Online Learning-based Clustering Approach for News Recommendation Systems

Minh N. H. Nguyen, Chuan Pham, Jaehyeok Son, and Choong Seon Hong,

Department of Computer Science and Engineering, Kyung Hee University, Korea,
Email: {minhnh, pchuan, sonjaehyeok, cshong}@khu.ac.kr.

***Abstract*—Recommender agents are widely used in online markets, social networks and search engines. The recent online news recommendation systems such as Google News and Yahoo! News produce real-time decisions for ranking and displaying highlighted stories from massive news and users access per day. The more relevant highlighted items are suggested to users, the more interesting and better feedback from users achieve. Therefore, the distributed online learning can be a promising approach that provides learning ability for recommender agents based on side information under dynamic environment in large scale scenarios. In this work, we propose a distributed algorithm that is integrated online K-Means user contexts clustering with online learning mechanisms for selecting a highlighted news. Our proposed algorithm for online clustering with lower bound confident clustering approximates closer to offline K-Means clusters than greedy clustering and gives better performance in learning process. The algorithm provides a scalability, cheap storage and computation cost approach for large scale news recommendation systems.**

I. Introduction

In the current research trends about intelligent systems, recommendation system is one of the hottest applications that can provide useful information and advantages for online users. In addition to the traditional accuracy issue, the recent recommendation systems also concern about dynamic environment, the large number of online users and new contents. The recent support from online learning theoretical analysis initiates a new promising adaptive approach for dealing with unknown environments. Using online learning techniques, the intelligent agents are able to make decisions from current and past observations. For example, Yahoo! Today module website [1], the prior version of Yahoo! News, is considered as the typical online learning application. In this system, one news is showed as a highlighted story from the news pool. Since the related or hot news have higher the click through rate than others, a news recommender system takes an important role to interact with user and display proper highlighted news. Consequently, variant online algorithms are applied for Yahoo! Today dataset [1] to achieve as high as possible the click through rate from suggested highlighted stories.

In the online learning approach, the general decision making framework Multi-Armed Bandits (MAB) is well-studied and analyzed in several domains, such as recommendation systems, Ad placement, online routing and computer game-playing [2]. MAB framework has variant settings depends on nature of environment such as stochastic, adversarial, and Markovian. Proposed algorithms for MAB framework are suitable for intelligent recommender agents by suggesting better items or products for users over the time. Specifically, a news recommender system has a lot of users and news are updated by time and the recommendation agent has to adapt with frequent changes. Its goal is improving user click through rate and user attention from displaying relevant news in pools from side information of user contexts.

In this paper, our proposal based on the assumption that users have the same interests if their contexts are similar. According to this assumption, we use an clustering technique to cluster user contexts. Leverage the original offline K-Means clustering technique [3], we propose two different strategies for online K-Means clustering. After grouping user contexts, each cluster has its knowledge and can maintain an online learning process. The advantage of separate learning produces distributed online manner when similar users share their learning feedback. The combined online approach has cheap computation, storage, scalability and suitable for the dynamic environment like user clicks.

II. Related Work & Background

The original recommender systems based on content-based filtering approach and collaborative filtering. These recommendation approaches require high computational cost and cannot be efficiently used for dynamic updating environment in news systems and handling *cold-start* problem. [4].

Multi-Armed Bandits framework provides the analysis framework of decision making with limited information from environment feedback [2]. Initially, the player does not have any knowledge from environment and requires **exploration** and **exploitation** process to perform. Exploration process helps the player to discover new knowledge and get experience from environment. While in exploitation phase, the player decides the best action uses the past and current knowledge. Online learning algorithms for MAB consider the trade-off between exploration and exploitation for adapting with a dynamic environment without loosing intermediate rewards.

”This research was supported by the MSIP, Korea, under the G-ITRC support program (IITP-2016-R6812-15-0001) supervised by the IITP.” *Dr. CS Hong is the corresponding author.

Two widely used context-free MAB algorithms are ϵ-greedy [5] and UCB1 [6].

III. Online Learning-based K-Means Clustering

A. Contextual Multi-Armed Bandits Formulation

In this paper, we use contextual MAB model with stochastic reward process [2] for dealing with the unknown randomized user feedback distribution. We define set $\mathcal{A}$ is the set of news, set $\mathcal{X}$ is the set of users context in the news recommendation system. The recommender agent will learn from user feedback and make suggestions through T discrete trials:

- Agent receives user context feature vector x_t from context space $\mathcal{X}$. Based on context x_t, recommender agent uses an algorithm $\mathbb{A}$ to select a news a_t for displaying.
- Agent observes user feedback r_{t,a_t} which is clicked or not. The reward r_{t,a_t} are followed unknown distribution.
- Agent updates its strategy and parameter model to improve for the next suggestion a_{t+1}.

Expected total regret [1] of contextual MAB setting after T trials measures the different between the best fixed total expected reward with total reward of algorithm $\mathbb{A}$:

$$R_{\mathbb{A}}(T) = E\Big[\sum_{t=1}^{T} r_{t,a_t^*}\Big] - E\Big[\sum_{t=1}^{T} r_{t,a_t}\Big]$$

The expected regret is a function of T and usually used for performance analysis of the algorithm $\mathbb{A}$. The goal of the recommender agent is playing and updating strategy to minimize the expected total regret from decisions. From theoretical analysis [6], context-free MAB algorithms, such as greedy and ϵ-greedy have total regret bounded by linear function of T, while decaying ϵ_t-greedy and UCB1 algorithm achieve total regret bounded by sublinear function of T. These algorithms are widely used in many applications of context-free MAB framework considering exploration and exploitation ability. Besides, LinUCB [1] for contextual MAB uses different binary classification models for user context. Under Bayesian probabilistic framework [1], weight parameters of models are updated by Gaussian posterior distribution. These works heavily based on the specific assumption of linear models between incoming context and expected user feedback. The more complicated fitting model is, the higher computation cost updating model requires over the time. Then they have problems with scalable issues. Different from the recent distributed online learning based clustering approaches [7] that depend on the metric space assumptions, we leverage K-Means clustering [3] using Euclidean distance between feature vectors.

B. Algorithms

Firstly, we introduce a simple scenario in order to demonstrate our proposal stages, as shown in Fig. 1. In this scenario, we have 4 context clusters associated with 4 news. The arrows from clusters to news illustrate users in these clusters prefer one news to others, such as users in C_2 are interested in news B than A, C and D. At time t, the agent receives the request context x_t. Context space $\mathcal{X}$ has 4 existing cluster representatives C_1, C_2, C_3 and C_4 constructed from the past events. Based on the distance between context x_t and cluster centers, cluster C_2 is selected. Then context x_t is assigned to cluster C_2 and cluster representative C_2 moves to the new center C_2'. Then only cluster C_2 executes the exploration-exploitation process to decide which news to display for the user. In this scenario, users in cluster C_2 prefer news B to others news, then news B is selected to display.

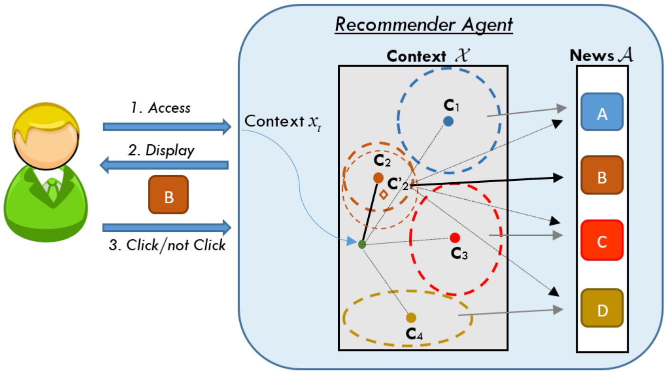


Fig. 1: OL-KMC learning with 4 context clusters and 4 news.

Algorithm 1 Online Learning K-Means Clustering (OL-KMC)

```
Initialization: Create news list for each cluster $\mathcal{A}_{C_k}$
    Set K initial cluster representatives $C_k$
Parameters: $\gamma \geq 0$
for $x_t \in \mathcal{X}$ do
    Greedy / LCB cluster searching:
        $C_k^* = \operatorname{argmin}_{C_k \in C_K} \|x_t - C_k\| - \gamma\sqrt{\frac{2\ln t}{n_{t,k}}}$
        Assign $x_t$ to $C_k^*$
    Update centroid:
        $C_k^* = \frac{x_t + n_k C_k^*}{n_k + 1}$
        $n_k = n_k + 1$
    Exploration - Exploitation process:
        $\epsilon_t$ - greedy / UCB1 algorithm
end for
```

As we mentioned in the scenario, algorithm OL-KMC has two stages which are online clustering Alg. 1 and distributed learning Alg. 2 and 3. For **online clustering stage**, we group context space $\mathcal{X}$ by K number of clusters based on the assumption that all users in a cluster have similar interests. In the Yahoo! news recommendation application, context of users can be user demographic information, location and behavior [8]. Each cluster has a tendency to prefer one news to the other news and recommender agents need to learn from dynamic feedback of inside contexts. In online K-Means version, a cluster C_k is summarized by a context representative C_k and the number of contexts n_k belong to it, where the cluster index $k = 1, \ldots, K$. For every incoming context, the agent update and restructure clusters in clustering stage. Since user contexts sequentially arrive at the agent, we do not know the

Algorithm 2 Decaying ϵ_t-greedy strategy [6]

```
1:  Selected cluster $C_k^*$ from OL-KMC
2:  Parameters: $c > 0$ and $0 < d < 1$
3:  $\epsilon_{t,k} = \min\{1, \frac{cK}{d^2 n_{t,k}}\}$
4:  Draw $val = Bernoulli(\epsilon_{t,k})$
5:  if $val = 1$ then
6:      $\mu_{a,k}^t = \frac{r_{a,k}^{1..t}}{n_{a,k}^t} \quad \forall a \in \mathcal{A}_{C_k^*}$
7:      $a_t^* = \operatorname{argmax}_{a \in \mathcal{A}_{C_k^*}} \mu_{a,k}^t$        (Exploitation)
8:  else
9:      Randomly choose news $a_t^*$        (Exploration)
10: end if
11: $n_{a^*,k}^t = n_{a^*,k}^t + 1$
```

Algorithm 3 UCB1 strategy [6]

```
Selected cluster $C_k^*$ from OL-KMC
Parameters: $\alpha > 0$
for news $a \in \mathcal{A}_{C_k^*}$ do
    $\hat{\mu}_{a,k}^t = \frac{r_{a,k}^{1..t}}{n_{a,k}^t} + \alpha\sqrt{\frac{2\ln n_{t,k}}{n_{a,k}^t}}$
end for
$a_t^* = \operatorname{argmax}_{a \in \mathcal{A}_{C_k^*}} \hat{\mu}_{a,k}^t$
$n_{a^*,k}^t = n_{a^*,k}^t + 1$
```

context before it appears. The agent only receives contexts one by one and proceeds reconstruction of clusters. Users in wrong clusters can receive unrelated news suggestion. For cluster searching step, we inherit an uncertain penalty on the Euclidean distance between context x_t and cluster representative C_k from UCB analysis [6], where

$$d_{t,k} = \|x_t - C_k\| - \gamma\sqrt{\frac{2\ln t}{n_{t,k}}}.$$

This is an approximation of lower confident bound (LCB) on independent distance random variables. In this step, the agent looks for cluster C_k^* that has the smallest adjusted distance $d_{t,k}$. When $\gamma = 0$, the cluster selection becomes greedy strategy and the selected cluster is the smallest distance between context x_t and cluster representative. LCB strategy has parameter $\gamma > 0$ and strongly penalize on the distances $d_{t,k}$ of the clusters that have small number of contexts $n_{t,k}$. Instead of optimistic strategy in UCB1 algorithm, LCB clustering use a pessimistic strategy. Intuitively, LCB strategy gives high chances for exploration while greedy strategy always does exploitation in cluster searching stage.

After choosing a cluster, the centroid of that cluster is updated based on the current representative C_k and the number of contexts n_k in cluster. The selected cluster will run **exploration - exploitation process** to learn from user feedback and suggest news a_t^* for users. For this step, we use two well-known online learning algorithms that have total regret bounded by sublinear function of T: decaying ϵ_t-greedy Alg. 2 and UCB1 Alg. 3. From context-free MAB regret analysis [6], ϵ_t-greedy and UCB1 strategy achieve logarithmic asymptotic regret. In decaying ϵ_t-greedy strategy, a Bernoulli random variable are generated with probability ϵ_t for exploration and $1-\epsilon_t$ for exploitation. The decreased probability ϵ_t by time is the control factor for exploitation and exploration associated with constant c, d at line 3 in Alg. 2. In exploitation stage, the agent only selects the current best expected number of clicked news. When recommender agent try different news over the time, it has more knowledge about rewards of news. Then exploration probability decreases and exploitation probability increases by time.

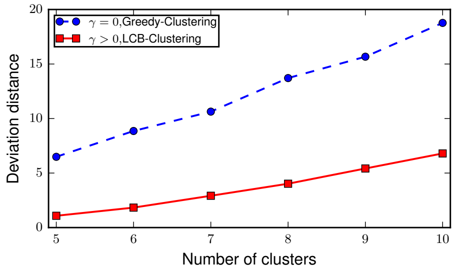


Fig. 2: Greedy and LCB clustering strategy deviation distance

In UCB1 strategy, the agent selects the news that has the best expected reward with an uncertain amount. UCB1 strategy follows optimistic strategy by choosing the best possible clicked news in the future rather than the current one.

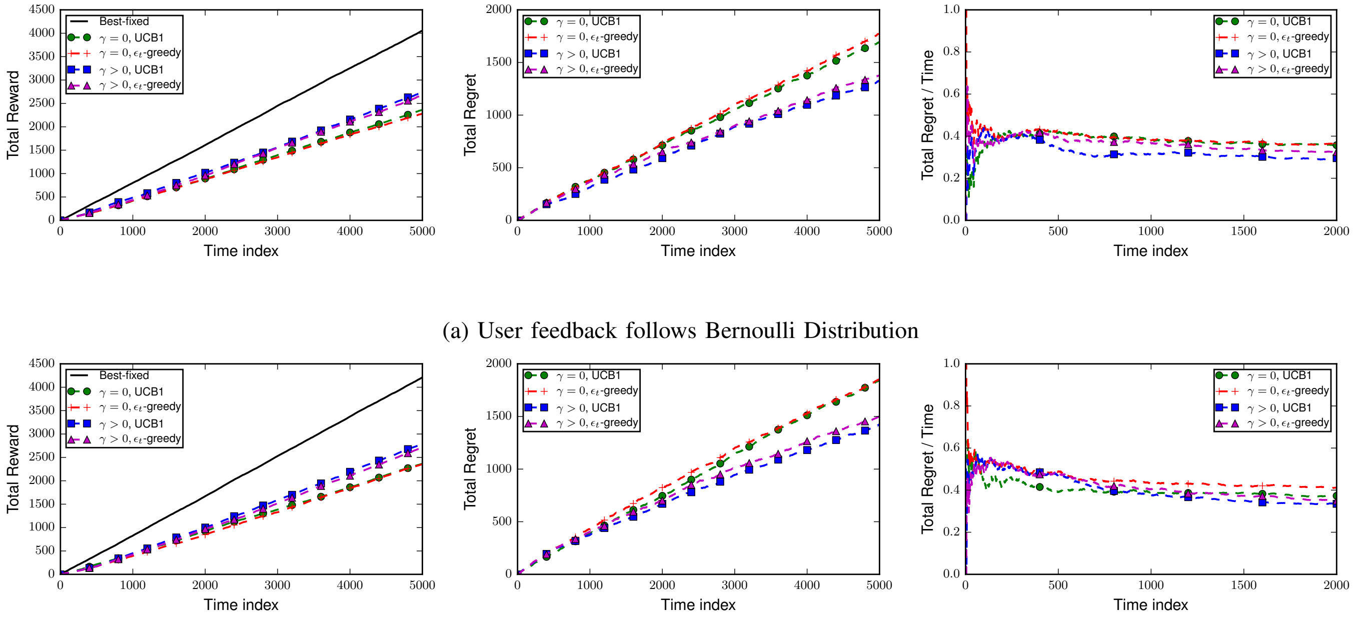
IV. Simulation Results

A. Simulation Environment

For simulation, we use the synthetic dataset following introduced Monte Carlo simulation approach. User contexts are randomly generated with 50 binary features, which have value 0 or 1 for each feature. Initially, we generate 5000 user contexts that sequentially arrive at the agent. Then we use offline K-Means clustering to group similar users into 10 clusters associated with 10 news. In order to demonstrate user cluster interests of one news more than others, the unknown reward distribution are controlled by the higher probability of Bernoulli distribution or the higher mean value of Gaussian distribution for one specific news. User click feedback generated by Bernoulli distribution with probability for the most interesting news of each cluster is 0.8 and the remaining news are 0.3.

B. Results

In Fig. 2, parameter γ are set to 0 for Greedy-Clustering and 1.1 for LCB-Clustering. We compare the deviation distance between offline K-Means with online clustering after 5000 samples from 10 simulations. LCB-Clustering produces closer distance to offline clustering centers than Greedy-Clustering. The average deviation of 10 clusters in LCB-Clustering is 6.8 while Greedy-Clustering is 18.77. When the number of clusters is increased the deviation distance also increase. The increment slop of LCB-Clustering is lower than Greedy-Clustering that can assure a better scalable number of clusters.



(a) User feedback follows Bernoulli Distribution

(b) User feedback follows Gaussian Distribution.

Fig. 3: Learning performance of OL-KMC using Greedy-Clustering ($\gamma = 0$) and LCB-Clustering ($\gamma > 0$).

Moreover, the deviation distance affects to the performance of recommender agent in both case of Gaussian or Bernoulli user feedback distribution. Fig. 3 is one sample of simulation results for OL-KMC where two different clustering strategies are combined with UCB1 and ϵ_t-greedy learning. In the best-fixed simulations, we always choose the news a_t^* which has the best expected reward $E\left[\sum_{t=1}^{T} r_{t,a_t^*}\right]$ for every round. The best-fixed lines are baselines to measure performance of OL-KMC algorithms. LCB-clustering($\gamma > 0$) with higher accuracy from clusters distance achieves better total reward and lower total regret compare to Greedy-Clustering ($\gamma = 0$). After 5000 requests arriving, total reward of LCB-Clustering is greater than Greedy-Clustering around 400 clicks with Bernoulli distribution and nearly 350 clicks with Gaussian distribution. Besides, LCB-Clustering using UCB1 and ϵ_t-greedy learning obtains very close total reward but in most of the simulations, UCB1 produces slightly better reward than ϵ_t-greedy does. The total regret of LCB-Clustering lines in both feedback distributions have lower slop compared to Greedy-Clustering lines. They have a tendency of moving from linear lines to the sublinear curves. Besides, LCB-Clustering algorithms have total regret over time decrease as we expected.

V. Conclusions

In this paper, we introduce the combined online clustering with context-free online learning within OL-KMC algorithms for news recommender systems. The algorithm provides scalability of system with cheap storage and computation cost in each iteration. Instead of recording the whole history of contexts and user feedback, we only need to store summary cluster information and the cumulative number of selected arms. Based on the summarized cluster information, OL-KMC can reconstruct clusters and perform as an adaptive learning algorithm in dynamic environment as user feedback.

References

[1] L. Li, W. Chu, J. Langford, and X. Wang, “Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms,” in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 297–306.

[2] S. Bubeck and N. Cesa-Bianchi, “Regret analysis of stochastic and nonstochastic multi-armed bandit problems,” *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012. [Online]. Available: http://dx.doi.org/10.1561/2200000024

[3] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[4] C. Lin, R. Xie, X. Guan, L. Li, and T. Li, “Personalized news recommendation via implicit social experts,” *Information Sciences*, vol. 254, pp. 1–18, 2014.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.

[6] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[7] L. Song, C. Tekin, and M. van der Schaar, “Clustering based online learning in recommender systems: a bandit approach,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4528–4532.

[8] W. Chu, S.-T. Park, T. Beaupre, N. Motgi, A. Phadke, S. Chakraborty, and J. Zachariah, “A case study of behavior-driven conjoint analysis on yahoo!: Front page today module,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 1097–1104.