Experiment Design for Data Science 2023 - Group 18

DAMIR DIZDAREVIC, 12141497
DANILO TONIC, 12229946
DUSAN SALOVIC, 12226592
MICHAEL HÖCHTL, 11775779

ACM Reference Format:
Damir Dizdarevic, Danilo Tonic, Dusan Salovic, and Michael Höchtl. 2023. Experiment Design for Data Science 2023 - Group 18. 1, 1 (January 2023), 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Shams et al. is the paper which discusses the challenge of the "cold-start task" in recommender systems, where a new user joins the system and the system initially has no knowledge of the user's preferences. The initial approach to this task is to take ratings already collected from a population of users, cluster them into groups, and train a decision-tree to learn a mapping from item ratings to the user group. However, this approach is often inaccurate due to the intra-cluster variability between users, which can be thought of as adding noise to the ratings. An alternative approach is to use Multi-arm bandits (MABs), but standard bandit algorithms also have poor performance on this task. Shams et al. proposed a novel cluster-based bandit algorithm for which they claim that it achieves fast learning in cold-start. They show that the group of a user is identified with significantly higher accuracy than with a decision-tree without incurring higher regret, meaning the learning performance is fundamentally superior to that of a decision-tree. Aim of this paper is to challenge the conclusions of the proposed algorithm by implementing it using the instructions from the paper while investigating if they are enough to recreate all of the algorithm's functionalities.

2 CLUSTER-BASED BANDIT ALGORITHM

The main task is to efficiently determine the group a new user belongs to by having them rate items in a set of items, V. We know that the way items are rated by a user varies depending on the group they belong to, and we have a set of user groups, G, where each user is a member of one group. By analyzing the ratings given by the new user, we aim to identify the group they belong to.

2.1 Group Indicator Vector

We aim to measure an indicator vector, I, for a new user, which represents the group that the user belongs to. The vector I should tend towards 1 for the group the user belongs to and towards 0 for all other groups as more user ratings are collected. To obtain this vector, we calculate

$R_n(g,h) = \sum_{i=1}^{n} \alpha_n^i(g,h) \frac{r(v_i)-\mu(h,v_i)}{\mu(g,v_i)-\mu(h,v_i)}$

Authors' addresses: Damir Dizdarevic, 12141497; Danilo Tonic, 12229946; Dusan Salovic, 12226592; Michael Höchtl, 11775779.

for each group g and h where n is the number of items rated by the new user, $r(v_i)$ is the new user's rating for item v_i, $\mu(g, v_i)$ is the mean rating of item v_i by users in group g and $\alpha_{n,i}(g, h)$ is a weighting such that $\sum_{i=1}^{n} \alpha_{n,i}(g, h) = 1$. An estimate of $\mu(g, v_i)$ is known. We can then calculate

$$I(g) = \min_{h \in G \setminus \{g\}} R_n(g, h)$$

as an indicator vector.
We then estimate the group that the new user belongs to by

$$\hat{g} \in \underset{g \in G}{argmax} I(g).$$

The key to fast learning is that $I(g)$ converges quickly to a (0,1) vector.

2.2 Fast Convergence

When attempting to distinguish between two groups of users, g and h, it is important to consider the properties of the items that are being used to make this distinction. Specifically, an item v can be useful in distinguishing between group g and group h if it satisfies two conditions. Firstly, the mean rating of v by users in group g should be significantly different from the mean rating of v by users in group h, as measured by $(\mu(g, v) - \mu(h, v))^2$. Secondly, the ratings of v should be consistent and reliable, as measured by the variance $\sigma^2(g, v)$.
With this in mind, we can define a measure of the ability of item v to distinguish group g from group h, given by

$$\Gamma_{g,h}(v) = \frac{(\mu_{g,v} - \mu_{h,v})^2}{\sigma_{g,v}^2}$$

The larger the value of $\Gamma_{g,h}(v)$, the better item v is at distinguishing group g from group h. To achieve fast convergence for $R_n(g, h)$, it is therefore desirable to choose a sequence of items v_i, $i = 1, ..., n$ such that

$$\Sigma_n(g, h) = \sum_{i=1}^{n} \Gamma_{g,h}(v_i)$$

is large.
Another way to understand the importance of this measure is to assume that for users belonging to group g, the rating $r(v)$ of item v is i.i.d. subgaussian with mean $\mu(g, v)$ and variance $\sigma^2(g, v)$. Under this assumption, standard concentration inequalities tell us that

$$\text{Prob}(|R_n(1, h) - 1| > \epsilon) < 2e^{-\frac{\epsilon^2}{2}} \sum_{i=1}^{n} \Gamma_{1,h}(v_i)$$
$$\text{Prob}(|R_n(g, 1) - 0| > \epsilon) < 2e^{-\frac{\epsilon^2}{2}} \sum_{i=1}^{n} \Gamma_{g,1}(v_i)$$

when we select $\alpha_i^n(g, h) = \frac{\Gamma_{g,h}(v_i)}{\sum_{i=1}^{n} \Gamma_{g,h}(v_i)}$. Therefore, for $R_n(g, h)$ to converge quickly, it is desirable to have a large value for $\Sigma_{i=1}^{n} \Gamma_{g,h}(v_i)$.

2.3 Exploration vs Exploitation

The discussion above suggests that for efficient learning, it is desirable to initially ask the user to rate items v_i for which $\Gamma_{g,h}(v_i)$ is highest. However, these items may not be those that receive high ratings, and there is a cost to this accelerated learning. Therefore, it is important to limit the duration of the initial exploration phase and quickly switch to an exploitation phase, where items that are predicted to receive high ratings by the new user are recommended. These items have a high value of $\mu(\hat{g}, v)$, where $\hat{g}$ is the estimated group of the new user. It is important to note that learning can still occur during the exploitation phase, as long as the items rated have non-zero $\Gamma_{g,h}(v_i)$, but the learning rate can be much slower than during the exploration phase. Additionally, when deciding when to switch from the initial exploration phase (items with high $\Gamma_{g,h}(v)$) to the subsequent exploitation phase (items with high $\mu(\hat{g}, v)$), it is necessary to balance the need to confirm the new estimate with the need to explore other possibilities. It may happen that a new user's rating for an item is unusually high or low for their group, and therefore, multiple items must be rated to correct for mistakes and gain confidence in the estimate of $\hat{g}$. To accomplish this, an upper confidence bound (UCB) strategy is employed. Specifically, at step n, the next item v to be presented to the user is the unrated item for which the learning rate $\Gamma_{\hat{g},h}(v)$ is highest and h is the group for which $\Sigma_n(\hat{g}, h)$ is lowest. By selecting h in this way, all pairs $(\hat{g}, h)$ of groups are explored in such a way that a similar amount of information, as measured by $\Sigma_n(\hat{g}, h)$, is gained about each pair.

2.4 Proposed algorithm

2.4.1 Algorithm. The exploration phase of the resulting bandit algorithm is outlined in Figure 1. The algorithm includes three design parameters: B, C, and D. In the particular implementation in the paper (Shams et al.), B is set to 5, C is set to 0.5, and D is set to 3. Once the exploration phase is complete, the algorithm enters its exploitation phase. The procedure is the same as the exploration phase, but instead of calling the $Explore(\hat{g})$ function, the algorithm calls the $Exploit(\hat{g})$ function, which selects the unrated item with the highest predicted rating for the user's estimated group $\mu(\hat{g}, v)$.

```
Algorithm 1: Cluster-based Bandit Algorithm
Input: Groups $G$, mean ratings $\mu(g, v)$ for item $v$ and
       variances $\sigma(g, v)^2$; parameters $B, C, D \in \mathbb{R}_+$.
1  Select initial $\hat{g} \in G$, e.g. randomly, and Explore($\hat{g}$)
2  for $n = 1, 2, 3. \dots$ do
3      Define the candidates as
         $M_n = \{g \in G : |\min_h |R_n(g, h)| - 1| \le C)\}$
4      if $M_n \neq \emptyset$ then
5          $\hat{g} \in \arg\max_{g \in G} I(g)$
6          Explore($\hat{g}$)
7          if $\Sigma_n(\hat{g}) \ge B$ or $(|M_n| = 1$ and $n > D \log_2 |G|)$ then
8              Estimate new user belongs to group $\hat{g}$
9              Exit exploration phase for $\hat{g}$
10     else
11         $(\hat{g}, h) \in \text{argmin}_{g,h} \Sigma_n(g, h)$
12         Explore($\hat{g}$)
```

```
Algorithm 2: Explore($\hat{g}$)
1  $V_n = \{v_1, \dots, v_{n-1}\}$ (set of items already rated by user)
2  $h \in \text{argmin}_h \Sigma_n(\hat{g}, h) = \sum_{i=1}^{n} \Gamma_{\hat{g},h}(v_i)$
3  Ask user to rate item $v_n \in \arg\max_{v \notin V_n} \Gamma_{\hat{g},h}(v)$
```

Fig. 1. Cluster-based Bandit Algorithm

2.4.2 Implementation inaccuracies. Algorithm includes divisions while calculating $\Gamma_{g,h}(v)$ and $R_n(g,h)$, which tend to create instabilities while running the code. If the standard deviation of ratings for certain group and item is very low or if the two mean ratings are very close, this can cause failure of program. When deciding whether we should switch to exploitation phase we have Σ_n as part of the condition. Σ_n have two arguments (two groups) and not one as it can be seen in Figure 1. We assumed that $\Sigma_n(\hat{g})$ implies minimal Σ_n value for every group h. Also, the pseudo-code states that if the condition is fulfilled that we estimate that user belongs to certain group, which can be understood as final decision on the group of the user, which is not the case since it is stated that learning is continued in the exploitation phase.

3 PERFORMANCE EVALUATION

3.1 Evaluation Setup

3.1.1 Datasets. In this study, the performance of the cluster-based bandit algorithm for cold start is evaluated on three standard datasets: Netflix (480,189 users, 17,770 movies, and 104M ratings from 1-5), Jester (73,421 users rating 100 jokes, 4.1M ratings from -10 to 10), and Goodreads10K (53,424 users rating 10,000 books, 5.9M ratings from 1-5). These datasets are not publicly available. In order to gather the needed data, data from the official github of the paper is used, although this github is nowhere stated in mentioned paper. The github is created by the author of the paper Douglas Leith and it is available on: https://github.com/doug-leith/SIGIR_clusterbasedbandit. Github provides the implementation in MATLAB while we are attempting to reproduce the results by using Python.

3.1.2 Initial clusterization. The users are clustered into groups using the BLC matrix-factorization clustering algorithm, with the number of groups/clusters varying from 4 to 32. Available data from github was already clusterized so we used data in that format. This helps in evaluation of the proposed algorithm without discussing the possible inaccuracies in clusterization algorithm implementation.

3.1.3 Baseline. The performance of the algorithm is compared against an optimized CART decision tree and the cluster-based bandit algorithm without the initial exploration phase. We are only analyzing the performance of the proposed algorithm and comparing them to the ones that are reported in the paper without implementing the baseline algorithms.

3.1.4 New users. New users' item ratings are generated from the group g by making a single draw from the multivariate Gaussian distribution with mean $\mu(g,v)$ and variance $\sigma^2(g,v)$ for each item. This testing technique is different from classic train/test splits of the data, and the paper reports that these two testing techniques actually produce the same results. Since the ratings are discrete and have certain range, effectively we won't be able to produce Gaussian distribution on the tails of the discrete range in exact way. If the sampled rating is bigger than the maximum rating we truncate it to the maximum one, making it more likely to sample it (analogous for minimum rating). This problem wasn't discussed in the paper.

3.1.5 Metrics. The performance is measured by the accuracy with which the group of a new user is estimated and the regret, which is calculated as the sum of $\sum_{i=1}^{n} r(v_i) - r(v_1^*)$, where v^i is the unrated item with the highest rating by the new user (so v_1^* is the highest rated item, v_2^* is the next highest and so on). These statistics are calculated over 1000 new users per group. By analyzing definition of regret formula, we cannot conclude what v_i^* actually is, since it is stated that it is unrated item with the highest rating, which implies contradiction. If in certain step the item is not yet rated, does that mean that it will be rated later and should we then also generate that rating as every other? Other option, which we used in our implementation, is that $r(v_i^*)$ is actually ith best mean rating across the unrated items in certain step. After using this assumption, since the best mean ratings in the first steps are usually higher that the first corresponding generated ratings, according to regret formula, we get function which decreases. That is not expected behaviour and it doesn't fit to the chart presented in the paper. It also undermines assumptions which were made with the definition of convergence time. In addition, unrated items in every step are different for almost every user which creates time-consuming operation of finding the best unrated items when calculating this metric. For this reasons we decided to only use accuracy to compare our results to the ones stated in the paper.

3.2 Results

In Table 1, the average accuracy measurements for the Netflix, Jester, and Books datasets are presented. The values shown are the average of the groups, with 1000 new users per group. From the paper, it can be observed that the cluster-based bandit algorithm consistently achieves higher accuracy than a decision tree and the cluster-based bandit algorithm without exploration phase. But, that isn't the case with the results we got with the same algorithm.

Dataset/Algo	Accuracy		
	#Groups		
	4	8	16
Netflix/DT	0.93	0.88	0.75
Netflix/CB-	0.83	0.79	0.65
Netflix/CB	**0.99**	**0.96**	**0.91**
Netflix/Our	0.86	0.58	0.34
Jester/DT	0.71	0.55	0.46
Jester/CB-	0.84	0.75	0.60
Jester/CB	**0.91**	**0.83**	**0.73**
Jester/Our	0.19	0.12	0.08
Books/DT	0.88	0.69	0.62
Books/CB-	0.82	0.72	0.60
Books/CB	**0.96**	**0.87**	**0.84**
Books/Our	0.92	0.63	0.39

Table 1. Mean accuracy for Netflix, Jester and Books datasets vs number of groups

In this exercise report, we report significantly worse results on all performed tests. In most cases, our results are worse than results obtained with decision trees. Especially, our results on the Jester dataset are very bad (random classifier would have a higher accuracy), which can be a consequence of the Jester dataset being the only one with a significantly wider range of ratings (-10 to 10) when compared to the other two (1 to 5). The paper also reports the worst accuracy for the Jester dataset for every number of groups. Our guess is that the problem is actually classification of certain groups, which tend to be very low compared to others, as it can be seen on Figure 2. Samples of certain clusters are classified as different class very often, and since there are approximately the same number of samples from each class, this makes mean accuracy lower. Still, we are reporting arguably increasing accuracy across iterations as it can be seen on Figure 3.
For the purposes of significance testing, we've chosen the experiment with the Netflix dataset and four clusters. Therefore, we ran 10 experiments with different seeds and got a mean accuracy $\overline{x} = 0.851$ with a standard deviation $s = 0.014$. Now, we are performing a two tailed t-test to determine the significance of the difference between the reported accuracy and our reproduced

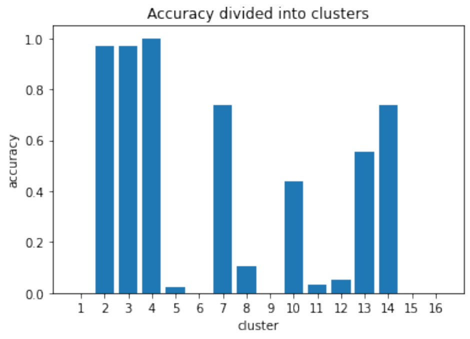

Fig. 2. Accuracy across groups for 16 clusters on Netflix dataset

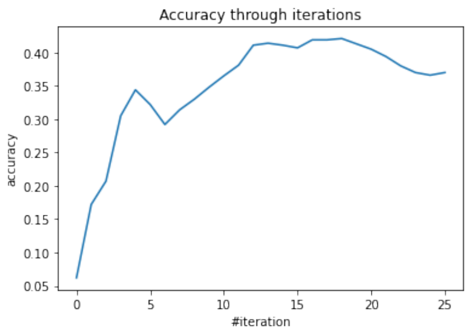

Fig. 3. Accuracy through iterations for 16 clusters on Netflix dataset

experiment accuracy. Our hypothesis for the t-test are as fallows:

H_0: The reproduced accuracy is equal to the reported accuracy (0.99)
H_1: The reproduced accuracy is not equal to the reported accuracy (0.99)

For significance level we are choosing $\alpha = 0.05$. We then calculate the t-value:

$$t = \frac{\overline{x} - 0.99}{\frac{s}{\sqrt{10}}} = -32.3$$

We ran 10 experiments, therefore we have to look up the critical t-value $t_{critical}$ from the t-distribution table with 9 degrees of freedom and our chosen significance level α.

$$t_{critical} = 2.262$$

Since $|t| > t_{critical}$, we're rejecting the null hypothesis. The difference between the reported accuracy and the observed accuracy is statistically significant.

4 SUMMARY

For this exercise, we are challenging the performances of a new method called cluster-based bandits as a solution to the challenge of quickly and accurately learning the preferences of new users in recommender systems. The technique takes advantage of the ability to group users into clusters based on similarity in preferences, which allows for faster learning due to the smaller number of clusters compared to items. Our results showed that taking the clarity and amount of information given about the algorithm and experiment, it was not possible to reproduce the results.

5 REFERENCES

Shams, S., Anderson, D., Leith, D. (2021). Cluster-based bandits: Fast cold-start for recommender systems new users. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1715-1724). ACM.