Received 2 March 2021
Revised 6 May 2021
Accepted 24 May 2021

An embedded bandit algorithm based on agent evolution for cold-start problem

Rui Qiu
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China and University of Chinese Academy of Science, Beijing, China, and
Wen Ji
Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

Abstract

Purpose – Many recommender systems are generally unable to provide accurate recommendations to users with limited interaction history, which is known as the cold-start problem. This issue can be resolved by trivial approaches that select random items or the most popular one to recommend to the new users. However, these methods perform poorly in many cases. This paper aims to explore the problem that how to make accurate recommendations for the new users in cold-start scenarios.

Design/methodology/approach – In this paper, the authors propose embedded-bandit method, inspired by Word2Vec technique and contextual bandit algorithm. The authors describe user contextual information with item embedding features constructed by Word2Vec. In addition, based on the intelligence measurement model in Crowd Science, the authors propose a new evaluation method to measure the utility of recommendations.

Findings – The authors introduce Word2Vec technique for constructing user contextual features, which improved the accuracy of recommendations compared to traditional multi-armed bandit problem. Apart from this, using this study's intelligence measurement model, the utility also outperforms.

Practical implications – Improving the accuracy of recommendations during the cold-start phase can greatly raise user stickiness and increase user favorability, which in turn contributes to the commercialization of the app.

Originality/value – The algorithm proposed in this paper reflects that user contextual features can be represented by clicked items embedding vector.

Keywords Cold-start, Multi-armed bandit, Word2Vec, Intelligence evolution

Paper type Research paper

1. Introduction

In the era of information explosion, recommender system has become an essential part of internet applications. It plays an important role of filtering information, selecting the information that users prefer to view from a large volume of rich media. For example, items

This work is supported by the National Key R&D Program of China (2017YFB1400100), the National Natural Science Foundation of China (61572466) and the Beijing Natural Science Foundation (4202072).

International Journal of Crowd Science
Vol. 5 No. 3, 2021
pp. 228-238
Emerald Publishing Limited
2398-7294
DOI 10.1108/IJCS-03-2021-0005

are Web page for search engine, videos for video site and articles for content publishing. Recommender system often makes recommendations based on user and item features. These features can be information over the items to recommend (the items-based approach) or to find users with similar tastes (the user-based approach) (Nguyen *et al.*, 2014).

Whether it is the content-based approach or user-based approach, it will face the problem of cold start. For the content-based approach, considering there is a new item on the platform, the item does not have any interaction with the user, such as explicit features (e.g. clicks and browsing) or implicit features (e.g. likes, comments and rates). If the item does not have enough exposure, it will lead to a lack of interaction between user and item, thereby, further reducing the amount of display, falling into a vicious circle. A simple solution is that we can put a corresponding tag when user uploads the content, which is conductive to recommender system to recommend the matched item to interested users. This is the first kind of cold-start problem.

For the user-based approach, when a new user – without any side information – is introduced to the system, we need to collect some data to build a good enough model before being able to produce any valuable recommendation. This is the second kind of cold-start problem. Normally, in the initial stage, new users have limited interaction with recommender system. When sufficient user features are not collected, the common solution is to recommend popular products to the user or fill in the field of interest when user registers. Continuously recommending popular content often brings short-term benefits, but it has trouble in mining user interests and gets stuck into bringing long-term benefits. Therefore, recommender system conduct certain explorations, that is, try to recommend different contents to users, and dynamically adjust recommendation strategies based on user feedback. That is explore/exploit schemes, which is an effective strategy to solve the cold-start problem. In this paper, we propose a new hybrid algorithm based on Word2Vec technique and contextual bandit algorithm. We construct user contextual information by embedding feature of items. The main contributions of this article can be summarized as follows:

- We cast the cold-start problem of recommender systems into explore/exploit problems, and introduce embedded-contextual information constructed by Word2Vec technique. We also consider the similarity between users with K-Nearest Neighbor (KNN) when calculating average reward.
- We originally regard the recommender system as an intelligence agent, and regard the cold-start phase as the evolution process of the agent. Then, we propose a new evaluation method to measure the utility of recommendations based on the intelligence measurement model in Crowd Science, which significantly compares the intelligence of different algorithms for the cold-start phase.

The rest of paper is organized as follows. Section 2 provides a brief overview of the related work. In Section 3, the proposed method is described in detail. In Section 4, the experiment results are provided. Finally, Section 5 concludes the paper.

2. Related work

The cold-start problem was already regarded as of the emergence of recommender system (Schein *et al.*, 2002). For a long time, several methods were proposed for these problems. However, these methods rely heavily on the auxiliary information available between user and item, and this information is not always available. Therefore, it is very difficult to build an accurate recommendation system in practice. For an instance, Lashkari *et al.* (1994) proposed an interview process for users to collect more information before recommendations. And, lots

of works have been conducted to improve the estimation speed of the parameters for new items or new users by using hierarchy of items or various side information (Agarwal *et al.*, 2009a; Yue *et al.*, 2012).

Another common strategy to mitigate the cold-start problem is to leverage exploration–exploitation (EE) dilemma. EE dilemma tends to be studied with so-called multi-armed bandit (MAB) tasks, such as the Iowa gambling task (Bechara *et al.*, 2005; Steyvers *et al.*, 2009). These are tasks in which people are faced with a number of options, each having an associated average reward (Schulz *et al.*, 2018). There are already many algorithm proposed for MAB problem. ϵ-greedy (Auer *et al.*, 2002) algorithm chooses one optimal item with a constant probability ϵ and pick it up uniformly at random with probability $1 - \epsilon$. Upper Confidence Bound (UCB) (Auer *et al.*, 2002) algorithm keeps a track of the mean reward for each arm up to the present trial and also calculates the upper confidence bound for each arm. The upper bound indicates the uncertainty in our evaluation of the potential of the arm. However, for recommender system, the situation is not so simple that recommendation is often accompanied with contextual features, which yield contextual MAB (Li *et al.*, 2010). While for LinUCB algorithm, the author did not consider how to construct features with better generalization capabilities, and also the synergy between users. To enhance the adaptability of recommender system, there are lots of works which focus on how to combine cluster technique with bandit algorithm. In Gentile (2014), each user is treated as a node, and the complete graph is constructed by connecting two edges between users at the initial stage. Nguyen and Lauw (2014) construct user clusters dynamically based on *K*-means. Gentile (2017) implements the underlying feedback sharing mechanism by estimating the neighborhood of users in a context-dependent manner.

In this paper, we consider this problem from a different perspective. We pay more attention on how to easily construct user contextual features with more generalization ability for cold-start problem. And, we propose a new evaluation method to measure the utility of recommendations from the perspective of Crowd Science.

3. Proposed method

In this section, we first formulate the cold-start problem, mainly introduce the classic MAB algorithm and content-based MAB algorithm, then we give brief introduction on Word2Vec technique, and describe how to leverage this technique to construct user contextual features in detail. Finally, we give a metric of the intelligence of recommender system in the cold-start phase from the perspective of Crowd Science.

3.1 Classic multi-armed bandit algorithm

The basic framework of classic MAB algorithm can be formulated as follows. Suppose the items in recommendation system are expressed as $A = \{a_1, \ldots, a_n\}$, where $n \in \mathbb{N}_+$. N is the number of candidate items. a_n is the nth item, where $1 \leq n \leq N$. We set $\mu_{k,1}, \ldots, \mu_{k,n}$ as the determinant average reward corresponding to the kth turn for each selected item $a_1, \ldots, a_n$. So in trial k:

- According to the known average reward value $\mu_{k,n}$ corresponding to each item a_n in the recommendation pool, directly calculate the user's expected reward value for the item. In general, select the known average return expected return value of the item, that is:

$$r'_{k,n} = f(k,n) = \mu_{k,n} = \frac{\sum_{n=1}^{k-1} \hat{r}_{k,n}}{k-1} \tag{1}$$

- Choose the item with highest expected reward a_{n^*} to recommend it to user, and get the true reward $\hat{r}_{k,n^*}$, n^* meet the following conditions:

$$n^* = argmax\left(r'_{k,1}, \ldots, r'_{k,n}\right) \tag{2}$$

- Based on the latest reward $r'_{k,n}$, we update the item a_{n^*}'s determinant average reward:

$$\mu_{k,n^*} = \frac{(k-1)\mu_{k,n^*} + \hat{r}_{k,n^*}}{k} \tag{3}$$

After K turns recommendation, the total reward is $\sum_{k=1}^{K} \hat{r}_k$. The final objective of MAB is to maximize the total reward of recommendation system. In addition, we use regret value $R(n)$ to represent the difference between the optimal total reward and the truth reward after K turn recommendation:

$$R(K) = \sum_{k=1}^{K} r_{k,N^*} - \sum_{k=1}^{K} \hat{r}_k, \tag{4}$$

where $N^* = argmax\{\mu_N(a_N)\}$, represent the optimal recommendation result. The more total reward and the less total regret value, user obtain the better recommendation. Another measurement of recommendation system is click-through time, we will use a function of these three factors to measure the intelligence quantity, and this part will be introduced later.

3.2 Contextual-based multi-armed bandit algorithm

The contextual-based MAB algorithm use the features of users and items to model feedback, then obtain better recommendation results. The process of content-based MAB algorithm is described as follows: we set the total items as $A = \{a_1, \ldots, a_n\}$, where $n \in \mathbb{N}_+$. N is the number of candidate items. a_n is the nth item, where $1 \leq n \leq N$. So in trial k:

- We first get the features of users and items C_k, and calculate the reward by decision function f with current features C_k:

$$r'_{k,n} = f(C_k) \tag{5}$$

- Choose the item with highest expected reward a_{n^*} to recommend it to user, and get the true reward $\hat{r}_{k,n^*}$, n^* meet the formulation 2.
- According to the latest feedback information $\left(C_k, r'_{k,n}\right)$, update the decision function.

Contextual-based MAB algorithm only needs to concentrate on how to construct features of users and items. However, effective feature construction can greatly improve the expressive ability of recommender system. In the next section, we introduce how to obtain user contextual information which inspired by Word2Vec (Mikolov *et al.*, 2013).

3.3 Embedded-bandit algorithm

Recommender system contains explicit interactions or implicit interactions between user and item. For example, user directly scoring a product is an explicit interaction, while user's viewing time on the product, like or comment, is an implicit interaction. The display interaction can directly express the user's preference for the product. For example, if user has a high rating for the movie, it expresses the user's like for this movie, while the implicit interaction often cannot directly derive user preference. Therefore, a way to extract user features through implicit interaction is needed. In the rest of this section, we first provide a brief overview of Word2Vec technique, and then we introduce our proposed method that adapts user embedding in contextual-MAB problem.

3.3.1 Brief overview of Word2Vec. Word2Vec (Mikolov *et al.*, 2013) contains two types of models, skip-gram and cbow. These two models aim at finding words low dimension representation that extract the co-occurrence between a word to its surrounding words in a sentence. In our proposed method, we use skip-gram model, which is more effective compared to cbow. The reason is, in skip-gram, each word is influenced by the surrounding words, and each word is predicted and adjusted k times when it is used as the central word. Therefore, when the amount of data or the number of occurrences of the word is small, such multiple adjustments produce more accurate word embedding.

Given a sequence of words $(w_i)_i^K$ from a finite vocabulary $W = \{w_i\}_i^W$, skip-gram algorithm aims at maximizing the following term:

$$\frac{1}{K}\sum_{i}^{K}\sum_{-c\le j\le c, j\neq 0} \log P\left(w_{i+j}|w_i\right) \tag{6}$$

where c is the context window size and $P\left(w_j|w_i\right)$ is the softmax function:

$$P\left(w_j|w_i\right) = \frac{exp\left(u_i^T v_j\right)}{\sum_{k\in I_w} exp\left(u_i^T v_k\right)} \tag{7}$$

where $u_i \in U(\subset \mathbb{R}^m)$ and $v_i \in V(\subset \mathbb{R}^m)$ are hidden vectors that correspond to the target and context representations for the word w_i, respectively, $I_w \stackrel{\Delta}{=} \{i, \ldots, |W|\}$ and the parameter m is chosen empirically and according to the size of data set. Different setting of m will affect the efficiency of KNN algorithm which is used in our proposed method, and we will compare the impact of different parameters on performance through experiments.

It is impractical by using equation (7) because of the computation cost of $\nabla P\left(w_j|p_i\right)$, which is a linear function of the vocabulary size $|W|$ that is usually in the size of $10^5 - 10^6$.

Negative sampling can alleviate the above problem by replacing the softmax function in equation (7) with:

$$P\left(w_j|w_i\right) = \theta\left(u_i^T v_j\right)\prod_{k=1}^{N}\theta\left(-u_i^T v_j\right) \tag{8}$$

where $\theta(x) = 1/1 + exp(-1)$ and N is a parameter that determines the number of negative examples to be drawn per a positive example. A negative word w_i sampled from the

unigram distribution raised to the 3/4th power. This distribution was found to significantly outperform the unigram distribution, empirically (Mikolov *et al.*, 2013).

3.3.2 Proposed embedded-bandit method. We proposed to adapt skip-gram model with negative sampling introduced in above section to embedded-bandit algorithm. It is really straightforward to apply skip-gram model to cold-start scene once we notice that a sequence of words is equivalent to a collection of items. A set of items comes from user behavior, for example, movies viewed or rated by users, and we sort these items by date or just shuffle items, which is equivalent to data augmentation. Sorting items by date can extract the changing of user interest, while in cold-start scene, there is not much interaction between the user and the recommendation system. We cannot provide enough contextual information, so we can only train the skip-gram model through the historical data of other users.

During cold-start phase, the recommendation system lacks user interaction information. Therefore, the recommend items may bring two results, one is that user's preference items, although there are no user contextual feature, through some exploration methods, it will hit user's interest and bring certain benefits, and the other is bringing new information to the recommendation system, and this kind of recommendation will bring long-term benefits. This is the usual dilemma between exploitation (of already available knowledge) vs exploration (of uncertainty), encountered in sequential decision-making under uncertainty problems (Nguyen *et al.*, 2014).

Our method based on LinUCB algorithm, which models personalized recommendation of new items as a contextual bandit problem (Li *et al.*, 2010). In the framework of LinUCB algorithm, we consider that the features of similar user have a synergistic effect when updating reward by equation (5). Specially, we adapt equation (5) to:

$$r'_{k,n} = f\left(\frac{1}{T}\sum_{t} C_t\right) \tag{9}$$

where T is parameter of KNN and C_t is all similar user embedding feature of the current user. Each C_t is calculated by the average of all positive items embedding vector. One problem is that if a new user appears, there are no positive items clicked, thus we simply average all items embedding vector to represent user contextual information. We use the average of all t's nearest neighbor features as final input feature. The process of proposed method can be described as: assume the set of items is $A = \{a_1, \ldots, a_n\}$, where $n \in \mathbb{N}_+$. N is the number of candidate items. a_n is the nth item, where $1 \leq n \leq N$; $f_1, \ldots, f_n$ represents the decision function of $a_1, \ldots, a_n$; U_i represents the ith user u_i's feature vector, where $i \in \mathbb{N}_+$. In trial k:

- Suppose system recommends items for user u_i, calculates the similarity with other user and chooses the T highest similarity users as nearest neighbor collection N_{u_i}. The similarity is calculated by classical Cosine similarity, and the similarity between u_i and u_j is computed as:

$$Sim_{u_i,u_j} = \cos(U_i, U_j) = \frac{U_i \times U_j^T}{|U_i| \times |U_j|}. \tag{10}$$

- For $\forall u_j \in N_{u_i}$, $\forall a_n \in A$, calculate the average user features and predict the expected reward by equation (9):

$$r_{u_i,a_n} = f_n\left(\frac{1}{|N_{u_i}|}\sum_{u_j \in N_{u_i}} U_j\right) \tag{11}$$

- Choose the item with highest expected reward a_{n^*} to recommend it to user *ui*, and get the true reward $\hat{r}_{u_i,a_n^*}$, n^* meet equation (2). If reward is positive, add this item to user's like item list, then update user feature by average of all item embedding vectors in this list.
- According to the latest feedback information $(U_i, \hat{r}_{u_i,a_n^*})$, update the decision function.

The proposed method will use two equations in LinUCB algorithm:

$$\begin{aligned} \hat{\Theta} &= A_a^{-1} b_a \\ f_a(C) &= \hat{\Theta}_a^T C + \sqrt{C^T A_a^{-1} C}, \end{aligned} \tag{12}$$

where Θ_a represents the expected weight vector of the item a, $f_a(C)$ represents the expected reward of the item a, C is the current contextual content, A_a indicates the accumulated input content of item a, while b_a is the correlation coefficient of A_a with respect to Θ_a:

Algorithm 1. Embedding-bandit for cold-start problem

procedure EMBEDDED-BANDIT (α, k, d, U, I, E)
  Initialization, for $\forall a \in I$, $A_a \leftarrow I_d$, $b_a \leftarrow O_{d\times 1}$
  for t in 1 : T **do**
    If user i is new: $U_i \leftarrow \frac{1}{|V|}\sum_{v\in V} E_v$, where E_v is embedding vector of item v, $|V|$ is the number of item.
    Input user embedding feature as contextual: $C \leftarrow U_i$
    $\forall v \in U$, compute the similarity between user u and v: $distance_{u,v} \leftarrow Sim(u,v)$
    $N \leftarrow$ top k the highest similarity user collection
    $\forall a \in I$, calculate the average user feature: $C \leftarrow \frac{1}{|N_{u_i}|}\sum_{u_j \in N_{u_i}} U_j$
    $\forall a \in I, \forall v \in N$, compute the expected reward: $r_a \leftarrow f_a(C)$
    Recommend item: $a^* \leftarrow argmax\{r_a\}$
    $\hat{r} \leftarrow$ the true reward of user feedback
    If reward is positive, update U_i: $U_i \leftarrow \frac{1}{|List|}\sum_{v\in List} E_v$, where List is the positive item collection of user i.
    Update the decision function of item a*: $A_{a^*} \leftarrow A_{a^*} + CC^T$, $b_{a^*} \leftarrow b_{a^*} + \hat{r}C$
  end for
end procedure

3.4 Intelligent measurement of recommender system

We use embedded-bandit method to solve the cold start problem, and effectively balance exploration and exploitation. As mentioned above, we regard the recommender system as an agent and the cold-start phase as the evolution of the agent. To describe the evolution of agents, we should first define how to measure the intelligence quantity. During the cold start process, we have the following evaluation indicators, average regret rate, click-through rate and total system reward. The larger the click-through rate and total system revenue, the

better, and the smaller the average regret rate, the better. Before giving the measurement function, we first define the calculation methods of the above three indicators:

$$\begin{aligned} regret &= \frac{1}{T}\sum_{t=1}^{T} r_t^* - \sum_{t=1}^{T} \hat{r}_t, \\ reward &= \sum_{t=1}^{T} \hat{t}_t, \\ CTR &= \frac{T^*}{T}, \end{aligned} \tag{13}$$

where T represents the total amount of recommendation and T^* represents the amount of successful recommendation, $\hat{r}_t$ represents the real reward of the tth recommendation and r_t^* represents the optimal recommendation. In Yang (2021), the intelligence measurement of end intelligence is defined as follows:

$$I_{end} = \frac{Q}{\mathcal{T}} \tag{14}$$

Where Q represents the comprehensive evaluation of the performance of agent in the task, and $\mathcal{T}$ represents the time consumed by the agent to complete the task [we use $\mathcal{T}$ to distinguish the interaction rounds T in equation (13)]. Inspired by this, we replace Q with $CTR \times \frac{reward}{regret}$, which represents the performance of recommender system, and replace $\mathcal{T}$ with the interaction rounds T. Then the measurement function comes to:

$$Q(ctr, reward, regret, T) = \frac{1}{T} \times CTR \times \frac{reward}{regret} \tag{15}$$

We claim that under this definition, the intelligence quantity will continue to increase with the optimization of the indicator. We will prove the validity of the method in the experiments.

4. Experiments

In this section, we conduct sufficient experiments to prove the effectiveness of the proposed method. First, we introduce data set used in experiments in detail. Then, we describe the experimental setting, which provides the implementation of the proposed algorithm and other methods to compare with it. Finally, we analyze and discuss about the experimental results.

4.1 Data sets and experiment setting

In the experiment, we use publicly available data sets, namely, MovieLens (Harper and Konstan, 2015) from GroupLens. Some details shown in Table 1. To get enough item sequence,

Table 1. Number of users, items and ratings in data sets

Data sets	No. of users	No. of items	No. of ratings
Movielens-latest-small	668	10,329	105,339
Movielens-1M	6,040	3,900	1,000,209
Movielens-latest	247,753	34,208	22,884,337

we use three data sets, Movielens-latest-small, Movielens-latest and Movielens-1M. The data set contains movie information, user rating score and user tag information. Among them, ratings are made on a five-star scale, with half-star increments (0.5–5.0 stars). As is customary in the recommendation world, we change the rating from a scale of 1–5 to binary value as follows:

- 1 if the rating is 4 or larger = positive item; and
- −1 if the rating is smaller than 4 = negative rating.

We use Movielens-latest and Movielens-1M as training data sets and Movielens-latest-small as test data sets. To overcome Out-Of-Vocabulary (OOV) problem, we set item with no rating to token "(unk)." During training Word2Vec, we set window size from 3 to 5, and for other parameters, we use default value. To improve the confidence of the experiment, we conduct 100, 500 and 1,000 interactions, respectively. The final result is the average of these three experiments.

4.2 Results and analysis

In the experiment, we compared three algorithms. Comparing ϵ-Greedy and UCB algorithm without contextual information to demonstrate that using contextual feature can capture more information between user and item, while comparing our based-method LinUCB to demonstrate that using embedded-bandit algorithm and considering user similarity increase total rewards and the intelligence quantity. The experiment results are show below.

As shown in Tables 2–4, our proposed method embedded-bandit surpasses the comparison algorithm in three indicators, The best score is in italics. For cumulative regrets, our methods got the lowest value in three different data sets. For average rewards, our methods got the best rewards in three different data sets. For the intelligence quantity, our

Table 2. Cumulative regrets for recommender system

Algorithms	Movielens-latest-small	Movielens-latest	Movielens-1M
ϵ-Greedy	522.34	1344.57	721.23
UCB	554.8	1567.26	820.1
LinUCB	508.56	1,290	711
Embedded-bandit	*500.4*	*1158.79*	*696.78*

Table 3. Max average rewards for recommender system

Algorithms	Movielens-latest-small	Movielens-latest	Movielens-1M
ϵ-Greedy	0.854	0.835	0.843
UCB	0.837	0.830	0.837
LinUCB	0.855	0.841	0.852
Embedded-bandit	*0.860*	*0.847*	*0.875*

Table 4. The intelligence quantity for recommender system

Algorithms	Movielens-latest-small	Movielens-latest	Movielens-1M
ϵ-Greedy	94.55	95.56	96.32
UCB	93.73	94.37	95.34
LinUCB	94.01	95.21	96.54
Embedded-bandit	*95.28*	*96.28*	*96.98*

methods got the highest value under our proposed intelligence quantity (IQ) measurement method.

5. Conclusion and future work

In this paper, we proposed user embedding based methods called embedded-bandit to solve the cold-start problem. Based on embedding vector, our method has better generalization ability, and with the consideration on user similarity, we use top k similar user as a same interest user group, and calculate the expected rewards by this group, which proved increase recommender system average rewards compared with three algorithms (ϵ-Greedy, UCB and LinUCB). However, our proposed method also meet some problems, such as the dimension of embedding is a hyper-parameter and hard to choose, and calculating the whole user similarity is computational cost. In the future, we will focus on the above problems, and attempt to use neural network method in the cold-start problem.

References

Agarwal, D., Chen, B.-C. and Elango, P. (2009a), "Spatio-temporal models for estimating click-through rate", *in Proceedings of the 18th International Conference on World Wide Web*, pp. 21-30.

Agarwal, D., Chen, B.-C., Elango, P., Motgi, N., Park, S.-T., Ramakrishnan, R., Roy, S. and Zachariah, J. (2009b), "Online models for content optimization", *Advances in Neural Information Processing Systems*, pp. 17-24.

Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002), "Finite-time analysis of the multiarmed bandit problem", *Machine Learning*, Vol. 47 Nos 2/3, pp. 235-256.

Bechara, A., Damasio, H., Tranel, D. and Damasio, A.R. (2005), "The Iowa gambling task and the somatic marker hypothesis: some questions and answers", *Trends in Cognitive Sciences*, Vol. 9 No. 4, pp. 159-162.

Gentile, C., Li, S. and Zappella, G. (2014), "Online clustering of bandits", *in International Conference on Machine Learning*. PMLR, pp. 757-765.

Gentile, C., Li, S., Kar, P., Karatzoglou, A., Zappella, G. and Etrue, E. (2017), "On context-dependent clustering of bandits", *in International Conference on Machine Learning*, PMLR, pp. 1253-1262.

Harper, F.M. and Konstan, J.A. (2015), "The Movielens datasets: history and context", *Acm Transactions on Interactive Intelligent Systems (TIIS)*, Vol. 5 No. 4, pp. 1-19.

Lashkari, Y., Metral, M. and Maes, P. (1994), "Collaborative interface agents", *In AAAI*, Vol. 94, pp. 444-449.

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*, pp. 661-670.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013), "Efficient estimation of word representations in vector space", arXiv preprint arXiv:1301.3781.

Nguyen, T.T. and Lauw, H.W. (2014), "Dynamic clustering of contextual multi-armed bandits", *in Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 1959-1962.

Nguyen, H.T., Mary, J. and Preux, P. (2014), "Cold-start problems in recommendation systems via contextual-bandit algorithms".

Schein, A.I., Popescul, A., Ungar, L.H. and Pennock, D.M. (2002) "Methods and metrics for cold-start recommendations", *in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 253-260.

Schulz, E., Konstantinidis, E. and Speekenbrink, M. (2018), "Putting bandits into context: how function learning supports decision making", *Journal of Experimental Psychology. Learning, Memory, and Cognition*, Vol. 44 No. 6, p. 927.

Steyvers, M., Lee, M.D. and Wagenmakers, E.-J. (2009), "A Bayesian analysis of human decision-making on bandit problems", *Journal of Mathematical Psychology*, Vol. 53 No. 3, pp. 168-179.

Yang, Z., Liang, B. and Ji, W. (2021), "An intelligent end-edge-cloud architecture for visual IOT assisted healthcare systems", *IEEE Internet of Things Journal*.

Yue, Y., Hong, S.A. and Guestrin, C. (2012), "Hierarchical exploration for accelerating contextual bandits".

Corresponding author
Wen Ji can be contacted at: jiwen@ict.ac.cn