Learning Choice Models for Simulating Users' Interactions with Recommender Systems⋆

Naieme Hazrati*, Francesco Ricci

Free University of Bolzano-Bozen, Bolzano, Italy

Abstract

To better understand the long-term effect of Recommender Systems (RSs) on users' choices, some recent studies have simulated users' interactions with RSs. The RS impact on users is then quantified by measuring global properties of the simulated choices, their distribution and quality. The accuracy of the simulated users' Choice Model (CM), i.e., how the simulated users make their choices among the recommended items, significantly contributes to the validity of the results. In fact, while some CMs have been suggested as plausible, none of them was proved to generate choices "close" to the actual choices, i.e., to those that real users have done, or will do, when exposed to the same recommendations.

In this paper, we study two CMs: the Multinomial Logit (MNL) and one based on CatBoost, an algorithm for gradient boosting on decision trees (ML). We train these models to correctly predict the target users' choices, given a set of system-generated recommendations. We found that, the ML model outperforms the MNL one with regard to classical accuracy metrics (precision and balanced accuracy), while MNL's generates choices that better reproduce the distribution of the real choices (Gini index, Shannon Entropy and catalogue coverage). We, therefore, argue that MNL, when simulating users' behaviour, is more suitable for understanding the global impact of a deployed RS.

Keywords

Recommender System, Choice Model, Simulation,

1. Introduction

Recommender Systems (RSs) are tools aimed at supporting the choice-making process of users and are often evaluated by measuring the precision and the quality of the generated recommendations [1]. However, to assess the true value of an RS, it is also important to understand the impact that it has on users' choice behaviour, e.g., the distribution and quality of the choices' that users make when considering the recommendations [2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

Some previous works have tried to assess how RSs can influence users' choices [12, 13, 14, 10, 15, 16]. These studies leveraged the simulation of repeated users' interactions with an RS along a temporal interval, by assuming that users select items among the recommended ones by adopting a given and "plausible" Choice Model (CM). By assuming the validity of the CM, these studies analysed aggregated measures of the impact of the RS on the distribution of the choices made by their users. The schema in Figure 1 shows the general design of the simulations proposed in the literature [11, 10, 13, 17, 15].

Perspectives on the Evaluation of Recommender Systems Workshop (PERSPECTIVES 2022), September 22nd, 2022, co-located with the 16th ACM Conference on Recommender Systems, Seattle, WA, USA.

*Corresponding author.

nhazrati@unibz.it (N. Hazrati); fricci@unibz.it (F. Ricci)

In a simulation design some important components must be properly selected: the RS, the CM of the users, the awareness set (users' knowledge of the catalogue of items), and the number of simulated choices per user. While simulations have obtained interesting results, some issues must be faced in order to increase their validity:

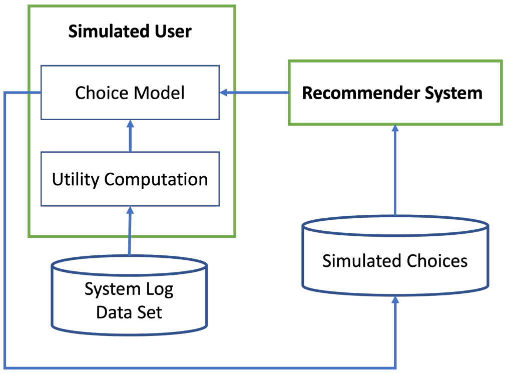


Figure 1: Architecture of the simulation of users choices in RSs.

1. ***Single criteria CM***: the CMs used in past studies have been designed by referring to a simple decision criteria, considered as important by the designer. Simulated users when exposed to recommendations, react by considering this single criteria, e.g., either the item popularity or its rating [10, 15, 18, 19]. Hence, users' interactions with the RS is assumed to be quite simple, and it is motivated by the desire to isolate the effect of these criteria on users and their choices. However, real users' CMs are expected to be more sophisticated, and be simultaneously influenced by a variety of factors, such as the combination of item popularity and perceived quality [20].
2. ***Accuracy of the CM***: the reliability of the simulation results is strictly dependent on the accuracy of the CM. We note that a proper CM should correctly identify the target users' choices, when they are exposed to the system-generated recommendations (choice set). In fact, a user's choice is dependent on the whole choice set, not only on the individual items, independently considered. However, to the best of our knowledge, previous studies have defined the CM either by relying on general heuristics (e.g., users tend to choose the top recommended items) or by fitting the parameters of a class of CMs so that simulated choices share with observed choices of real users some target properties: for instance, the choice diversity of simulated choices is similar to the diversity of real choice data) [10, 17, 14].

To address these limitations, we consider two CMs [21] and we fit them to reproduce observed users' choice behaviour when they are exposed to a set of recommendations. The CMs are the Multinomial Logit CM (MNL) and CatBoost, an algorithm for gradient boosting on decision trees (ML). The data set used to learn the CMs contains historical interactions of users with a particular RS (an operational one). Data includes the shown recommendations and the consequent choices of the users. Moreover, these CMs depend on several features of the users and the items, hence the CMs incorporate several criteria.

After the historical choices information is used to train the CMs, the CMs are exploited to simulate the choices over multiple time intervals. More precisely, the CMs are initially trained with choice data collected until a given timestamp t_0, and then we simulate users' choices in successive intervals of time. At the end of each time interval simulation, the CM is retrained with the choices simulated in the past intervals. This approach has a better potential to show the predictive power of a simulation: after the given timestamp t_0 no additional information about true users' behaviour is given to the simulation. Eventually, we evaluate the considered CMs' accuracy in reproducing the true observed choices (after t_0). We first measure the CMs accuracy of predicting the actual users' choices. By using classical evaluation metrics (precision and accuracy), we have found that the ML choice model is more accurate than the MNL one. In a second stage, we compare how the two CMs reproduce the global distribution of the actual choices. We measure metrics such as the Gini index, the Shannon Entropy, and the catalogue's coverage. We observe that, differently from the accuracy metric, in this case the MNL model better reproduces the distribution of the actual users' choices, compared to the ML model. Hence, in this respect, MNL may be a better option to simulate and anticipate the collective choice behaviour of users.

In conclusion, our study shows that simulations have the potential to draw a proper picture of the long-term impact of an RS, hence and can help RS researchers to anticipate the long term impact of a deployed RS. However, the selection and adaptation of the simulated users' CM is a major component that requires a proper definition and training.

2. Related Work

Inspired by economics literature, most of the simulation studies aimed at understanding the impact of RSs on users' choice behaviour, adopted the Multinomial Logit (MNL) model. MNL assumes that a user exposed to a set of items (choice set) evaluates them by computing their utility. Then, the user chooses an item with a probability that grows with the item's utility. Items' utility is defined/learned based on additional assumptions, or heuristics, chosen by the simulation designer.

For instance, Fleder et al. [13] generated a synthetic data set of users and item profiles, in the form of randomly generated vectors. The item's utility decreases as the Euclidean distance between the user and the item profiles grows; as a consequence, the smaller the distance between the user and the item profiles, the more likely it is for the user to choose the item. Their study was influential, even though their approach has some limitations. Firstly, the simulation is based on a CM that depends on the distance between randomly generated user and item profiles, and there is no evidence that it can properly depict the actual choice behaviour of real users. As a matter of fact, their findings can only provide a limited picture of how users make choices in actual applications. To address this limitation, in this paper, we use a data set of real users' interactions with an RS. The considered CMs are trained using these interactions; hence, we build CMs that have a better potential to produce an output that matches the actual CM of the users.

In our previous study [10], we used a real rating data set in order to define a simulation process that could faithfully predict the actual choice behaviour of RSs users. We also assumed that the users' CM follows the MNL model, but the user utility for an item was estimated as proportional to the predicted rating of the item. The users were assumed to choose among the items in their choice set, which is there called awareness sets and contains both recommendations and some popular and high utility items. The simulations were run with alternative RSs. The CM behaviour was adapted to obtain a Gini index similar to the Gini index computed for the actual users' chcoices. Moreover, the users were assumed to consider only one criteria when making choices, namely, the rating of the considered item. However, as a matter of fact, the correctness of the CM was not properly tested. In fact, even though we tuned the CM to reproduce a "correct" Gini index, we were not able to properly fit the model, as we did not have information about the actual users' choice sets, which was considered when making an observed choice.

We note that the global distribution of simulated choices depends on the choice sets of the simulated users, their assortment and distribution, and this information should be exploited in the training of the CM. Moreover, the MNL model used in [10] depends only on the predicted ratings of the items, hence it makes a simplifying assumption that users are influenced by a single criteria in their choices. To address these limitations, in this paper, we leverage a data set of users' choices, where we have information about the users' actual choice sets, i.e., the recommendations provided to the users, and their subsequent choices. We use this data set to learn two candidate CMs (MNL and ML) that depends on multiple features of the users and the items. Then we assess the accuracy of the two CMs in simulating the users' choices.

Other studies investigated the impact of alternative users' CMs on the distribution of users' choices. By simulating simple user CMs, these works aimed at understanding the impact of specific choice behaviours on the global distribution of the choices. For instance, Yao et al. [7] simulated alternative CMs, varying the users' tendency to pick popular items. Although such CMs are simple and probably quite distant from those of real users, these studies contributed to developing a qualitative understanding of the impact of some specific behaviours. In addition, Szlávik et al. [14] modelled alternative CMs when users receive recommendations, assessing the impact of users' reliance on recommendations on the diversity of choices. In a more recent simulation study of users' rating behaviour [15], several user choice models, referred to there as consumption strategies, were considered. The study aimed at understanding the effect that the users' reliance on recommendations has on the performance of the RS. Finally, in [19], the authors model four alternative choice behaviours, analysing the impact of users' tendencies to choose more popular, more recent, highly rated items, or to rely more on the recommendations (modelling items' position bias).

3. Data Set

We have used data provided by the Recombee company that were logged from a retail website selling health and sport-related products such as sports clothes, sport-related accessories and protein complements. Users' timestamped interactions with the website are stored. Precisely, for each user, the timestamped recommendations that each user received are recorded, and the consequent clicks, purchases, and cart-additions are stored as well. Data comes from a web system log, and the system features multiple "endpoints" where recommendations are presented to the target user, e.g., on the home page, at the bottom of the items' detailed view web page, or on the user's cart page. The number of recommendations may differ at each endpoint.

We have performed our analysis on a sample of this data set, which is obtained by one of these endpoints only. More precisely, the endpoint that we consider is the bottom area of each item's page, where 12 recommendations are shown to the user. Recommendations come from different RSs; some recommendations are related to the main item presented in the page, while others are generated by another specific recommendation algorithms, which we ignore. Users may click on some of the recommended items, and some of these clicks may bring to a purchase. We have performed our analysis on a six months span of the data, by considering users with at least 20 recorded purchases. This filter is applied to reduce the data to a processable size, and skip users with incomplete profiles. In the finally used data set, there are 250,000 recommendation requests with 935 users and 5600 items.

Our analysis aims at modelling users' choices when they are exposed to the system-generated recommendations. Here a user's choice is a "click" on one of the received recommendations. This click will take the user to the detailed page of the item, which is again, augmented with another set of 12 recommendations. We note that a user may leave the page without clicking any of the recommended items. Each recommended item is described by features, such as, brand, category, item type (single or bundle), section, and price. The users are also characterised by several features, such as, age, city, postcode, and gender. We also use, for each recommended item, items popularity in the past x days ($x \in \{1, 5, 10, 30\}$), items' age (the time difference between the release date of the item and the recommendation time), user and items' embedding (from ALS matrix factorisation), user and item collaborative filtering score (dot product of the embeddings). We note that the embeddings are computed using the entire data set. Table 1 shows the features used in our CMs.

Table 1
User/Item interaction features used in the considered CMs

Recommendation rank,	Item category,
Item popularity in the past x days ($x \in \{1, 5, 10, 30\}$),	Item sub-category,
Item's age,	User's city,
Price,	User's age,
Regular price,	User's gender,
Outlet (Boolean),	Item embedding,
Brand,	User embedding,
Item type (in a bundle or single),	User-Item embeddings dot product.

4. Choice Models and Simulation of Choices

We aim at building a simulation process that, starting from an initial set of system log data, the data present in the log up to a certain point in time t_0, simulates the subsequent choices made by the users when they are exposed to the system-generated recommendations.

U is the set of users, and the set of items is denoted by I. We assume to have the RS generated choice sets and the corresponding users' choices; a user selected one or more items when exposed to a sequence of choice sets (each one is composed by a set of recommended items). The choices logged up to time t_0 are stored in the set Q^0. The elements of this set are triples (u_k, i_k, C_k), $k = 1, \dots, K_0$, and $K_0 = |Q^0|$. Each triple is composed by a user $u_k \in U$, that chose item $i_k \in I$, when the choice set was $C_k \subset I$. Note that $i_k \in C_k$ and a user may appear multiple times in this set as it may have performed multiple choices before time t_0.

The rest of the choice data, observed after t_0, is split into L time intervals. We indicate with Q^l the set of observed choices registered in the time interval $]t_{l-1}, t_l] = \{t \in \mathbb{R} : t_{l-1} < t \leq t_l\}$. We want to simulate the choices in each interval $l \in \{1, \dots, L\}$, by using the knowledge of the choices contained in $Q^0 \cup \hat{Q}^1 \cup \cdots \cup \hat{Q}^{l-1}$. We denote with $\hat{Q}^l$ the set of simulated choices in the interval $]t_{l-1}, t_l]$. In other words, the simulation of the choices in a time interval uses the knowledge of choices observed before t_0 and the simulated choices in the previous intervals. A choice is simulated by using a choice model (CM); given an observed choice set (present in a data set) the CM simulates/predicts the choices that the user has made when exposed to that choice set.

We use two CMs to simulate/predict the users' choices when exposed to system-generated recommendations: the Multinomial Logit (MNL) model and the CatBoost model. The details of each model are discussed in the following.

4.1. MNL - Multinomial Logit Choice Model

The Multinomial Logit (MNL) choice model, is based on the computation of the utility of a user u for an item i, which is assumed to be $v_{ui} = \beta' \cdot x_{ui}$, where x_{ui} is the joint feature vector representation of u and i, and β' weights the importance of the user and items features. β' must be learned by using a set of training choices. We note that, in addition to the items in I, we assume that in a choice set, there is an ever-present dummy item, labeled as i_0. This represents the no-choice action of the user, i.e., it is the choice when the user does not select any of the recommended items in the choice set. We force the utility of the no-choice to be null, i.e., $v_{ui_0} = 0$. We also note that in a real observed choice set, the user can choose multiple items. Hence, in MNL, we treat each choice independently and we create a separate data point for each item that was chosen. For instance, if the user u is recommended with $C = \{1, 2, 3, \dots, 12\}$ and selects items 1 and 2, then our history will contain two triples: $(u, 1, \{1, 2, 3, \dots, 12, i_0\})$ and $(u, 2, \{1, 2, 3, \dots, 12, i_0\})$. Under the Multinomial Logit Choice Model, if a set of recommendations C (choice set) is generated for the user u, then the probability that item $i \in C$ is chosen by user u is given by:

$$P_{ui}(C) = \frac{e^{\beta' \cdot x_{ui}}}{1 + \sum_{j \in C} e^{\beta' \cdot x_{uj}}} \tag{1}$$

We note that the value 1 in the denominator is used to properly define a distribution of probability when the dummy item is added to C to form the choice set. In this way, since v_{ui_0} is always equal to 0, the probability of choosing the dummy item is equal to $1/(1 + \sum_{j \in C} e^{\beta' \cdot x_{ju}})$. Our learning goal is: given a set of observed choices, Γ, e.g., the choices in Q^0, to compute the vector β' that minimises a proper cost function: the mismatch between simulated and real choices. We use Maximum Likelihood Estimation (MLE) to estimate the β' coefficients. Accordingly, the MLE problem is formulated as below:

$$\max_{\beta} \ell\ell(\beta|\Gamma) \tag{2}$$

Where the Log-likelihood is computed as following:

$$\ell\ell(\beta|\Gamma) = \sum_{(u,i,C)\in\Gamma} \beta' \cdot x_{ui} - \log(1 + \sum_{j\in C} e^{\beta' \cdot x_{uj}}) \tag{3}$$

Since the data set we have used is extremely imbalanced, i.e., 95% of the choices are no-choices, we select 10% of the no-choice events together with all the true choices to proper items, and solve Eq. 3 by using stochastic gradient ascent. However, since the relative size of choice and no-choice data is manipulated, we overestimate the size of β', which leads to an overestimation of the probability of choice compared to no-choice. Hence, we scale down the values of β by a constant coefficient δ. The value of δ is learned using the validation data set.

Choice Simulation As we mentioned before, our goal is to simulate the choices of the users in L time intervals successive to a given time point t_0. So, in a first step, the MNL model is trained on the choices in Q^0 to simulate choices in $]t_0, t_1]$ and produce a set of choices $\hat{Q}^1$. Then, in the successive time intervals ($]t_{l-1}, t_l]$, $l = 2, \ldots, L$), MNL is trained on the set of choices $\Gamma = Q^0 \cup \hat{Q}^1 \cup \cdots \cup \hat{Q}^{l-1}$ to generate the simulated choices $\hat{Q}^l$. That is, the observed choices in Q^0 together with the simulated choices in the previous intervals are iteratively used for retraining the CM.

4.2. ML - CatBoost based Choice Model

We use the same data and features used in the MNL to train a second CM. The computational goal is here to predict, for each pair of user and recommended item, whether that item is chosen by that user or not. Hence, we solve a binary class classification problem, where class 1 is associated with “choice”, and class 0 is associated with “no-choice”. We call this CM generically as ML. ML, differently from MNL, does not leverage any information coming from the fact that a choice is one of the 12 recommendations and treats each recommendation independently from the others.

The precise ML model used for choice prediction is CatBoost [22] (short for “categorical boosting”); it is a gradient boosting algorithm on decision trees. CatBoost was selected among multiple tested models (ADA, XGboost, Random Forest, and Logistic Regression) in a preliminary analysis based on precision and recall performance. Another motivation to select CatBoost is its classification good performance with input features of multiple types (numerical, categorical, and ordinal) [21, 22]. We recall that our input feature vector (joint representation of the user and the item) contains a mixture of feature types: numerical (e.g. embeddings), ordinal (e.g. rank of the recommended items) and categorical (e.g. brand). CatBoost is trained to minimise cross entropy, and the parameters of the model are tuned with the validation data set.

Unlike MNL, where we introduce a dummy item for the no-choice option, here, the no-choice option is implicitly considered: a no-choice is predicted if none of the recommendations are predicted to be chosen by the user (i.e., when the label of all the 12 recommendations are predicted as 0). Moreover, MNL assumes that a user, when presented with a set of 12 recommendations, can select only one item among them. While the ML, since is classifying each recommendation independently, can predict more recommendations to be chosen. To make the two models comparable, we modify ML so that if more than one recommendation is predicted to be chosen, we set as user choice the item with the highest prediction confidence.

5. Experimental Results

5.1. Choice Prediction Precision and Accuracy

We first compare our models in terms of precision and balanced accuracy scores. Precision is calculated for each choice set, and it is the ratio of the choice sets where the model has simulated the correct choice. Giving label 0 when a recommendation is not chosen, and label 1 when it is chosen, balanced accuracy measures the average of accuracy in predicting each label. Table 2 shows the precision and balanced accuracy scores calculated for all the predictions over the L time intervals. The shown metrics are the average values calculated over five repetition of the simulation. In the Table we also show the standard deviation of the metrics.

Clearly, ML outperforms MNL. Hence, one can conclude that the ML model is better at predicting individuals' choices. However, in general, the accuracy of both of the models is not very high. The reason for these small precision scores could be the inherent noise that exists in the data: humans are not consistently making choices. Moreover, if a user does not respond to a slate of recommendations (no-choice), we do not know whether the user did not like to choose any of the items, or she simply did not even see them. Finally, our prediction models are clearly limited and introduce specific biases to make the prediction problem solvable (e.g., the utility that drives the MNL model is a linear function of the selected features).

Table 2
Performance of MNL and ML models on the predictions of users' choices.

	Precision (std)	Balanced Accuracy (std)
MNL	0.11 (± 0.004)	0.13 (± 0.003)
ML	0.16 (± 0.006)	0.21 (± 0.006)

5.2. Choice Distribution Metrics

Here, we compare the CMs by analysing the distribution of the generated choices. The metrics here considered are:

1. Gini index: the choice diversity is measured using the *Gini index*, which is used in the literature to quantify item consumption inequality [23, 12, 13, 14, 24, 25]. A high Gini index indicates a low diversity of the choices. Gini index is close to 1 when there is a high inequality, and it is 0 when there is a perfectly uniform distribution across items [26].
2. Choice Coverage: Choice Coverage measures the fraction of the items that have been chosen (in the simulation) at least once by any user. We note that the number of items may change over time since some new items may be added to the catalogue at the beginning

of each time interval. We also note that while the Gini index quantifies how much the choices are uniformly distributed among the items in the catalogue, and it is sensitive to how many times an item is chosen, Choice Coverage measures the spread of the choices.

3. Shannon Entropy: is another measure of diversity and it is defined as follows:

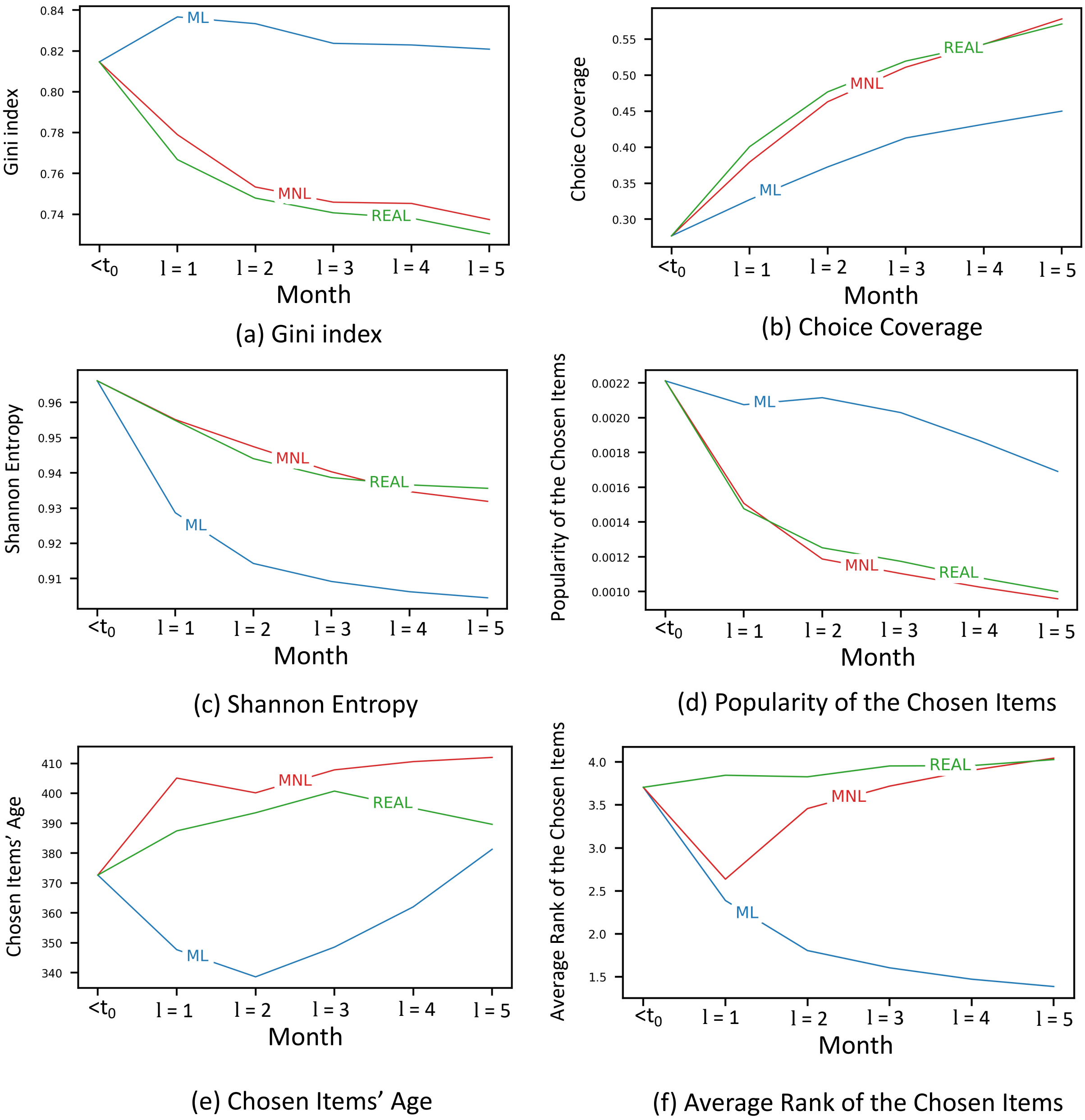
$$H = -\sum_{i=1}^{n} p_i log(p_i); \tag{4}$$

   where n ($n \leq |I|$) is the number of unique items that have been chosen at least once, p_i is the probability of choosing item i, estimated as the number of times the i-th item was chosen, divided by the total number of choices recorded. As the maximum value of H depends on the number of items n that have been chosen at least once, H is then normalised by dividing it by $log(n)$.
4. Popularity: is the average of the number of times the chosen items were actually chosen.
5. Chosen Items' Age: is the average (in days), on the chosen items, of the time passed from when the chosen items were first available in the catalogue.
6. Average Rank of the Chosen Items: measures the rank of the chosen items in the recommendation list.

Figure 2 shows the evolution of the considered metrics over the simulation intervals. We note that at the '$< t_0$' value on the x axis it is shown the metric calculated on the actual choices up to t_0. On the other values of the x axis ('$l = 1$', ..., '$l = 5$') are shown the metric value computed on the simulated choices from t_0 until the end of the corresponding interval. Hence, for instance, in Figure 2 (a), at point '$l = 4$' it is shown the Gini index calculated over the accumulated choices made in months 1, 2, 3 and 4. One could also show the metric values calculated over choices simulated within every single time interval; a similar, but less smooth, overall behaviour can be observed We opted to show the accumulated metrics to offer a clearer understanding of the evolution of the choices' distribution.

Moreover, to precisely quantify the differences between the simulation and real curves shown in Figure 2, in Table 3 we show the Root Mean Squared Error (RMSE): the data points on a simulation curve metric (MNL and ML) are compared to the data points on the REAL curve metric. For instance, the RMSE for the Gini index of the choices simulated by the MNL model (0.007), represents the difference between the "REAL" Gini index computed on the real choices and the Gini index of the choices simulated by MNL (over 5 simulation months). We note that this value is the average of the RMSE over the five simulation runs.

We first focus on the three metrics that show different forms of choice diversity: Gini index, Choice Coverage and Shannon Entropy. We note that the Gini index values of the choices simulated by the MNL model are more similar to those computed on the real choices in the data set ("REAL"), compared to the corresponding Gini index values computed for the choices simulated by ML. The Gini index values of the ML model are much larger than the Gini index values of the observed choices. This means that with ML, there is a significant concentration of the choices over a small set of items. The smaller Gini index obtained by MNL could be related to the stochastic nature of MNL; while the ML model predicts the label based on a learned probability threshold, the MNL model assumes that a target user, when receives a set



(a) Gini index

(b) Choice Coverage

(c) Shannon Entropy

(d) Popularity of the Chosen Items

(e) Chosen Items' Age

(f) Average Rank of the Chosen Items

Figure 2: Evolution of the considered choices' metrics on the choices simulated with MNL, CatBoost (ML), and on the real choices

of recommendations, uses a randomised CM for choosing the recommendations. In fact, with MNL, there is a higher chance for every recommended item to be chosen. Moreover, in the ML model, some of the item-related features (brand, popularity) may increase the probability of choosing a smaller set of items more often. For instance, the ML model gives high importance score to features such as brand, popularity and category of the recommended items. Hence, the chosen items are more likely to be of specific brands, or to belong to specific category, or to be popular. As a consequence, the diversity of the chosen items is smaller compared to MNL's produced choices. Similar observations can be done when considering the Choice Coverage and Shannon Entropy in Figure 2 (b) and (c).

The popularity of the chosen items (Figure 2 (d)) further signals that the choice simulations performed with MNL and ML are substantially different. The Popularity of choices generated by ML is much higher than the popularity of the true users' choices. Conversely, MNL obtains popularity values much more similar to those of the observed choices. For the age of the chosen items (Figure 2 (e)), again, MNL is shown to generate simulated choices with a closer age to that of the observed choices. In addition, Figure 2 (f) shows that while ML has a noticeable bias towards the highest ranked items, the position bias of MNL's simulated choices is similar to the position bias that the actual users have in their choices. In fact, we found that although both CMs score the rank of the recommended item as a valuable feature, the MNL's stochastic nature ultimately yields choices that mitigate this bias, which is instead very strong in the ML model.

Table 3
RMSE calculated on choice distribution metrics of CMs with regard to actual choice distribution.

	Gini index	Choice Coverage	Shannon Entropy	Chosen Items Popularity	Chosen Items Age	Average Rank of the Chosen Items
MNL	0.007	0.01	0.002	0.00005	15.16	0.57
ML	0.082	0.10	0.029	0.00076	41.15	2.22

Finally, the results of this comparison, which are summarised in Table 3, show that the MNL may be preferable to the ML when the goal is to measure global properties of the "distribution" of the choices. Moreover, when selecting the CM to be used in a simulation, one should also consider other differences between the two models. Firstly, MNL cannot be used to predict how many choices will be actually done for a given choice set. Conversely, while ML was adapted to simulate only one choice, it can easily generate multiple choices for a given choice set, since each item is independently predicted whether it will be chosen or not. Secondly, the stochastic nature of MNL may allow it to generate choices that are more diverse (See Figure 2), while ML is biased towards modelling choices for some items with specific features, e.g., brand, category, rank, and popularity.

6. Conclusion and Future Work

While some recent studies have used simulations to understand the long-term impact of RSs on users' choices, the validity of such simulations has also been criticised, because the accuracy of the adopted choice model was not assessed. In this paper, we consider two choice models (MNL and ML) and we measure their quality in reproducing the true observed choices. We have found that, considering classical accuracy metrics, which estimate how closely the predicted/simulated choices overlap the actual choices, the ML model performs better than the MNL one. However, MNL's simulated choices are much closer, than those produced by ML, to the actual choices in terms of their global distribution (the Gini index, Shannon Entropy, catalogue coverage). Hence, MNL seems to be more suitable in simulations, when the goal is to understand the global impact of the RS on the users' choice distribution, for instance, if the choices suffer from some particular bias, such as, the concentration bias.

There are still some limitations that need to be addressed in the future. First of all, we have tested the two CMs only with one data set. Hence, running the simulation with additional data sets is in order, to further validate the observations made in our initial analysis. In addition, while we have simulated the users' click-over recommendations, one can also imagine to consider purchases, because they are better indicators of real users' preferences and corresponding choices. Hence, in the future, simulations of both clicks and possible subsequent purchases will be conducted. Moreover, we plan to improve the predictive power of the simulation by modelling the number of times each user will ask for recommendations in a time interval, and other contextual variables (e.g., the intent of the user). Another interesting analysis relates to the comparison of the features that the two CMs find valuable and useful in the prediction of the choices. This can also help to find a better selection of predictive features. Finally, we plan to understand whether a CM, that is trained with the users' interactions with an RS, can be reused to model users' interactions with another RS. If so, one can use a validated CM to predict the long-term impact of not even yet deployed CMs on users' choices.

7. Acknowledgments

The authors would like to thank Pavel Kordik for his insightful ideas during the research and the experiments. They would also like to thank Petr Kasalický for his assistance in preparing the data set for the experiments.

References

[1] F. Ricci, L. Rokach, B. Shapira, Recommender systems: Techniques, applications, and challenges, Recommender Systems Handbook (2022) 1–35.

[2] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, L. Pizzato, Multistakeholder recommendation: Survey and research directions, User Modeling and User-Adapted Interaction 30 (2020) 127–158.

[3] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, L. Pizzato, Beyond personalization: Research directions in multistakeholder recommendation, arXiv preprint arXiv:1905.01986 (2019).

[4] D. Jannach, M. Jugovac, Measuring the business value of recommender systems, ACM Transactions on Management Information Systems (TMIS) 10 (2019) 1–23.

[5] M. Kunaver, T. Požrl, Diversity in recommender systems–a survey, Knowledge-Based Systems 123 (2017) 154–162.

[6] F. Huseynov, S. Y. Huseynov, S. Özkan, The influence of knowledge-based e-commerce product recommender agents on online consumer decision-making, Information Development 32 (2016) 81–90.

[7] S. Yao, Y. Halpern, N. Thain, X. Wang, K. Lee, F. Prost, E. H. Chi, J. Chen, A. Beutel, Measuring recommender system effects with simulated users, arXiv preprint arXiv:2101.04526 (2021).

[8] Z. Ying, C. Caixia, G. Wen, L. Xiaogang, Impact of recommender systems on unplanned purchase behaviours in e-commerce, in: 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), IEEE, 2018, pp. 21–30.

[9] P. Hosein, I. Rahaman, K. Nichols, K. Maharaj, Recommendations for long-term profit optimization, In 1st International Workshop on the Impact of Recommender Systems at RecSys (2019).

[10] N. Hazrati, F. Ricci, Recommender systems effect on the evolution of users' choices distribution, Information Processing & Management 59 (2022) 102766.

[11] N. Hazrati, F. Ricci, Simulating users' interactions with recommender systems, in: Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, 2022, pp. 95–98.

[12] D. M. Fleder, K. Hosanagar, Recommender systems and their impact on sales diversity, in: Proceedings of the 8th ACM conference on Electronic commerce, ACM, 2007, pp. 192–199.

[13] D. Fleder, K. Hosanagar, Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity, Management science 55 (2009) 697–712.

[14] Z. Szlávik, W. Kowalczyk, M. Schut, Diversity measurement of recommender systems under different user choice models, in: Fifth International AAAI Conference on Weblogs and Social Media, 2011.

[15] J. Zhang, G. Adomavicius, A. Gupta, W. Ketter, Consumption and performance: Understanding longitudinal dynamics of recommender systems via an agent-based simulation framework, Information Systems Research 31 (2020) 76–101.

[16] A. J. Chaney, B. M. Stewart, B. E. Engelhardt, How algorithmic confounding in recommendation systems increases homogeneity and decreases utility, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 224–232.

[17] D. Bountouridis, J. Harambam, M. Makhortykh, M. Marrero, N. Tintarev, C. Hauff, Siren: A simulation framework for understanding the effects of recommender systems in online news environments, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, ACM, 2019, pp. 150–159.

[18] D. Lee, K. Hosanagar, Impact of recommender systems on sales volume and diversity, Proceedings of the Thirty Fifth International Conference on Information Systems (2014).

[19] N. Hazrati, F. Ricci, The impact of recommender system and users' behaviour on choices' distribution and quality, in: International Workshop on Algorithmic Bias in Search and Recommendation, Springer, 2022, pp. 12–20.

[20] A. J. Chaney, Recommendation system simulations: A discussion of two key challenges, arXiv preprint arXiv:2109.02475 (2021).

[21] J. Feldman, D. J. Zhang, X. Liu, N. Zhang, Customer choice models vs. machine learning: Finding optimal product displays on alibaba, Operations Research 70 (2022) 309–328.

[22] A. V. Dorogush, V. Ershov, A. Gulin, Catboost: gradient boosting with categorical features support, arXiv preprint arXiv:1810.11363 (2018).

[23] C. Matt, T. Hess, C. Weiß, The differences between recommender technologies in their impact on sales diversity, ICIS (2013).

[24] D. Lee, K. Hosanagar, How do recommender systems affect sales diversity? a cross-category investigation via randomized field experiment, Information Systems Research 30 (2019) 239–259.

[25] P. Adamopoulos, A. Tuzhilin, P. Mountanos, Measuring the concentration reinforcement bias of recommender systems, rN (i) 1 (2015) 2.

[26] R. Dorfman, A formula for the gini coefficient, The review of economics and statistics (1979) 146–149.