



HAL
open science

Exploration Principles for Decision-Making Systems with Binary Feedbacks

Henri Sohier, Romain Barbedienne, Sara Yasmine Ouerk, Sinitandjon Yaya
Yeo

► **To cite this version:**

Henri Sohier, Romain Barbedienne, Sara Yasmine Ouerk, Sinitandjon Yaya Yeo. Exploration Principles for Decision-Making Systems with Binary Feedbacks. IEEE International Systems Conference (SysCon), Apr 2024, Montreal, Canada. hal-04554762

HAL Id: hal-04554762

<https://hal.science/hal-04554762>

Submitted on 22 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exploration Principles for Decision-Making Systems with Binary Feedbacks

Henri Sohier
IRT SystemX
Palaiseau, France
henri.sohier@irt-systemx.fr

Sara Yasmine Ouerk
IRT SystemX
Palaiseau, France
sara-yasmine.ouerk@irt-systemx.fr

Romain Barbedienne
IRT SystemX
Palaiseau, France
romain.barbedienne@irt-systemx.fr

Sinitandjon Yaya Yeo
Ecole Polytechnique, IRT SystemX
Palaiseau, France
sinitandjon.yeo@polytechnique.edu

Abstract—The challenge for a recommender system is to explore a complete solution space, preventing informational ‘filter bubbles’ and ensuring a diverse range of recommendations. This paper introduces five exploration principles to better understand users and items while taking limited risks, as poor recommendation can also diminish user trust. The five principles apply to any type of recommendation system and are illustrated using a recommendation system for tourist points of interest (PoI). A novel interface was tested with open data and permitted to gather a preliminary set of user feedbacks, based on open data and a novel interface. Exploration raises two questions: which items should a given user be exposed to, and which users should be exposed to a given item. This latter question is studied with different exploration strategies, in order to check whether certain populations of users help quickly and efficiently learn about lesser-rated items. Synthetic data were used to compensate for the relative data scarcity of newly launched recommendation systems. Random user selection often proved sufficient, which can limit constraints when determining which items should a given user be exposed to, thus simplifying the exploration process. The principles outlined extend beyond PoI recommendation, having potential applications in industrial contexts such as simulation model recommendation and system parts suggestion.

Index Terms—exploration, recommendation, decision-making, synthetic data

I. INTRODUCTION

Recommender systems have become ubiquitous in the digital landscape, being employed in various domains such as e-commerce, entertainment, and social media [1]. These systems work by predicting the user’s preferences and recommending items accordingly. There are primarily three approaches to building recommender systems: content-based, collaborative filtering, and hybrid [2]. Content-based systems rely on known item features, while collaborative filtering relies on learned item features which are computed using similarities between user choices. The hybrid approach combines both.

Recommender systems can be based on explicit user feedback, such as liking or disliking recommendations. Explicit feedback is a direct indication of the user’s preferences and can be used to improve recommendation accuracy [3]. Recommender systems can also be based on user personal data (e.g.,

browser history) or implicit feedback (e.g., video watch time) at the cost of possible privacy concerns, more complexity and less interpretability [4].

The launch of any system is hampered by the scarcity of data, which is a critical element for the effective functioning of these systems [5]. The problem of such cold starts is often discussed for large systems already under operation, which must integrate new users or recommendable items. However, data scarcity is also an interesting challenge for all new systems, which must apply smart strategies for their very first users. Recommendations are sometimes facilitated by asking the user for demographic data (birthdate, sex, etc.), but this approach may not be worth the privacy concerns it raises [6].

The effectiveness of a recommender system is often gauged by its accuracy, i.e., its ability to predict whether a user will like an item. However, focusing solely on accuracy leads to the risk of a ‘filter bubble,’ where all recommendations are similar [7]. It is crucial to balance the accuracy of a recommender system with its ability to cover the whole solution space, i.e., its ability to make new types of recommendations without neglecting any kind of item [8]. This trade-off between accuracy and solution space coverage is a significant challenge in the design of recommender systems. Various approaches have been proposed to address this challenge, such as diversification algorithms [9], serendipity-oriented approaches [10], and exploration-exploitation strategies [11].

The exploration-exploitation dilemma is a foundational concept in the realm of algorithmic decision-making, particularly evident in the ‘bandit’ algorithm framework [12]. The ‘bandit’ algorithm tackles the multi-armed bandit problem, wherein a player selects from multiple options (or ‘arms’), each yielding a reward from an undisclosed probability distribution. The overarching goal is to optimize rewards accumulated over successive choices. The main difficulty lies in determining when to ‘explore’ lesser-known options versus ‘exploit’ the best-performing ones based on historical data.

Within the bandit algorithm’s framework, exploration aims to amass more data on various distributions, potentially resort-

ing to random selections. Conversely, exploitation capitalizes on historical rewards to guide decisions. Several strategies, such as the epsilon-greedy approach, employ a probability epsilon, determining the likelihood of an exploration-driven choice [13]. The epsilon-first strategy prioritizes exploration initially, while the epsilon-decreasing strategy gradually shifts from exploration to exploitation as more data accumulate [14]. Sophisticated bandit algorithms, like the UCB (Upper Confidence Bounds) [15] and the Thompson sampling algorithm [16], offer nuanced mechanisms to balance these aspects.

In the context of recommender systems, the "bandit" algorithm's principles provide invaluable insights, though with specific distinctions. Recommendation does not neatly align with the traditional multi-armed bandit model, where rewards follow predefined probability distribution. In recommender systems, users typically do not only look for "optimal" items they could stick to. Always recommending a specific type of items would diminish their perceived value [17].

This paper presents key principles to improve the exploration of original recommendations in the context of systems starting with scarce data. This approach can be applied to a variety of domains and use cases. For example, in industry, in the context of choosing the simulation model best suited to the problem. Or, in more critical piloting systems, for example, where the decision-maker has to choose an action to be carried out quickly, while taking into account his previous decisions in similar situations and/or the decisions of his team-mates in the same situation (recommendation), or proposing slightly different actions and then taking his feedback to update the system (exploration).

As an application example, the use case considered in this paper is the recommendation of POIs in France, with a like/dislike system, using the rich open-source data available in France. Different strategies to gain information on recommendable items are tested.

This work is part of an exploratory research activity at IRT SystemX, a half-private, half-public technological research institute on digital engineering. Beyond recommending POIs, this work aims at identifying possible challenges and solutions for new research projects on recommender systems in industrial companies (e.g., in the automotive or energy sector). A collaboration with the Free University of Bozen-Bolzano, in Italy, has provided valuable insights.

II. USE CASE

The exploration principles and approach can be applied to different recommendation systems. They have been applied to a use case for a tourist attraction recommendation system. The use case considered is a concept of web application focused on personal POIs recommendation (including both places and events). An interface was prototyped for test. As represented in Fig. 1, users are presented with a continuous series of POI recommendations. For each recommendation, users have the option to like, dislike, or simply pass. Additionally, they can indicate whether they have previously visited the suggested location. Users can either be geolocated or manually input

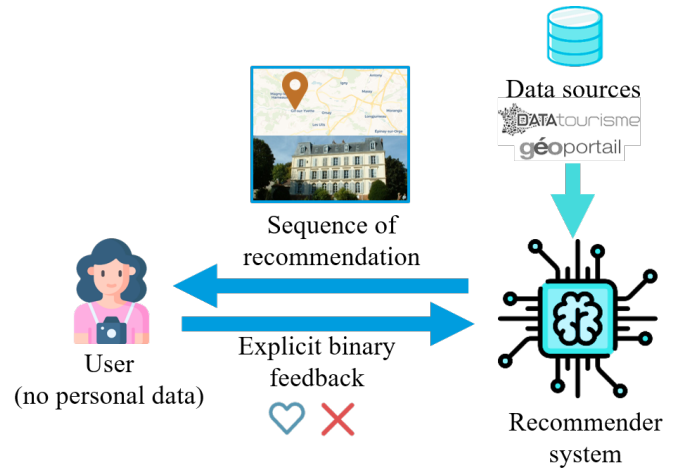


Fig. 1. General representation of the considered recommender system

any address. Thus, this concept is focused on the challenge to efficiently use a limited set of explicit data. The user is not asked for any additional personal information such as its age or its sex, both for privacy and simplicity. The user does not have to let the application record additional data on his/her behavior (mouse movement, clicks on menus, etc.). Fig. 2 represents a part of the interface prototyped for test.

At any time, users can review their recommendation history and filter, for example, liked but unvisited POIs. Not only does this service alleviate the tedious task of manually searching for outing ideas, but it also paradoxically helps users in breaking out of their informational bubbles. People naturally tend to fall into repetitive patterns, such as using the same search keywords when they look for information on search engines. While recommender systems are often criticized for potentially reinforcing such informational bubbles, they can also help pushing boundaries and expanding horizons.

The prototype leverages Datatourisme's graph database (datatourisme.fr), which, at the time of testing, encompassed 194,382 POIs. Datatourisme is a French initiative which consolidates, standardizes, and disseminates data sourced from French tourism operators and public institutions. Datatourisme includes 163 categories of POIs, with the following four top-level categories: Places, Products, Events, and Tours. For the purposes of this study, these categories were restructured as depicted in Fig. 3 to encapsulate three primary user motivations: 'Walking' (contemplative outings), 'Learning' (intellectual outings), and 'Having fun' (active outings). While Datatourisme serves as the primary data source, there's potential to integrate other French open-source datasets. For instance, the French National Geographic Institute offers data on population density and standard of living for any given GPS coordinate (www.geoportail.gouv.fr), which could enrich the contextual understanding of a POI's location.

To accumulate an initial dataset, the interface was introduced to 68 testers who were shown a total of 2418 recommendations simply based on the proximity between user



Fig. 2. Test interface (in French)

and POI. The feedback comprised 568 likes, 799 dislikes, and 1,051 instances where testers simply passed the recommendation. On 226 occasions, testers indicated prior visits to the recommended POIs. These recommendations spanned 1,244 unique Points of interest, with each POI being recommended an average of 1.94 times. Each tester was shown an average of 35.56 recommendations. This test was partly supported by the agglomeration community of Paris-Saclay, in the south of Paris.

This initial dataset was sufficient to develop simple rating predictors by matrix factorisation. A simple rating predictor was for example developed with a content-based approach solely based on the three root categories 'Walking', 'Learning' and 'Having fun'. With predicted values between -1 and $+1$, using a threshold between likes and dislikes at 0.6 permitted to reach a precision above 0.65 (65% of predicted likes were actual likes). Another rating predictor developed with a hybrid approach, to include an embedding learned by collaborative filtering, reached a precision above 0.7. While these predictors can be used for recommendation, they must be associated to exploration algorithms.

III. EXPLORATION PRINCIPLES

The dual objectives of recommendation and exploration often stand in delicate balance. On one hand, recommendation seeks to present users with items that align closely with their known preferences, ensuring a high probability of acceptance. On the other hand, exploration ventures into the less charted territories of a user's preferences, proposing items for which the user's reaction is unpredictable. This exploratory approach, while bearing inherent risks, is pivotal for deepening the sys-

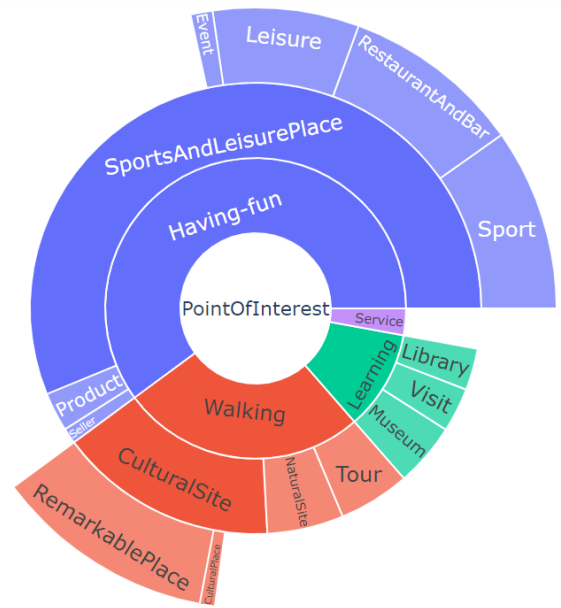


Fig. 3. Taxonomy for classifying points of interest

tem's understanding of the user, thereby refining subsequent recommendations.

This work introduces five exploration principles designed to glean more about users and items while limiting the risk of user dissatisfaction and the risk to ensnare in a narrow segment of the item space. Central to several of these principles is the concept of clusters which can be manually defined based on domain knowledge or automatically computed.

1 - Exploration of Distant Items. Exploration should target items for which the user's reaction remains uncertain, as these items help learn more about the user. These items are the most "distant", in the item space, from those the user has previously rated. Distant items must be explored in both content-based and collaborative filtering approaches. Such exploration requires to define: 1) a mathematical distance; 2) a computationally reasonable way to assess item distances, avoiding combinatorial explosion.

2 - Exploration of Lesser-Rated Items. Exploration aims to gather information about items that are less understood. These items are those with fewer ratings. This principle will benefit both collaborative filtering and content-based recommendation approaches. In the case of collaborative filtering, it enables recommendation system to learn more about the latent variables of the item. And in the case of content-based systems, it enables recommendation system to know whether the content-based variables have been properly filled in. For example, in a tourism application, if an item is listed only as a castle, but contains a garden. This item can be corrected to be recommended both to users who like castles and those who like gardens.

3 - Exploration of Well-Rated Items. The risks of user dissatisfaction during exploration can be mitigated by consid-

ering popular items. If an item is generally appreciated by others, it is more likely to resonate with the user. Two types of rating values should be considered: averages weighted by the number of ratings (to identify items well-rated by a large user base), and non-weighted averages (to identify items well-rated regardless of the number of users). In the latter case, exploration can progressively balance the rating value (i.e. gather information about the item).

4 - Exploration of Appreciated Clusters of Items. The risk of user dissatisfaction during exploration can also be mitigated by favoring appreciated clusters. For this approach, items are grouped into clusters, according to the similarity of their features. In this way, items with features that are closer in the latent representation space are more likely to be grouped together in the same cluster. Two types of rating values should be considered: the average user rating for the items cluster in question, or the average rating of the items cluster independently of the user (i.e. the average rating of the cluster he may have received from all users of the system). The user preference can either be proportional to the cluster's rating value or be more intricate.

5 - Exploration of Larger Clusters of Items. Prior principles might lead to a focus on smaller clusters (which might have a significant proportion of lesser-rated items, for instance). To mitigate this, exploration should be biased towards larger clusters. The bias can be proportional to the cluster's size or be more sophisticated.

The principle of exploring Lesser-Rated Items poses an important question: to which users should a lesser-rated item be recommended? Are there users whose ratings can swiftly provide insights into an item? If so, how does this interplay with the other principles? The rest of the work aims at answering these questions.

IV. SYNTHETIC DATA

In their early stages, recommendation systems often grapple with limited data. Despite this scarcity, it is imperative to test these systems with progressively larger datasets to ensure their scalability. One viable approach is the generation of synthetic data [18].

Synthetic data is defined as artificially labeled information. The proposed approach is based on stochastic modeling. The aim is to generate 3 elements: synthetic POIs, synthetic ratings, and a distinction between past ratings and future ratings.

A. Generation of synthetic POIs

In the first part, the approach artificially generates n_m POIs (with $n_m \in \mathbb{N}^*$). Each POI is expressed as a vector of dimension $n + 3$ with $n \in \mathbb{N}$. Each dimension X_k corresponds to a feature that characterizes the POI. The first feature corresponds to the POI ID and the last two to the GPS coordinates. The other features depend on the dataset and the recommendation system. Indeed, features can represent a latent space (collaborative filtering) or real features (content based).

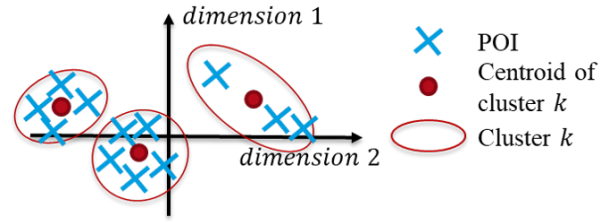


Fig. 4. item set generation in dimension 2

the coordinates are generated by considering that for each POI cluster, the POI features were distributed according to a centered multivariate normal distribution. the position of the centroids (the average of the multivariate normal distribution) is achieved by following a random distribution. The covariance matrices are given by the user. This will enable us to test the limits of our approaches when two clusters are difficult to separate (high variance). For example, for $n = 2$, with 3 clusters containing 4, 5 and 3 POIs respectively, the POI space is shown in Fig 4.

The choice to generate synthetic data in the form of clusters is justified by the exploration-exploitation dilemma that can be considered with clusters. Clusters can manifest themselves in two main ways: as manually defined categories, or as automatically computed groups based on element properties.

B. Generation of synthetic ratings

The second part introduces the concept of user. Let n_u be the number of users generated. $n_u \in \mathbb{N}^*$. For each user $j \in \llbracket 1, n_u \rrbracket$ and each POI $i \in \llbracket 1, n_m \rrbracket$, the approach generates a score $s(i, j)$. Note that the scores here are binary, so $s(i, j) \in \{-1, 1\}$ $s(i, j) = 1$ if user j liked POI i and $s(i, j) = -1$ if user j disliked POI i . Note that for other uses, scores are not always binary. It can be a discrete or a continuous variable. In this case, Bernoulli's law no longer applies. Another, more appropriate law will have to be chosen. Typically $s(i, j)$ is represented in matrix form S .

$$S = \text{POI} \begin{matrix} \xrightarrow{\text{User}} \\ \left(\begin{array}{cccc} s(1, 1) & s(1, 2) & \cdots & s(1, n_u) \\ s(2, 1) & s(2, 2) & \cdots & s(2, n_u) \\ \vdots & \vdots & \ddots & \vdots \\ s(n_m, 1) & s(n_m, 2) & \cdots & s(n_m, n_u) \end{array} \right) \end{matrix} \quad (1)$$

The generation of the score matrix must ensure that individual users have specific preferred clusters and that certain items are popular. The matrix generation process is based on probabilistic generation. First, a matrix P is generated indicating for each user j the probability $p(i, j)$ that he will assign a positive score to item i (where for each $i \in \llbracket 1, n_m \rrbracket$ and $j \in \llbracket 1, n_u \rrbracket$, $p(i, j) \in [0, 1]$). Then, based on this probability, a score ranging $s(i, j)$ from -1 to 1 is determined using a Bernoulli distribution of probability $p(i, j)$.

The distribution of $p(i, j)$ has been chosen to best fit the real data. The real POI and user distributions in relation to

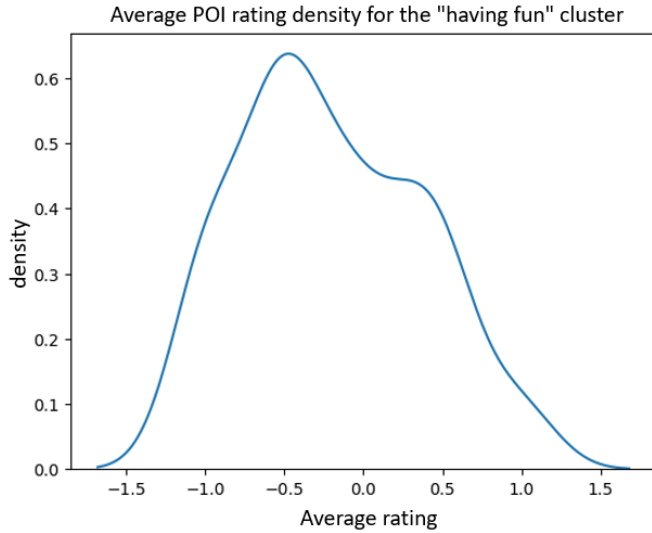


Fig. 5. Density distribution of POIs according to score related to the "Having fun" cluster

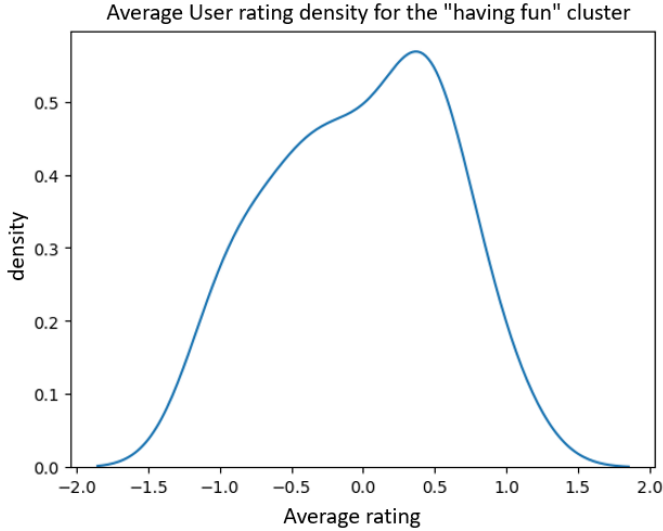


Fig. 6. Density distribution of users according to scores related to the "Having fun" cluster

score, for the "having fun" cluster for example, are shown in Fig. 5 and 6. Note here that the density curves shown in the figures have been estimated using the kernel density estimation method, which can result in densities appearing outside the typical range, specifically less than -1 and greater than 1 [19]. For the remainder of the paper, these distributions are approximated using truncated normal distributions with bounds between -1 and 1 .

Thus, the generation of $p(i, j)$ was carried out in such a way that the mean of $p(i, j)$ according to users ($\frac{1}{n_u} \sum_{j=1}^{n_u} p(i, j)$) and the mean of $p(i, j)$ according to POI ($\frac{1}{n_m} \sum_{i=1}^{n_m} p(i, j)$) respect the truncated normal law distributions.

The parameters of the truncated normal distributions were calculated by comparing the distributions with the actual data. The Kolmogorov test [20] is used to measure the similarity between these density distributions. The test returned a p-Value

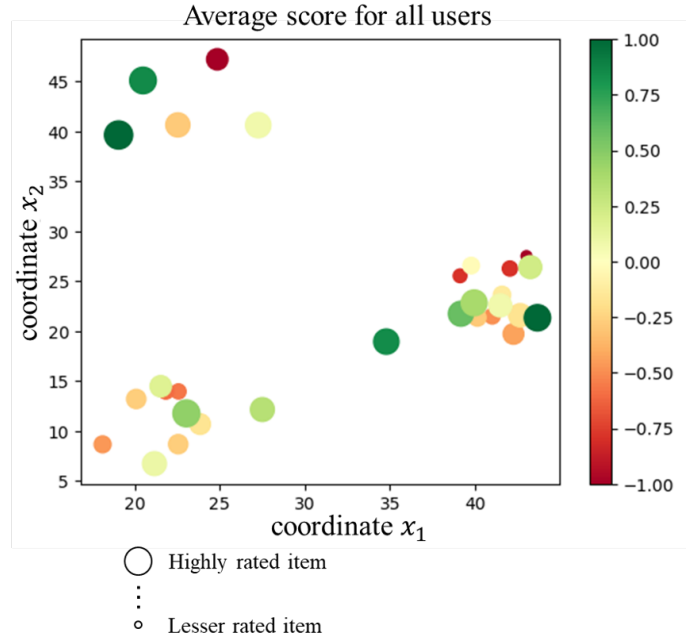


Fig. 7. Example of a data-set generated in 2 dimensions (for $n = 2$), 3 clusters, 30 POIs and 20 users

of 0.1, which is greater than the value $\alpha = 0.05$, allowing us to accept the null hypothesis that the generated mean densities are identical to those of the real cases.

C. Generation of the mask matrix

The third part consists of generating the mask matrix. This matrix indicates whether or not user j has already rated item i . For each user j with $j \in \llbracket 1, n_u \rrbracket$ and each POI i with $i \in \llbracket 1, n_m \rrbracket$, the approach generates a values $r(i, j)$, $r(i, j) = 0$ if the user did not rate the item yet, and $r(i, j) = 1$ if the user has already rated the item.

This identification of past ratings are based on uniform distribution. It can also be based on parameters such as the geographical distance between a user and a POI, using their respective latitudes and longitudes. The further away a user is from a POI, the less likely it is that he or she has already voted for it. The probability of a user voting for an item decreases rapidly with distance to the POI.

The outcome of the generated data is illustrated in Fig. 7. For visualization purposes, features were generated in two dimensions. Three item clusters were constructed, with 30 POIs generated for a set of 20 users. Average ratings vary between -1 and 1. The colors represent the average rating of a POI, while the circle's size is proportional to the number of users who voted for that particular POI.

V. SELECTION OF OPTIMAL USERS FOR ITEM EVALUATION

A. Description of the exploration algorithms

In line with the principles outlined in section III, items with fewer ratings can be presented to users to glean insights from their feedback. This prompts the question: Are there specific

users to whom these items should be recommended in order to gather information more effectively? Using the generated data, we explored five distinct strategies.

The first strategy, termed *random*, is a random selection of users who will evaluate the item. This method will serve as a reference in comparison with other methods.

The second strategy, termed *rank by user*, aims to determine whether a POI corresponds to the appropriate item cluster. Users are ranked according to their average scores corresponding to the cluster to which the POI is associated. The POI is first presented to users who show a preference for that cluster, ensuring that the item matches their interests. It's then shown to users who generally dislike that item category, to gauge whether they have a negative view of the proposed POI.

The third strategy, *rank by cluster*, groups users by clusters. Each user is represented as a vector whose dimension is equal to the number of item clusters. Each component mirrors the average score allocated to items within that cluster. Clustering was performed using the k-means algorithm, with the optimal cluster count determined via the elbow method [21]. User clusters are then ranked based on the average rating they have accorded to the item cluster the POI is part of. Similar to the previous strategy, the POI is first proposed to a random user from an high-ranking cluster and subsequently to one from a low-ranking cluster.

The fourth strategy is named *cluster-based sampling*. In this approach, we group users according to their average scores in the clusters of items. We then randomly select users to whom we wish to present the item. The probability that a user will be selected depends on the size of the various item clusters obtained (i.e. a user belonging to a larger cluster has a higher probability of being presented with the item). We consider the opinion of the largest class of users to be crucial.

Lastly, the fifth strategy, *cluster-based sampling +*, which is a variant of the previous strategy. Once a group of users has been chosen (the most populated clusters have a higher probability of being selected), the selected user is the one closest to the cluster centroid. The underlying principle is that users close to the centroid are more representative of their group than those further away.

B. Application

In order to compare these strategies, one dataset was generated. It includes 1,000 POIs and 400 users. A POI is considered as well-characterized if the mean of the actual scores from users who have voted for the POI (the effective average, $m_{eff} \in [-1, 1]$) is close to the mean of the actual scores when every user has voted for the POI (the final average, $m_{final} \in [-1, 1] \setminus \{0\}$).

$$r_{know} = \frac{m_{eff}}{m_{final}} \quad (2)$$

Note that if m_{final} is null, the POI is not evaluated. This measure tends towards 1 as the number of users voting increases. The five previously discussed algorithms were applied for each POI. The evolution of r_{know} was calculated as a

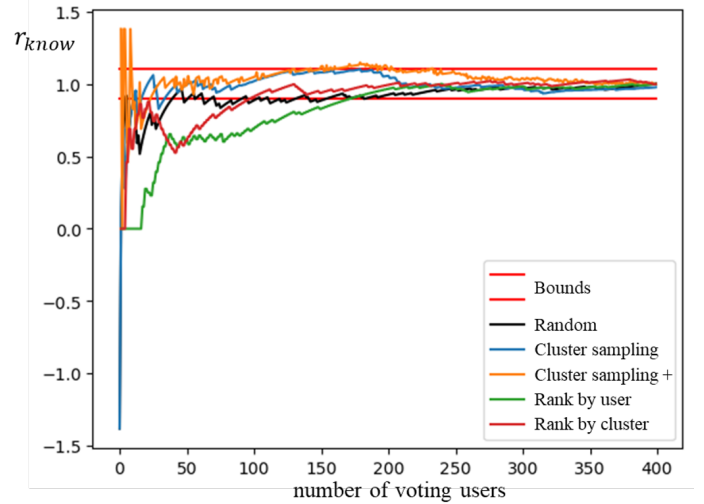


Fig. 8. Comparison using the first dataset

function of the votes for a given POI and averaged across all POIs in the dataset.

The results for each computational strategy are represented in Fig. 8. The effective score is considered close to the real score (final score) if the r_{know} ratio is in the range $[0.9, 1.1]$. This interval is also represented in the figure by the 2 horizontal red lines. Once the ratio r_{know} enters this interval and remains within, the POI is considered as well-characterized.

An analysis of the results reveals some notable findings. While the "cluster-based sampling" and "cluster-based sampling+" methods show rapid convergence, they don't significantly outperform the "random method". Unlike the "random method", which shows variability across simulations, both "cluster-based sampling" and "cluster-based sampling+" tend to initially overestimate the mean before settling down. Their curves exceed 1 before converging.

The "Rank by User" and "Rank by Cluster" methods have curves that are always under an r_{know} ratio lower to 1. Furthermore, for all tests performed, there was an initial convergence towards 0.5. This suggests that the curves can be corrected by a machine learning model. The main objective of this machine learning model is to predict the final average score m_{final} of an item according to: its features $\{X_k\}_{k \in [1, n_m]}$, the average score given by users who have already voted m_{eff} , and the number of users who have already voted n_{eff} . Note that users are selected using one of the exploration strategies described above. In order to train the machine learning model, data were generated using the presented synthetic data generation algorithms. 400 users and 1000 POI are generated. For each point of interest, all users are unmasked sequentially in accordance with the exploration algorithm studied. Each unmasked user is considered an independent observation. The POIs are divided in a ratio of 0.6. 60% of POIs are used as a training set to train the machine learning model. 40% of POIs are used as a test set to test the trained machine

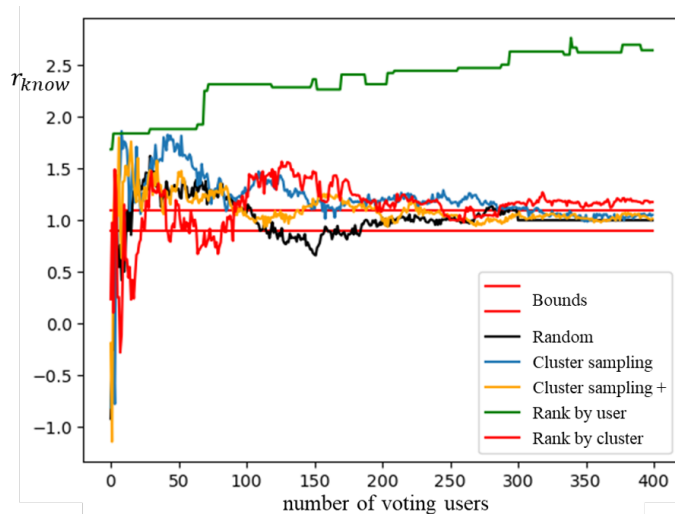


Fig. 9. Prediction results with XGBoost

learning model. The loss function used is the mean squared error (MSE): $loss = \frac{1}{n} \sum_1^n (m_{final} - \widehat{m}_{final})^2$ where n is the total number of data, m_{final} is the real average final mean and \widehat{m}_{final} is the average score mean predicted by machine learning model.

An XGBoost model was implemented to predict this final average. The test set is used in order to calculate the r_{know} . As can be seen in Fig. 9, the XGBoost model does not significantly improve the results of each method. The "Rank by cluster" model is the only one to have been improved by the XGBoost model. These preliminary results could be improved by other prediction models, such as a neural network that can make use of the massive data generated by the synthetic model.

However, the proposed approach is limited by the fact that the data are derived from synthetic data generated from real data. The acquisition of real data could improve the results.

VI. CONCLUSION

Five important principles are expressed to improve exploration, with a certain emphasis on clusters. A key challenge of the exploration of Lesser-Rated Items is to determine which users should be exposed to these items. Synthetic data, crafted to mirror real-world scenarios, was employed to test various user selection strategies. Random user selection often proved sufficient, simplifying the exploration process.

This study was applied into recommender systems for Points of interest (POIs) based on binary user feedback, where each recommendation can be liked or disliked. Testers showed high engagement even with basic recommendations, quickly building an initial dataset. The system's design could counter the informational bubbles users often create by repetitively searching the same content.

This study could be applied to other recommendation systems, for example to recommend simulation models in a company [22]. This is an important industrial challenge, as the simulation engineers or simulation architects spend a lot

of time producing a model that has already been developed by another engineer. In this context, the item to be recommended would be the simulation model, and the user would be the simulation engineers. The feature of the element would be the various characteristics of the simulation. These features can be based on the MIC.core concept [23]. The GPS coordinates correspond to the various company entities. Indeed, a simulation model may be difficult to obtain if it has been produced by another department within the company. And clusters can represent simulation domains (electronics, heat transfer, mechanical, etc...)

Another application would be the recommendation of MBSE system models of similar systems. When a system is developed, it is rarely from scratch. A recommendation system would enable a system engineer to start from a system model that has already been produced. In this context, the item to be recommended would be the system model, and the user would be the system architect. The features of item are the various characteristics of the system. These characteristics can be extracted from a system model, as the main SysML blocks (if the system model is implemented in SysML). The GPS coordinates correspond to the company's various entities. The group can be made up of a multilevel taxonomy [24].

Future work could explore how to adhere to the exploration principles when managing the trade-off between recommendation and exploration. Additionally, future studies could consider how to incorporate temporal dynamics into these principles so that they can help ensure that the recommender system remains relevant over time and adapts to both the cyclical nature of seasons and the evolving preferences of users.

REFERENCES

- [1] F. Ricci, L. Rokach, and B. Shapira, *Introduction to recommender systems handbook*. Springer, 2011, pp. 1–35.
- [2] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.
- [3] M. J. Pazzani and D. Billsus, *Content-based recommendation systems*. Springer, 2007, pp. 325–341.
- [4] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *Proceedings of the 10th international conference on Intelligent user interfaces*, 2005, pp. 167–174.
- [5] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002, pp. 253–260.
- [6] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 2008, pp. 208–211.
- [7] E. Pariser, *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.
- [8] P. Castells, N. J. Hurley, and S. Vargas, *Novelty and diversity in recommender systems*. Springer, 2015, pp. 881–918.
- [9] C. N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 22–32.
- [10] T. Murakami, K. Mori, and R. Orihara, "Metrics for evaluating the serendipity of recommendation lists," in *Proceedings of the 2008 ACM conference on Recommender systems*, 2008, pp. 11–18.

- [11] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan, "Scene: A scalable two-stage personalized news recommendation system," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2010, pp. 125–134.
- [12] 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.
- [13] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [14] 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.
- [15] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.
- [16] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *Advances in neural information processing systems*, 2011, pp. 2249–2257.
- [17] Y. C. Zhang, D. O. Séaghdha, D. Quercia, and T. Jambor, "Auralist: introducing serendipity into music recommendation," in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012, pp. 13–22.
- [18] P. Merinov, D. Massimo, and F. Ricci, "Behaviour-aware tourist profiles data generation," in *Proceedings of the 13th Italian Information Retrieval Workshop (IIR 2023)*, 2023, pp. 3–8.
- [19] P. Van Kerm, "Adaptive kernel density estimation," *The Stata Journal*, vol. 3, no. 2, pp. 148–156, 2003.
- [20] V. W. Berger and Y. Zhou, "Kolmogorov–smirnov test: Overview," *Wiley statsref: Statistics reference online*, 2014.
- [21] P. Bholowalia and A. Kumar, "Ebk-means: A clustering technique based on elbow method and k-means in wsn," *International Journal of Computer Applications*, vol. 105, no. 9, 2014.
- [22] S. Mejdal, O. Penas, R. Barbedienne, R. Plateaux, M. Bisquay, and J.-p. Brunet, "Ontology-based search engine for simulation models from their related system function," in *INCOSE International Symposium*, vol. 31, no. 1. Wiley Online Library, 2021, pp. 1145–1159.
- [23] H.-M. Heinkel, H. Sohler, and P. Mai, "Mic core specification," Jul 2023. [Online]. Available: <https://mic-core.github.io/MIC-Core/main/>
- [24] H. Sohler, L. Petitdemange, and P. Lamothe, "Identification of systems with similar chains of components for simulation reuse," in *2021 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE)*. IEEE, 2021, pp. 1–8.