tilde{S}^p)^{-1}} .$$

- Conduct conversations to user cluster p to faster reduce the uncertainty coming key-terms

$$k_t = \arg\max_{k \in \mathcal{K}} \frac{\|X(S^p)^{-1}(\tilde{S}^p)^{-1}\tilde{x}_k\|_2^2}{1 + \|\tilde{x}_k\|_{(\tilde{S}^p)^{-1}}^2} .$$

Analysis

Theorem

Under mild assumptions, let $\beta = \sqrt{\lambda}\sqrt{d \log T} + (1-\sqrt{\lambda})\sqrt{d + \log T}$) *and* m *be the number of the underlying user clusters, then the cumulative regret of the algorithm CtoF-ConUCB+ after* T *rounds satisfies*

$$R(T) \leq \sum_{j=1}^{m} 4\beta\sqrt{dp^j T \log(T/d)} + \mathcal{O}\left(\left(\frac{1}{p} + \frac{1}{p\gamma^2\lambda_x^3}\right)\log T\right).$$

The improvements come from the clustering of bandits[5] and the conversations conducted on user clusters.[6]

Synthetic Datasets

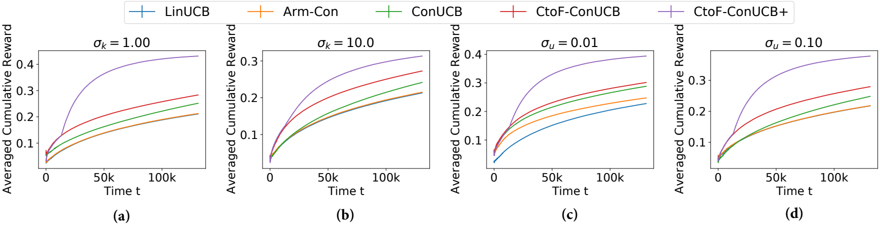

Figure 3. Results on synthetic datasets with varying levels of item similarities in (a) (b), and user similarities in (c) (d).

Observations:

- Key-term generator works
- Conversations conducted on user clusters could further improve the performance

Real Datasets

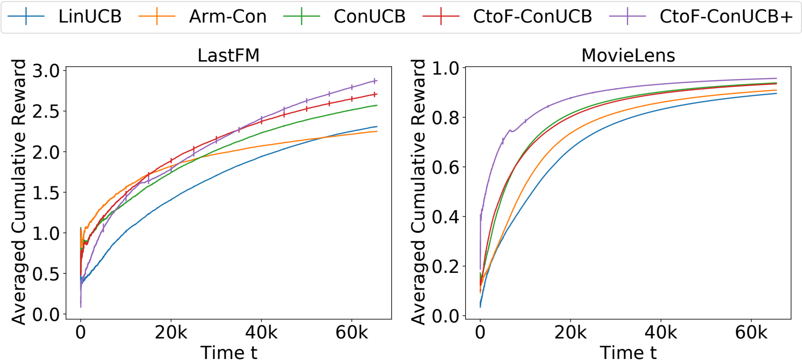

Figure 4. Results on LastFM and MovieLens 25M datasets.

Observations:

- CtoF-ConUCB+ still clearly outperforms ConUCB in real datasets

Real Datasets

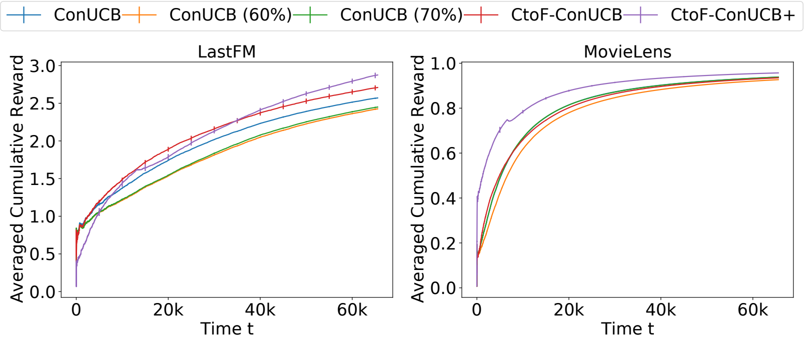

Figure 5. The effects of missing key-terms in ConUCB.

Observations:

- The performance of ConUCB starts to degrade when more than 30% key-terms are removed

Use Case

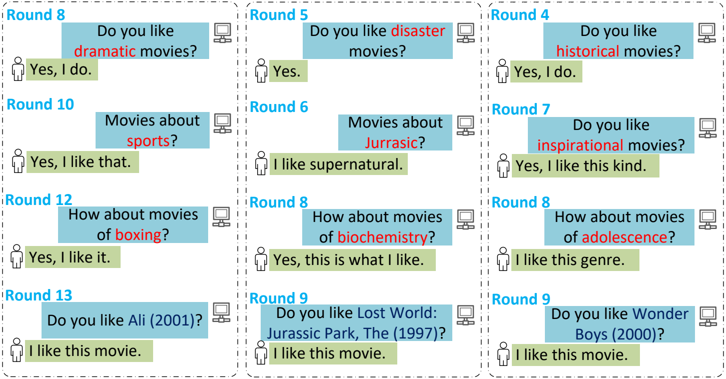

Figure 6. Three use case examples from the logged results.

Our algorithm learns to generate meaningful coarse-to-fine grained conversational key-terms.

Conclusions

In this work:

- Automatic generation of conversational key-terms by item clustering and semantic mapping
- Elicit user preferences in a coarse-to-fine grained manner by conducting conversations on user clusters

The End

- Q&A?
- Thank you!

References I

- Li, Lihong et al. "A contextual-bandit approach to personalized news article recommendation". In: *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*. ACM, 2010, pp. 661–670.
- Li, Shuai et al. "Improved Algorithm on Online Clustering of Bandits". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 2923–2929.
- Xie, Zhihui et al. "Comparison-Based Conversational Recommender System with Relative Bandit Feedback". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '21. Virtual Event, Canada: Association for Computing Machinery, 2021, 1400–1409.

References II

Zhang, Xiaoying et al. "Conversational Contextual Bandit: Algorithm and Application". In: *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. ACM / IW3C2, 2020, pp. 662–672.