Flexible Generation of Preference Data for Recommendation Analysis

Simone Mungari
University of Calabria
ICAR-CNR
Revelis s.r.l.
Rende, Italy
simone.mungari@unical.it

Erica Coppolillo
University of Calabria
ICAR-CNR
Rende, Italy
erica.coppolillo@unical.it

Ettore Ritacco
University of Udine
Udine, Italy
ettore.ritacco@uniud.it

Giuseppe Manco
ICAR-CNR
Rende, Italy
giuseppe.manco@icar.cnr.it

Abstract

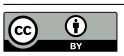
Simulating a recommendation system in a controlled environment, to identify specific behaviors and user preferences, requires highly flexible synthetic data generation models capable of mimicking the patterns and trends of real datasets. In this context, we propose HyDRA, a novel preferences data generation model driven by three main factors: user-item interaction level, item popularity, and user engagement level. The key innovations of the proposed process include the ability to generate user communities characterized by similar item adoptions, reflecting real-world social influences and trends. Additionally, HyDRA considers item popularity and user engagement as mixtures of different probability distributions, allowing for a more realistic simulation of diverse scenarios. This approach enhances the model's capacity to simulate a wide range of real-world cases, capturing the complexity and variability found in actual user behavior. We demonstrate the effectiveness of HyDRA through extensive experiments on well-known benchmark datasets. The results highlight its capability to replicate real-world data patterns, offering valuable insights for developing and testing recommendation systems in a controlled and realistic manner. The code used to perform the experiments is publicly available: https://github.com/flexibledatageneration/HYDRA.

CCS Concepts

• **Information systems** → **Collaborative filtering**; **Recommender systems**; • **Computing methodologies** → **Simulation support systems**; *Uncertainty quantification.*

Keywords

Data Generation, Benchmarking, Recommendation, Probabilistic Modeling



KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737398

ACM Reference Format:
Simone Mungari, Erica Coppolillo, Ettore Ritacco, and Giuseppe Manco. 2025. Flexible Generation of Preference Data for Recommendation Analysis. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737398

1 Introduction

Ensuring clean and reliable interaction data is a major concern in the context of recommendation [19, 42], social network analysis [2, 37], and machine learning in broad sense [22]. As algorithms become more powerful and sophisticated, the demand for reliable benchmarking studies to evaluate and compare their capabilities across various perspectives and scenarios is growing [45]. However, the availability of benchmark open-source datasets is limited since large industrial companies generally do not release their vast amounts of proprietary data to the public. As a result, the need for dependable datasets is more urgent and valuable than ever.

Traditionally, evaluation takes place using a variety of publicly available real-life datasets. Notable examples are Movielens [17] and Netflix [6], Lastfm [7] and Yahoo! Music [14], Epinions [40], and Amazon [20]. However, these real-life benchmark datasets exhibits several limitations. First, they tend to focus on specific domains, limiting the generalizability of findings across diverse applications. Additionally, many of these datasets lack the scale needed to assess the performance of algorithms in large, real-world settings, which can lead to inaccurate performance assessments. Lastly, intrinsic biases present in these datasets, such as filter bubbles or echo chambers resulting from feedback loops in operational environments [1, 16, 23], can distort the evaluation of recommender systems, leading to misleading conclusions about their effectiveness.

Standardized datasets can help mitigate these issues by offering a broader, unbiased, and scalable framework for testing and improving recommendation algorithms. To obtain them, crowd-sourcing can be considered as a viable option [25, 26, 33, 55]. However, recruiting real users is generally expensive and time-consuming. Besides, user's behavior data poses several privacy concerns and challenges that prevent actual recruitinig strategies and in general the disclosure and availability of preferences. Indeed, a more practical

and comprehensive solution to fill the aforementioned gaps and address the scarcity and inadequacy of real data consists of generating data synthetically.

In this regard, some methods try to replicate the characteristics of real-world data by learning their distribution [3, 32], by performing augmentation [47] or compression [51]. These methods learn directly from the ground truth, thus providing a reliable replication of real-world data. Nevertheless, such approaches are bound to the underlying originary scenarios, thus lacking flexibility. On the other hand, generative probabilistic frameworks represents viable alternatives for producing synthetic datasets [4, 27, 38, 44, 48]. However, current synthetic generation methods of preference data face significant limitations, including narrow customization and unrealistic distributions. These methods struggle to accurately reflect nuanced user preferences and behaviors. Additionally, the data often lacks variability, leading to overly simplistic patterns that fail to capture the complexity of real-world user interactions and network structures.

In this paper, we present HyDRA (Hyper-Parametric Data generation for Recommendation Analysis), a flexible generative probabilistic framework for generating preference data. The theoretical foundation of our method relies on three main components: modeling *user beliefs, leanings, and attitude to engage and interact*; assessing *item exposure and potential popularity*; and evaluating the likelihood that *users' and items' inherent features match*. From this theoretical model, we parameterize the framework by introducing multiple control factors to leverage a fully controllable generation process. Our method notably allows for: (*i*) regulating user-item interactions; (*ii*) freely tweaking the underlying data distributions; and (*iii*) faithfully replicating the characteristics of pre-existing datasets. We perform an extensive evaluation of HyDRA, assessing its efficacy under the aforesaid perspectives.

The rest of the paper is structured as follows. Section 2 provides an overview of state-of-the-art synthetic generation procedures. In Section 3, we formally define our methodology and discuss its theoretical foundation. Section 4 describes the instantiation of HyDRA based on insights from the modeling discussion. We present the experimental results in Section 5, demonstrating the flexibility of HyDRA. Finally, Section 6 offers pointers for future developments.

2 Related Work

Given the scarcity of high-quality real-world datasets for recommender systems, there has been considerable effort in the literature to generate synthetic data. In this broad context, we specifically concentrate on producing realistic user-item interactions. This task poses several challenges, including the necessity to replicate the topological characteristics of real-world scenarios [4, 15, 27, 32]. The present literature covers a wide range of different approaches, which we outline below.

Data Augmentation and Condensation. Data augmentation consists of expanding an existing dataset while preserving its structural properties. This valuable task has been extensively studied in the literature. Vo and Soh [47] propose a framework based on Variational Autoencoders (VAEs) for generating novel items that users will probably interact with. Belletti et al. [5] propose an expanding approach based on an adaption of Kronecker Graphs. More recent approaches have also explored the adoption of Large Language Models in this regard [34].

In an opposite perspective, condensation refers to the task of compressing the original data while still maintaining their properties. Wu et al. [51] propose a novel framework for condensing the original dataset while addressing the long-tail problem with a reasonable choice of false negative items. Jin et al. [24] propose a strategy aimed at compacting a graph preserving its features for classification tasks. The process is performed by a gradient matching loss optimization and a strategy to condense node features and structural information simultaneously.

Semi-Synthetic Generation. Differently from data augmentation methods, these approaches involve models learning directly from a ground-truth dataset to generate a fully synthetic one. For instance, Bobadilla and Gutierrez [8] introduce a Wasserstein-GAN architecture [3] for this purpose. The resulting synthetic dataset exhibits similar patterns and distributions of users, items, and ratings compared to the real dataset used in the experimental evaluation.

A specific sub-field of this research area focuses on generating synthetic data to protect sensitive real-world information [30]. These methods are typically applied in fields such as finance and healthcare, and in general in any context where data privacy is crucial [29, 30]. For instance, in [43], the framework alters a subset of real data values to produce a new semi-synthetic dataset. Similarly, in [44], the model starts by generating a dense user-item matrix using a probabilistic matrix factorization approach based on Gaussian distributions, further masking some user preferences.

Probabilistic Models. The generation of synthetic data is often performed by adopting probabilistic approaches. Different kinds of distributions have been explored for user-item content sampling, such as the typical Gaussian [50], the Bernoullian [52], the Dirichlet and Chi-square distributions [46]. Among the others, Hu et al. [21] propose a Bayesian Generative Framework and modeling procedure based on Gibbs Sampling for binary count data with side information.

Simulation-based approaches. Driven by the objective of integrating generation and recommendation, other methods aim at simulating realistic interactions for inactive users [48, 49, 54]. In [39], the intent is to generate a binary preference matrix with five different topics spanning from the Far Left to the Far Right of the political spectrum. Their main limitation is the lack of a study of the generated user and item distributions. Chaney et al. [9] extend the model proposed in [41] for allowing multiple interactions for the same user. The histories are generated by sampling from a (noisy) utility matrix which represents the actual preferences of users.

Table 1 reports the portion of methods presented in this section that generate synthetic datasets from scratch, thus presenting similarities with our proposal. The table summarizes the differences with our methodology.

3 Data Modeling

The foundational intuition behind our proposal is to jointly model the observations of users and items in a preference dataset along with intrinsic information within the data, represented as data statistics. While the latter are essentially hidden information embedded within the list of user-item pairs, they could be distorted by

Model	No training	Realistic	Flexibile			Source Code
			Distributions	Interactions	Topics	
HyDRA (ours)	✓	✓	✓	✓	✓	✓
Tso et al. [46]	✓	✗	✗	✗	✓	✗
Smith et al. [44]	✓	✓	✗	✓	✗	✗
Chaney et al. [9]	✗	✗	✗	✗	✗	✗
Zhang et al. [52]	✓	✗	✗	✓	✓	✗
Ribeiro et al. [39]	✓	✓	✗	✗	✓	✓

Table 1: Comparison between the proposed framework and the state-of-the-art models which generate synthetic data from scratch. The column "No training" indicates whether the method allows to generating interactions without a training phase; "Realistic" refers to the conformity of the generated data to the typical properties of real-world data distributions (e.g. long-tailed); "Source Code" denotes the availability of the code for the experiments; finally, the column "Flexible" refers to the adaptability of the approach in terms of (*i*) fitting a specific distribution, (*ii*) ad-hoc tuning the synthetic user-item interactions, and (*iii*) introducing a given number of topics, further regulating the users interest accordingly.

a probabilistic model if not explicitly considered. From this insight, considering the number of items adopted by each user and the number of users for each item as key statistics, it follows that HyDRA is governed by three main components: User-Item Matching, User Engagement Level, and Item Popularity.

User-Item Matching. This factor summarizes the level of affinity between user preferences and item characteristics. Typically, users select items that meet and satisfy their tastes and needs. We believe this phenomenon is not absolute; rather, it depends on other two factors. First, a user does not have visibility of the entire set of items, making it unrealistic for a choice to be driven solely by preferences. Secondly, a user-item pair will be observable in the preference matrix only if both the user and the item have achieved a certain level of exposure: user engagement level and item popularity, respectively.

User Engagement Level. The probability of observing a user is a consequence of their affiliation with interaction platform. Thus, the generative model must define a distribution capable of simulating the frequency of user occurrences in the dataset.

Item Popularity. Previous studies in the field of recommendation systems [10, 56] have shown that user behavior is significantly influenced by the phenomenon of popularity bias. Users tend to be influenced by global trends, and an item is often chosen because of its popularity, regardless of its actual affinity with various users. This reinforces the principle that an item's value increases as it is adopted by more individuals [31].

Let $U = \{1, \dots, n\}$ and $I = \{1, \dots, m\}$ be the set of users and items, respectively. Let $\mathcal{D}$ be the dataset of all user preferences, expressed as pairs (u, i), where $u \in U$ and $i \in I$. From $\mathcal{D}$, we can derive the vectors $\mathbf{z} \in \mathbb{N}^n$ and $\mathbf{y} \in \mathbb{N}^m$, such that $z_u = |\{i : (u,i) \in \mathcal{D}\}|$ represents the level of engagement of user u and $y_i = |\{u : (u,i) \in \mathcal{D}\}|$ represents the popularity of item i. Formally, given a set ν of parameters governing the preferences, HyDRA defines the likelihood of the observed tuple $(\mathcal{D}, \mathbf{z}, \mathbf{y}, \nu)$ by utilizing an implicit feedback approach. This means we define a random variable $r \in \{0, 1\}$ to indicate whether a pair (u, i) exists in the dataset. Consequently, we have:

$$P(\mathcal{D}, \mathbf{z}, \mathbf{y}, \nu|\Theta) = P(\nu)\, P_\nu(\mathcal{D}|\mathbf{z}, \mathbf{y})\, P(\mathbf{z}, \mathbf{y}|\Theta)\,, \tag{1}$$

where

$$P_\nu(\mathcal{D}|\mathbf{z}, \mathbf{y}) = \prod_{u,i \in \mathcal{D}} P_\nu(r = 1, u, i|z_u, y_i) \prod_{u,i \notin \mathcal{D}} [1 - P_\nu(r = 1, u, i|z_u, y_i)] \tag{2}$$

and $P(\mathbf{z}, \mathbf{y}|\Theta) = P(\mathbf{z}|\Theta)\, P(\mathbf{y}|\Theta)$ represents the probability of observing the frequencies $\mathbf{z}$ and $\mathbf{y}$. As mentioned, the probabilities $P_\nu(r, u, i|z_u, y_i)$, $P(\mathbf{z}|\Theta)$ and $P(\mathbf{y}|\Theta)$ represent the User-Item Matching, User Engagement Level, and Item Popularity components, respectively. In particular, $P_\nu(r, u, i|z_u, y_i)$ can be factored as

$$P_\nu(r = 1, u, i|z_u, y_i) = P_\nu(r = 1|u, i)\, P(u|z_u)\, P(i|y_i)\,, \tag{3}$$

assuming independence between the pairs (u, z_u) and (i, y_i).

In addition, we model the distributions governing User Engagement Level and Item Popularity as a mixture of components:

$$P(\mathbf{z}|\Theta) = \sum_{k=1}^{K} \pi_k \prod_{u \in U} P_{\theta_k}(z_u)\,; \quad P(\mathbf{y}|\Theta) = \sum_{h=1}^{H} \psi_h \prod_{i \in I} P_{\vartheta_h}(y_i). \tag{4}$$

Here, $\{1, \dots, K\}$ and $\{1, \dots, H\}$ represent partitions over two probabilistic spaces containing the chosen distributions for the User Engagement Level and the Item Popularity, respectively, whose parameters are θ_k and ϑ_h, with $k \in \{1, \dots, K\}$ and $h \in \{1, \dots, H\}$. Within the equations, the probabilities π_k and ψ_h indicate mixing distributions. Combining Equations 2, 3 and 4, we can rewrite Equation 1 as:

$$P(\mathcal{D}, \mathbf{z}, \mathbf{y}, \nu|\Theta) = P(\nu)\, P_\nu(\mathcal{D}|\mathbf{z}, \mathbf{y}) \sum_{k=1}^{K} \sum_{h=1}^{H} \pi_k \psi_h \prod_{u \in U} P_{\theta_k}(z_u) \prod_{i \in I} P_{\vartheta_h}(y_i) \tag{5}$$

3.1 User-Item Matching

Drawing inspiration from [9] and [44], we instantiate $\nu = \{\nu_u, \nu_i\}$ as the set of the model parameters $\nu_u = \{\rho_u, \mu_u^\rho\}_{u \in U}$ and $\nu_i = \{\alpha_i, \mu_i^\alpha\}_{i \in I}$, relative to users and items, that govern the probability of observing preferences. First, we define *User-Item Matching* by exploiting latent factors:

$$P_\nu(r = 1|u, i) = \lambda \cdot \rho_u^\top \alpha_i\,, \tag{6}$$

where ρ_u and α_i are latent vectors of size f, representing instantiations over a domain of features that characterize user preferences and item properties. The λ component is a regularization term based on the likelihood of observing an interaction in the specific domain of interest, regardless of the users/items involved, thus modeling the intrinsic expected density of $\mathcal{D}$.

Both ρ_u and α_i are modeled considering a hierarchical process based on Dirichlet distributions. The first step draws the prior probabilities μ_u^ρ and μ_i^α for each user and item over the latent space, while the second step draws their posterior distributions. For users, we define:

$$\rho_u \sim \text{Dirichlet}(10\, \mu_u^\rho) \quad \text{and} \quad \mu_u^\rho \sim \text{Dirichlet}(\mathbf{1}_f)\,, \tag{7}$$

where $\mathbf{1}_f$ denotes a vector composed of f ones. The $\mathbf{1}_f$ parameter of the Dirichlet distribution that models μ_u^ρ allows the prior probability of users to evenly distribute in the latent space. This implies that a user may uniformly assume any kind of role, ranging from a *generalist*, who is interested in all features within the latent space, to a *specialist*, who focuses their attention on a specific topic. However, once a point is selected, the user is likely to exhibit weighted preferences for a subset of features. This behavior is modeled by the Dirichlet distribution with parameters $10\,\mu_u^\rho$, where μ_u^ρ represents the center of the user's preferences. The scaling factor of 10 encourages users to explore a broader range of latent factors, counteracting the tendency of the Dirichlet distribution to prioritize vertices of the latent space when its parameters are in the interval $[0, 1]$, such as in the case of μ_u^ρ.

A notable difference in our approach compared to [9] is in the sampling of μ_u^ρ that is performed for each user. This means that all generated users are evenly mapped in the latent space, thus potentially exhibiting a wide range of behaviors, from generalist to specialist. In contrast, single sampling leads to user stereotyping, where users cluster around a shared behavioral profile that acts as an unpredictable center of mass. Figure 10 in Appendix A illustrates this difference.

Similarly to ρ_u, for each item i we define α_i as:

$$\alpha_i \sim \text{Dirichlet}(0.1\,\mu_i^\alpha) \quad \text{and} \quad \mu_i^\alpha \sim \text{Dirichlet}(\mathbf{100}_f)\,, \tag{8}$$

where $\mathbf{100}_f$ denotes a vector composed of f entries, each equal to 100. In this case, we constrain μ_i^α to be at the center of the space. In our opinion, if users have the freedom to explore the entire latent space, items should be confined to a limited subset of topics. Therefore, we model α_i according to a Dirichlet distribution with parameters μ_i^α scaled down by a factor of 0.1. This approach emphasizes the vertex prioritization phenomenon of the Dirichlet distribution, making it highly unlikely for items to encompass all latent factors. Again, Figure 11 in Appendix A illustrates this aspect.

In summary, the probability of observing a preference is conditioned by a global prior component, defined as:

$$P(v) = \prod_{u\in U} P(v_u) \prod_{i\in I} P(v_i)\,, \tag{9}$$

where:

$$P(v_u) = \text{Dirichlet}(\rho_u; 10\,\mu_u^\rho)\,\text{Dirichlet}(\mu_u^\rho; \mathbf{1}_f) \tag{10}$$

$$P(v_i) = \text{Dirichlet}(\alpha_i; 0.1\,\mu_i^\alpha)\,\text{Dirichlet}(\mu_i^\alpha; \mathbf{100}_f)\,. \tag{11}$$

Generating partitions. The proposed protocol can be easily adapted to generate topical user *communities* and item *categories*. Specifically, we can partition $U = \{U_1, U_2, \ldots, U_c\}$, and $I = \{I_1, I_2, \ldots, I_g\}$ to represent populations of users and items that exhibit strong internal homophily and external heterogeneity across the latent space. This ensures that HyDRA can simulate realistic topical connections between groups of similar users who share preferences for a subset of features and groups of items characterized by those features. To disable a feature in a user community c (or item category g), it is sufficient to set, for each user in c (or each item in g), the related component of μ_u^ρ (or μ_i^α, respectively) to an arbitrarily small constant ϵ. Users within the community will be evenly distributed around the remaining latent factors, while items will maintain their uneven distribution across these factors.

In terms of likelihood, the proposed partitions can be modeled by a suitable adaptation of Equation 9, to include a mixture of priors over the μ_u^ρ and μ_i^α components.

3.2 User Engagement and Item Popularity

An additional element of flexibility can be achieved by appropriately modeling the factors of User Engagement Level and Item Popularity, which in Equation 5 act as regularization terms for User-Item Matching. Empirical evidence shows that engagement and popularity typically adhere to Long-Tail distributions [11, 12, 35]. As aforesaid, our approach models these factors as mixtures of such distributions: P_{θ_k} for users and P_{ϑ_h} for items, with $k \in \{1, \ldots, K\}$ and $h \in \{1, \ldots, H\}$. The distributions we focused on in our modeling are show in Table 2.

Distribution	Parameters θ/ϑ	$f(x)$	C
Power-Law	α	$x^{-\alpha}$	$(\alpha-1)x_{\min}^{\alpha-1}$
Power-Law with Exponential cut-off	α, λ	$x^{-\alpha}e^{-\lambda x}$	$\frac{\lambda^{1-\alpha}}{\Gamma(1-\alpha, \lambda x_{\min})}$
Exponential	λ	$e^{-\lambda x}$	$\lambda e^{\lambda x_{min}}$
Stretched Exponential	λ, β	$x^{\beta-1}e^{-\lambda x^\beta}$	$\beta\lambda e^{\lambda x_{\min}^\beta}$
Log-Normal	μ, σ	$\frac{1}{x}\exp\left[-\frac{(\ln x-\mu)^2}{2\sigma^2}\right]$	$\sqrt{\frac{2}{\pi\sigma^2}}\left[\text{erfc}\left(\frac{\ln x_{\min}-\mu}{\sqrt{2}\sigma}\right)\right]^{-1}$

Table 2: List of Long-Tail distributions employed. The complete form is $P(x) = C\,f(x)$, such that $\int_{x_{min}}^{\infty} C\,f(x)dx = 1$.

In practice, the modeling is based on the assumption that observations $\mathbf{z}$ and $\mathbf{y}$ adhere to long-tailed distributions, in a two step process devised as follows:

$$\begin{aligned} k &\sim \text{Discrete}(\pi) & h &\sim \text{Discrete}(\psi) \\ z_u &\sim P_{\theta_k},\ u \in U & y_i &\sim P_{\vartheta_h},\ i \in I \end{aligned} \tag{12}$$

Here, $P_{\theta_k}(z_u)$ represents the probability of observing the frequency z_u relative to user u. The resulting mixture of distributions enables adaptation to different probability shapes, thus resulting a more faithful representation of the characteristics of real-world observations.

The final component in modeling the User Engagement is the probability of observing user u (respectively, item i for Item Popularity) within a preference, which is proportional to its frequency[1]:

$$\begin{aligned} P(u|z_u) &= Bernoulli(\varrho_u) & P(i|y_i) &= Bernoulli(\varphi_i) \\ \varrho_u &\sim Beta'(z_u/m, \sigma_u) & \varphi_i &\sim Beta'(y_i/n, \sigma_i) \end{aligned} \tag{13}$$

Notably, the engagement level z_u of a user is crucial for both the component $\prod_u P_{\theta_k}(z_u)$ and the component $P(u|z_u)$. The first term represents the global likelihood of observing the degree distribution of users. By modeling it as a long-tail distribution, this probability tends to favor realistic situations where many users with low occurrence in a few highly frequent users. The second term, however, encourages the selection of frequent users among those generated when creating user-item pairs. A similar reasoning can be done focusing on the items.

[1]Here, $Beta'(\mu, \sigma)$ represents an alternative parametrization of the $Beta(\alpha, \beta)$ distribution based on the specification of mean and variance, as suggested in [9].

3.3 Inference and Estimation

The above formalization relies on a parameter set Θ that can be readily estimated from real-world data. More formally, given $\hat{\mathcal{D}} = \{\mathcal{D}, \mathbf{z}, \mathbf{y}, \nu\}$, we aim at estimating the optimal parameter set Θ. To achieve this, we employ a variational approach by defining $Q(h, k|\hat{\mathcal{D}}, \Theta)$ as a proposal probability distribution. Then, the following inequality holds:

$$\begin{aligned}\log P(\hat{\mathcal{D}}|\Theta) \geq \log P_\Theta(\nu) + \log P_\nu(\mathcal{D}|\mathbf{z}, \mathbf{y}) \\ + \sum_{k=1}^{K}\sum_{h=1}^{H} Q(h, k|\hat{\mathcal{D}}, \Theta)\Bigg\{ \log(\pi_k\, \psi_h) \\ + \sum_{u\in U} \log P_{\theta_k}(z_u) + \sum_{i\in I} \log P_{\vartheta_h}(y_i)\Bigg\} \\ - \sum_{k=1}^{K}\sum_{h=1}^{H} Q(h, k|\hat{\mathcal{D}}, \Theta) \log Q(h, k|\hat{\mathcal{D}}, \Theta)\,.\end{aligned} \tag{14}$$

Within the inequality, the lower bound is characterized by the log-likelihood of the preferences $\mathcal{D}$, the expected log-likelihood of the frequencies $\mathbf{z}$ and $\mathbf{y}$, and the cross-entropy of the proposal distribution Q. Specifically, the latter serve as a regularization component, indicating that in the absence of additional information, all probability distributions are considered equally likely. It can be shown that the optimal Q, given the other components, can be expressed as:

$$Q(h, k|\hat{\mathcal{D}}, \Theta) = \frac{\pi_k\, \psi_h \prod_u P_{\theta_k}(z_u) \prod_i P_{\vartheta_h}(y_i)}{\sum_{k=1}^{K}\sum_{h=1}^{H} \pi_k\, \psi_h \prod_u P_{\theta_k}(z_u) \prod_i P_{\vartheta_h}(y_i)}\,. \tag{15}$$

Using the optimal Q, we can then formulate a loss component over two subsets $S \subseteq \mathcal{D}$ and $\bar{S} \subseteq U \times I \setminus \mathcal{D}$, related to the other model parameters:

$$\begin{aligned}\ell(\Theta|\hat{S}) = -\log P_\Theta(\nu) \\ - \sum_{u,i\in S} \log\left(P_\nu(r=1|u,i)\, P(u|z_u)\, P(i|y_i)\right) \\ - \sum_{u,i\in \bar{S}} \log\left(1 - P_\nu(r=1|u,i)\, P(u|z_u)\, P(i|y_i)\right) \\ - \sum_{u\in U} \log\left(P_\Theta(\varrho_u)\right) - \sum_{i\in I} \log\left(P_\Theta(\varphi_i)\right) \\ - \mathbb{E}_{h,k\sim Q}\left[\log(\pi_k\, \psi_h) + \sum_{u\in U} \log P_{\theta_k}(z_u) + \sum_{i\in I} \log P_{\vartheta_h}(y_i)\right].\end{aligned} \tag{16}$$

where $\hat{S} = \{S, \bar{S}, \mathbf{z}, \mathbf{y}, \nu\}$ and $\lambda_\Theta = \exp \upsilon$.

An Expectation-Maximization scheme for parameter estimation can thus be devised by combining the alternating computation of Q with gradient descent, as outlined in Algorithm 1.

Algorithm 1 Parameter Estimation

Input: $\mathcal{D}$, ν
Output: Θ
Initialize $\Theta^{(0)}$ randomly
Compute $\mathbf{z}$ and $\mathbf{y}$ from $\mathcal{D}$
$\hat{\mathcal{D}} \leftarrow \{\mathcal{D}, \mathbf{z}, \mathbf{y}, \nu\}$
$t \leftarrow 0$
while Θ has not converged **do**
  Compute $Q(h, k|\hat{\mathcal{D}}, \Theta^{(t)})$ {Expectation step, Eq. 15}
  Sample a batch $S \subseteq \mathcal{D}$ of preferences
  Sample a batch $\bar{S} \subseteq U \times I \setminus \mathcal{D}$ of negative pairs
  Update $\Theta^{(t+1)}$ by gradient descent {Maximization step}
    $\nabla_\Theta \ell\left(\Theta^{(t)}|S, \bar{S}, \mathbf{z}, \mathbf{y}, \nu\right)$ {Eq. 16}
  $t \leftarrow t + 1$
end while

4 Synthetic Data Generation

The probabilistic modeling framework outlined so far enables the definition of a generative stochastic process for preferences. The general structure of the process is detailed as follows:

Sampling Process

1: Draw user distribution choice $k \sim \text{Discrete}(\pi)$
2: Draw item distribution choice $h \sim \text{Discrete}(\psi)$
3: For $u \in U$:
 3.1: Draw $\rho_u \sim \text{Dirichlet}(10\,\mu_u^\rho)$, where $\mu_u^\rho \sim \text{Dirichlet}(\mathbf{1}_f)$
 3.2: Draw $\varrho_u \sim \textit{Beta}'(z_u/m, \sigma_u)$, where $z_u \sim P_{\theta_k}$
4: For $i \in I$:
 4.1: Draw $\alpha_i \sim \text{Dirichlet}(0.1\,\mu_i^\alpha)$, where $\mu_i^\alpha \sim \text{Dirichlet}(\mathbf{100}_f)$
 4.2: Draw $\varphi_i \sim \textit{Beta}'(y_i/n, \sigma_i)$, where $y_i \sim P_{\vartheta_h}$
5: For $u \in U$ and $i \in I$:
 5.1: If $\text{Bernoulli}\left(\lambda \cdot \rho_u^\top \alpha_i \cdot \varrho_u \cdot \varphi_i\right)$ then:
  5.1.1: Generate (u, i)

Taking inspiration from Equation 3, the first step of the algorithm is to select the distributions $k \in \{1, \ldots, K\}$ and $h \in \{1, \ldots, H\}$ that govern User Engagement Level and Item Popularity, respectively, through two Discrete distributions (lines 1-2). Subsequently, the process characterizes users and items (lines 3-4), by (a) choosing the latent features (μ_u^ρ, ρ_u) and (μ_i^α, α_i) ; (b) sampling Engagement Level and the Popularity from the chosen distribution k and h; devising their probability of occurrence $P(u|z_u)$ and $P(i|y_i)$, as indicated in Equation 13. Finally, in line 5, leveraging Equations 6 and 5, a Bernoulli trial is conducted for each user $u \in U$ and each item $i \in I$. The User-Item Matching, i.e., the probability of generating the user-item pair (u, i), is given by the combination of the sampled components. If the trial succeeds, the pair (u, i) then is generated.

HyDRA. Based on the general schema described so far, we tweak the probabilistic generative model to adapt the synthetic generation of datasets of implicit preferences for recommendation systems. The resulting algorithm, HyDRA offers extensive control options, enabling the simulation of various user behaviors and item characteristics. The pseudo-code of HyDRA is shown in Algorithm 2. In the following, we discuss its control options.

Generation of communities and categories. Manipulating the priors of the Dirichlet distributions allows for the generation of user communities and item categories. Given a partition of U and a partition of I, lines 3 and 4 of the Sampling Process can be extended by filtering the hyper-parameters $\mathbf{1}_f$ and $\mathbf{100}_f$. By utilizing a custom function, referred to as `FilterPrior` (lines 6 and 13 in HyDRA), which maps elements of the partition to distinct subsets

Algorithm 2 HyDRA

1: **Input:** user set U and item set I; user communities $U_1, \dots, U_c$ and item categories $I_1, \dots, I_g$ with associated features; Distribution priors π, ψ; number of latent features f and disabling factor ε; noise weight δ and prior χ, density control weights λ, ζ, ξ, minimum frequency τ; variability of the Beta' distributions β, σ_u, and σ_i.
2: **Output:** $\mathcal{D}$
3: Draw user distribution $k \sim \text{Categorical}(\pi)$
4: Draw item distribution $h \sim \text{Categorical}(\psi)$
5: **for** $s \in \{1, \dots, c\}$ and $u \in U_s$ **do**
6: $\hat{f} = \texttt{FilterPrior}(\mathbf{1}_f, s, \varepsilon)$ {Generate communities}
7: Sample user hyper-parameters $\mu_u^{\rho} \sim \text{Dirichlet}(\hat{f})$
8: Sample user latent factors $\rho_u \sim \text{Dirichlet}(10\,\mu_u^{\rho})$
9: Draw the user interactions $z_u \sim P_{\theta_k}$
10: Define *User Engagement Level* $P(u|z_u)$ as $\varrho_u \sim \textit{Beta}'(z_u/m, \sigma_u)$
11: **end for**
12: **for** $s \in \{1, \dots, g\}$ and $9 \in I_s$ **do**
13: $\hat{f} = \texttt{FilterPrior}(\mathbf{100}_f, s, \varepsilon)$ {Generate categories}
14: Sample item hyper-parameters $\mu_i^{\alpha} \sim \text{Dirichlet}(\hat{f})$
15: Sample item latent factors $\alpha_i \sim \text{Dirichlet}(0.1\,\mu_i^{\alpha})$
16: Draw the item interactions $y_i \sim P_{\vartheta_h}$
17: Define the *Item Popularity* $P(i|y_i)$ as $\varphi_i \sim \textit{Beta}'(y_i/n, \sigma_i)$
18: **end for**
19: $\mathcal{D} \leftarrow \emptyset$
20: Sample $\omega \sim \text{Beta}'(\chi, \beta)$ {Noise factor}
21: **for** $u \in U$ **do**
22: $\mathcal{D}_u \leftarrow \emptyset$
23: **while** $|\mathcal{D}_u| < \tau$ **do**
24: **for** $i \in I$ **do**
25: Define the *User-Item Matching* $T_{u,i} \leftarrow \rho_u^{\top}\,\alpha_i$
26: $T_{u,i} \leftarrow \delta\, T_{u,i} + (1-\delta)\,\omega$ {Adding noise}
27: $T_{u,i} \leftarrow \lambda \cdot T_{u,i} \cdot \varrho_u^{\zeta} \cdot \varphi_i^{\xi}$ {Densification/Sparsification}
28: Sample $\eta_{u,i} \sim \textit{Beta}'(T_{ui}, \beta)$ {Adding variability}
29: **if** Bernoulli$(\eta_{u,i})$ **then**
30: $\mathcal{D}_u \leftarrow \mathcal{D}_u \cup \{(u,i)\}$
31: **end if**
32: **end for**
33: **end while**
34: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_u$
35: **end for**

of latent features, it becomes possible to generate well-defined communities and categories. This is achieved by specializing them on the features whose prior is not set to ε, an arbitrarily small constant. This approach ensures that the generated data reflects the desired structural properties and relationships within the latent space.

Noise injection. The User-Item Matching component represents the degree of appreciation a user has for an item. However, real-world navigation through a catalog of items often involves some degree of randomness. To simulate this phenomenon, HyDRA introduces noise into the matching process by incorporating a noise term, ω, into Equation 6 using a linear combination parameterized by $0 \le \delta \le 1$ (line 26). The value of ω is sampled (line 20) from a Beta distribution, parametrized to guarantee mean (the hyper-parameter χ) and variance β.

User minimum frequency. HyDRA ensures that every user in U is involved in at least a minimum number of observations, denoted as τ. Line 23 of the algorithm mandates that for each user u, the generation of pairs (u, i) with $i \in I$ is repeated until the user's preference list, $\mathcal{D}_u$, has a size greater than or equal to τ. This guarantees sufficient representation of each user in $\mathcal{D}$.

Density manipulation. To adjust the density of the preference matrix generated from the user-item observations (line 27), HyDRA introduces a coefficient λ and two exponents ζ and ξ. Specifically, as already mentioned, λ scales the User-Item Matching, either amplifying ($\lambda > 1$) or diminishing ($\lambda < 1$) the overall probability of occurrence. In particular, higher values of λ increase the likelihood of generating more successes in the Bernoulli trial (line 29). The exponents ζ and ξ adjust the distributions of users and items, respectively. Higher values amplify the importance of user engagement and item popularity over the feature matching. By contrast, values approaching zero result in a dominance of the bernoulli trial governed by the feature matching.

Increased variability. A further element of randomness is introduced in line 28, where, rather than directly relying on a Bernoulli trial, the process incorporates an element of variance β within the sampling process.

A summary notation table for both the modeling and the resulting sampling process can be found in the Appendix.

Complexity Analysis. The computational complexity of HyDRA is governed by its hierarchical generative design, the cardinality of the user and item sets, and the nested stochastic operations. Let $|U| = n$ and $|I| = m$ denote the number of users and items, respectively. Let f be the number of latent features, c the number of user communities, and g the number of item categories.

Latent Factor Initialization. For each user $u \in U_s$ and item $i \in I_s$, the algorithm performs:

- `FilterPrior` operation: $O(f)$.
- Sampling from Dirichlet distributions: $O(f)$.
- Sampling scalar interactions and Beta variables: $O(1)$.

Thus, initializing all user and item latent parameters yields:

$$O(nf) + O(mf) = O((n+m)f)\,.$$

Interaction Generation. The most computationally intensive component is the dataset generation loop. For each user $u \in U$, the algorithm evaluates all items $i \in I$ until a minimum number of interactions τ is met. In the worst case, this involves:

- Inner product $\rho_u^{\top}\alpha_i$: $O(f)$.
- Scalar arithmetic, exponentiation, and sampling: $O(1)$.

The total cost, in the average case, for matching across all users is therefore bounded by:

$$O(nmf\tau)\,.$$

Overall Complexity. Combining the two phases, the total time complexity of HyDRA is:

$$O(nf + mf + nmf\tau) = O(nmf\tau)\,.$$

This reflects the expressiveness and flexibility of the model, enabling structured generation with nuanced control over sparsity, variability, and noise.

Memory Complexity. The memory requirements are dominated by:

- User and item latent vectors: $O(nf + mf)$.
- Generated dataset $\mathcal{D}$: average case $O(n\tau)$.

Optimized tensor-based implementations and sparse data structures can mitigate memory overhead while preserving efficiency.

5 Experimental Evaluation

To evaluate the effectiveness of HyDRA in generating preference data, we address two key research questions:

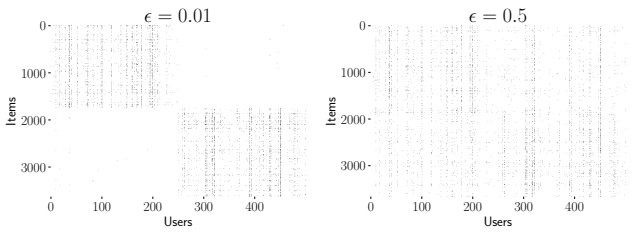
RQ1. How flexible is HyDRA in generating user-item *interactions*? Does it provide control on the resulting *data distributions*?

RQ2. Are the interactions generated by HyDRA realistic? Can the framework *replicate* the properties of real benchmark datasets?

To address **RQ1**, we conducted experiments to demonstrate the flexibility of the proposed hyper-parametrization in controlling the distributions.

Interactions among user communities and item categories. To conduct this set of experiments, we generate a synthetic dataset featuring two user communities and two item categories. For simplicity, we divide both users and items into two equal-sized groups, denoted as communities U_1 and U_2 and categories I_1 and I_2, respectively. The communities and categories are correlated: users in U_1 primarily prefer items in category I_1, while users in U_2 exhibit a preference for items in I_2.

Figure 1 illustrates the impact of the ε parameter on the visibility of communities and categories within the interaction matrix. When ε is close to 0 (Figure 1a), the matrix is dominated by two strongly connected components. As ε increases, the matrix becomes more dispersed, eventually reaching a state of complete homogeneity, as seen in Figure 1b. Additionally, Figure 2 presents histograms of user interactions within category I_1 for each community, demonstrating a transition from fully clustered to entirely overlapping interactions.



(a) **(b)**

Figure 1: Visualization of the user-item interaction matrix by varying the ε parameter. The X-axis reports the users, while the Y-axis represents the items. A dot in the position (u, i) indicates the user u interacted with item i.

Finally, Figure 3 presents the degree distributions for users and items in a log-log scale. Specifically, it compares the distributions of the entire dataset with those of the generated communities and categories. Remarkably, the degree distributions of the subpopulations maintain the same characteristics as those of the complete dataset, demonstrating the effectiveness of our method in preserving the overall structure while allowing for distinct community and category formations.

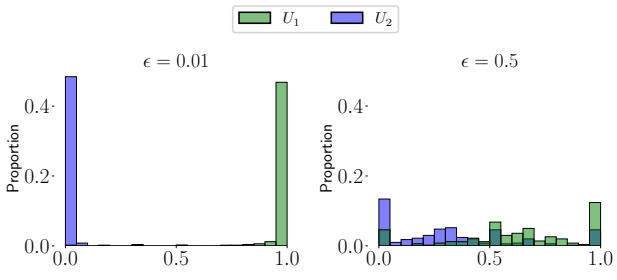


Figure 2: Histograms of user interactions with a specific topic of interest. The X-axis represents the percentage of items in I_1 within the users history. The Y-axis shows the proportion of users having that percentage.

Tweaking Distributions. We validate the flexibility of HyDRA by demonstrating that the degree distribution functions as a *plug-and-play* component. In Figure 4, we present the results of generation experiments where various combinations of prior distributions were applied. These combinations showcase the model's flexibility by producing a wide range of distribution shapes, emphasizing the precise control possible in the data generation process.

We also conduct a sensitivity analysis of the weighting hyperparameters ζ, ξ, and λ to evaluate their impact on the generated preferences and the resulting degree distributions. In this analysis, the frequency distributions are regulated by power-law, although similar behaviors are observed for the other priors listed in Table 2. Figure 5 illustrates the effects of these hyperparameters. Specifically, as ζ and ξ approach 0, the corresponding user engagement and item popularity distributions exhibit a bell-shaped curve, driven by the influence of the Beta distribution on user-item interactions. By contrast, higher values of these parameters sharpen the distribution, producing shapes that range from power-law to more peaked distributions. Regarding λ, increasing its value scales the overall sampling probability, thereby influencing the density of both user and item distributions.

Additional experiments exploring the effects of ζ, ξ, and λ on the degree distributions are provided in the Appendix.

To answer **RQ2**, we perform a final set of experiments to test the ability of generating realistic datasets. We consider two aspects.

Comparing distributions. In this study, we utilize existing datasets as reference benchmarks, focusing on Movielens-1M[2], which pertains to movie ratings, Yahoo! Music[3] (Yahoo-R3), which reflects user preferences collected during normal interactions with the Yahoo Music service, and Amazon Musical Instruments Reviews[4], which contains comments and ratings published by users w.r.t. musical instruments.

The results of the data generation process are illustrated in Figure 6. The top row shows the actual distributions from the datasets Movielens-1M, Yahoo-R3, and Amazon, while the bottom row presents the synthetic samples generated by HyDRA. As evident from the comparison, the generated samples closely replicate

[2]https://grouplens.org/datasets/movielens/1m/
[3]https://webscope.sandbox.yahoo.com/catalog.php?datatype=r
[4]https://www.kaggle.com/datasets/eswarchandt/amazon-music-reviews

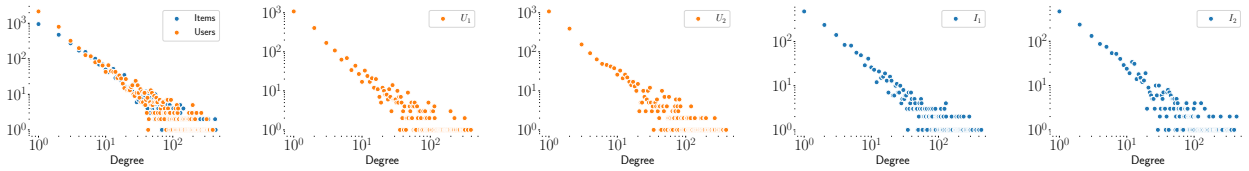


Figure 3: User/item degree distributions for the partitions obtained with $\varepsilon = .01$. The first graph shows the global degree distributions. Graphs 2 and 3 focus on the user communities U_1 and U_2, whereas 4 and 5 on item categories I_1 and I_2.

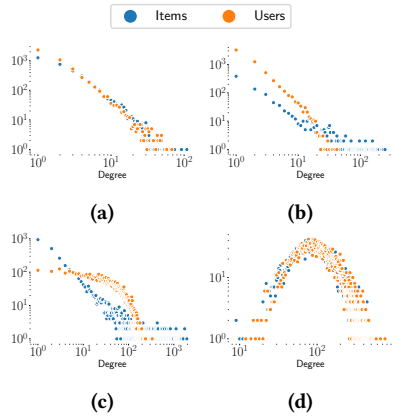


Figure 4: Degree distributions with the following priors: (a) Power-Law with exponential cut-off for users and items; (b) Power-Law with exponential cut-off for users and Power-Law for items; (c) Stretched Exponential for users, Power-Law for items; (d) Log-Normal distribution for users and items.

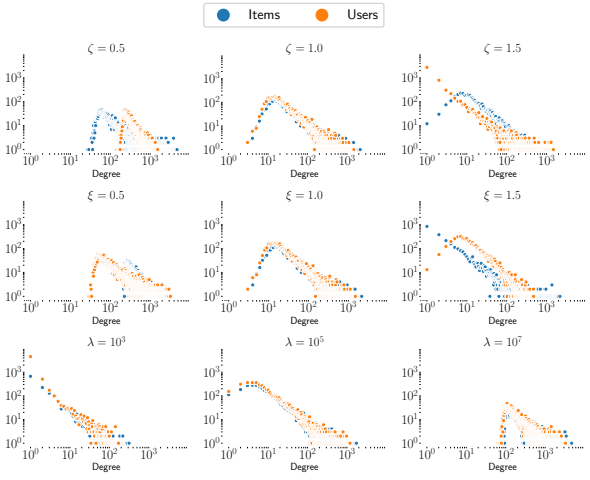


Figure 5: Effects of the ζ, ξ and λ hyper-parameters on the distributions of the generated data.

the structural properties of the original datasets. Details regarding the parameters for the synthetic samples generation can be found in the Appendix.

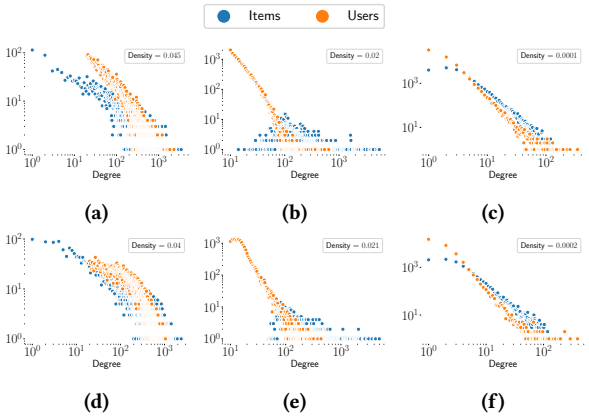


Figure 6: Real and Synthetic degree distributions of users and items. The top row depicts Movielens-1M (a), Yahoo-R3 (b) and Amazon (c), while the bottom row shows the corresponding synthetic samples ((c), (d) and (e)), generated by HyDRA.

Comparing benchmarking capabilities. To evaluate the quality of the generated data, we compared the performance of several recommendation algorithms on both the real and synthetic datasets. The underlying hypothesis is that if the synthetic data accurately mirrors the structural properties of the real data, the performance trends of recommendation algorithms should be similar across both datasets. We tested a suite of algorithms, including *ItemKNN* [13], *MultiVAE* [28], *BPR* [38], *Neural Matrix Factorization (NeuMF)* [18], and a popularity-based baseline (*Pop*). Similar recommendation scores and algorithm rankings across the datasets would indicate high similarity in data distributions.

In these experiments, we produced multiple instances of the generated data by varying the coefficient δ, to emulate noisy interactions that typically can occur within real preferences. We trained the recommendation algorithms on both real and synthetic datasets[5], and evaluated their performance in terms of Recall and Hit-Rate. Results with cut-offs 5, 10 and 20 are shown in Figures 7, 8, and 9, respectively. Notice that, on Amazon, due to the extreme sparsity of the dataset, we perform a pre-processing step where we removed the users and items having degree lower than 5.

The findings indicate a strong alignment between the performance metrics on the real and synthesized datasets, with the exception of cases where excessive noise was introduced. Most importantly, the ranking of algorithms on the synthetic data is consistent

[5]For training, we adopted the Recbole library [53]

with that on the real data, demonstrating that HyDRA can effectively work as a proxy for real data in benchmarking scenarios.

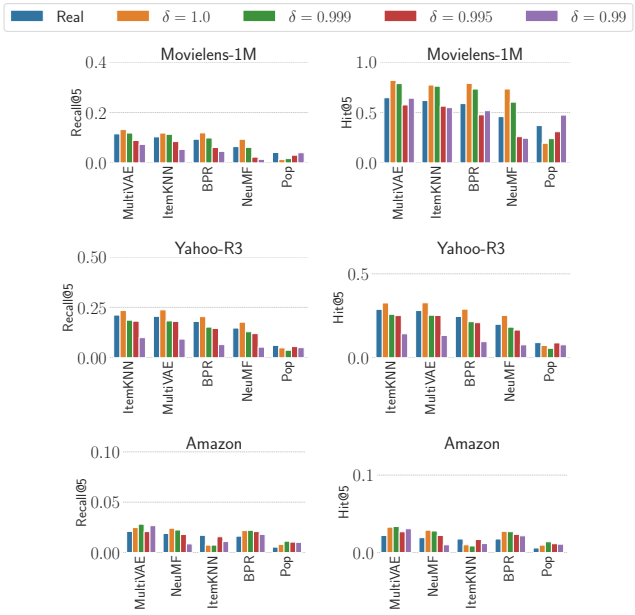


Figure 7: Performance comparison on Movielens-1M, Yahoo-R3, and Amazon, in terms of Recall@5 (left-column) and Hit-Rate@5 (right-column). Colors represent results on the real dataset and generated samples with varying values of δ.

6 Conclusions and Future Work

The paper has proposed HyDRA, a customizable generative probabilistic framework for user-item interactions. Our main intuition consists in modeling the generation process as a combination of three main factors: *User-Item Matching*, *User Engagement Level*, and *Item Popularity*, thus allowing control of each individual component. We implemented the theoretical formulation by endowing the framework with several control parameters that enable substantial control of the data generation process, under the aforementioned perspectives. We conducted extensive experimentation with HyDRA, showing its effectiveness concerning these tasks.

The proposed framework is also amenable for further extensions. In particular, the model can be adapted to cope with more complex features, such as explicit ratings, or textual content associated with preferences (e.g., reviews [36]). Additionally, the framework can be further extended to include user-user interactions or item-item relationships.

Acknowledgements

This work has been partially funded by MUR on D.M. 351/2022, PNRR Ricerca, CUP H23C22000440007, and on D.M. 352/2022, PNRR Ricerca, CUP H23C22000550005. It is also supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union – NextGenerationEU.

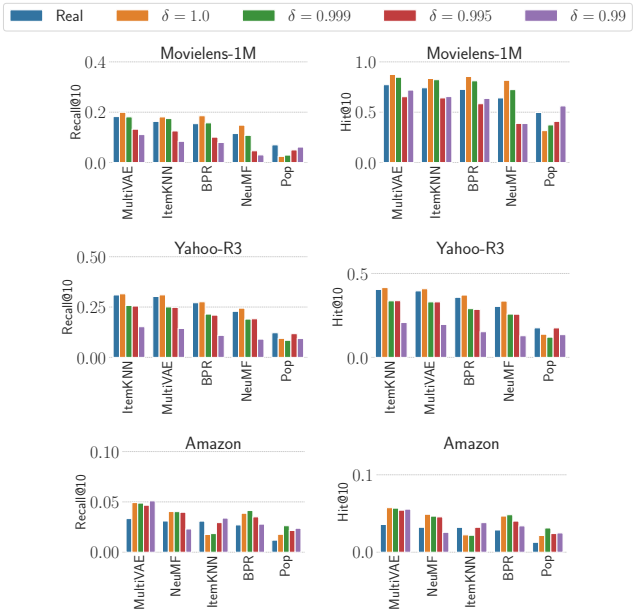


Figure 8: Performance comparison on Movielens-1M, Yahoo-R3, and Amazon, in terms of Recall@10 (left-column) and Hit-Rate@10 (right-column). Colors represent results on the real dataset and generated samples with varying values of δ.

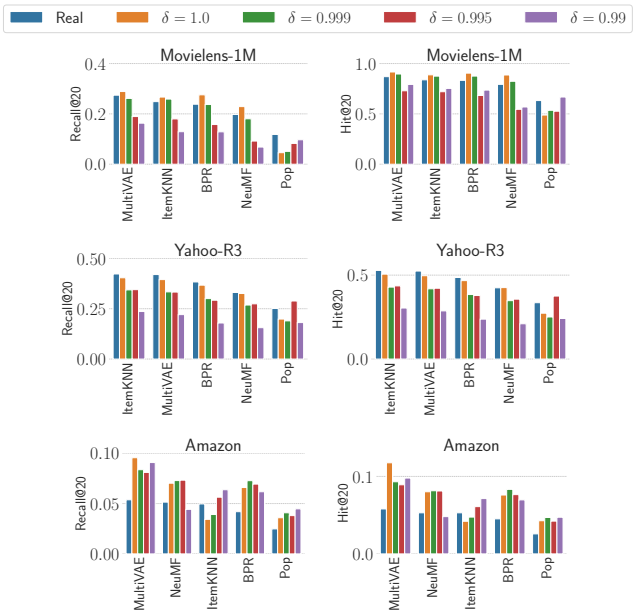


Figure 9: Performance comparison on Movielens-1M, Yahoo-R3, and Amazon, in terms of Recall@20 (left-column) and Hit-Rate@20 (right-column). Colors represent results on the real dataset and generated samples with varying values of δ.

References

[1] Gediminas Adomavicius, Jesse C. Bockstedt, et al. 2022. Recommender systems, ground truth, and preference pollution. *AI Magazine* (2022), 177–189.

[2] Dua'a Al-Hajjar, Nouf Jaafar, Manal Al-Jadaan, and Reem Alnutaifi. 2015. Framework for Social Media Big Data Quality Analysis. In *New Trends in Database and Information Systems II.*

[3] Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML.*

[4] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* (1999).

[5] Francois Belletti, Karthik Lakshmanan, Walid Krichene, Yi-Fan Chen, and John R. Anderson. 2019. Scalable Realistic Recommendation Datasets through Fractal Expansions. *ArXiv* (2019).

[6] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD cup and workshop.*

[7] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval.*

[8] Jesus Bobadilla and Abraham Gutierrez. 2024. Wasserstein GAN-based architecture to generate collaborative filtering synthetic datasets. *Applied Intelligence* (2024).

[9] Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. 2018. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems.*

[10] Sushma Channamsetty and Michael D. Ekstrand. 2017. Recommender Response to Diversity and Popularity Bias in User Profiles. In *International Florida Artificial Intelligence Research Society Conference.*

[11] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. 2009. Power-Law Distributions in Empirical Data. *SIAM Rev.* (2009).

[12] Erica Coppolillo, Marco Minici, Ettore Ritacco, et al. 2024. Balanced Quality Score: Measuring Popularity Debiasing in Recommendation. *ACM Trans. Intell. Syst. Technol.* (2024).

[13] Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.* (2004).

[14] Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. 2012. The Yahoo! Music Dataset and KDD-Cup '11. In *Proceedings of KDD Cup 2011 competition.* 8–18.

[15] Paul Erdős, Alfréd Rényi, et al. 1960. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci* (1960), 17–60.

[16] Min Gao, Junwei Zhang, Jundong Li Junliang Yu, et al. 2021. Recommender systems based on generative adversarial networks: A problem-driven perspective. *Information Sciences* (2021), 1166–1185.

[17] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* (2016), 19:1–19:19.

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, et al. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th Edition of WWW.*

[19] Bernd Heinrich, Marcus Hopf, Daniel Lohninger, Alexander Schiller, and Michael Szubartowicz. 2019. Data quality in recommender systems: the impact of completeness of item content data on prediction accuracy of recommender systems. *Electronic Markets* (2019).

[20] Yupeng Hou, Jiacheng Li, Zhankui He, et al. 2024. Bridging Language and Items for Retrieval and Recommendation.

[21] Changwei Hu, Piyush Rai, and Lawrence Carin. 2016. Non-negative Matrix Factorization for Discrete Data with Hierarchical Side-Information. In *International Conference on Artificial Intelligence and Statistics.*

[22] Abhinav Jain, Hima Patel, Lokesh Nagalapatti, et al. 2020. Overview and Importance of Data Quality for Machine Learning Tasks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*

[23] Ray Jiang, Silvia Chiappa, Tor Lattimore, et al. 2019. Degenerate Feedback Loops in Recommender Systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (AIES '19).* 383–390.

[24] Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, et al. 2021. Graph Condensation for Graph Neural Networks. *ArXiv* (2021).

[25] Dokyun Lee and Kartik Hosanagar. 2014. Impact of Recommender Systems on Sales Volume and Diversity. In *Procs. Association for Information Systems ICIS.*

[26] Dokyun Lee and Kartik Hosanagar. 2019. How Do Recommender Systems Affect Sales Diversity? A Cross-Category Investigation via Randomized Field Experiment. *Inf. Syst. Res.* (2019).

[27] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. 2010. Kronecker Graphs: An Approach to Modeling Networks. *J. Mach. Learn. Res.* (2010).

[28] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the World Wide Web Conference, WWW.*

[29] Derek Lilienthal, Paul Mello, Magdalini Eirinaki, and Stas Tiomkin. 2023. Multi-Resolution Diffusion for Privacy-Sensitive Recommender Systems. *CoRR* (2023).

[30] Fan Liu, Zhiyong Cheng, Huilin Chen, Yinwei Wei, et al. 2022. Privacy-Preserving Synthetic Data Generation for Recommendation Systems. In *Proceedings of the 45th International ACM SIGIR.*

[31] Yudan Liu, Kaikai Ge, Xu Zhang, and Leyu Lin. 2019. Real-time Attention Based Look-alike Model for Recommender System. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*

[32] Nguyen Luong Vuong, Nam Vo, and Jason Jung. 2023. DaGzang: a synthetic data generator for cross-domain recommendation services. *PeerJ Computer Science* (2023).

[33] Christian Matt, Thomas Hess, and Christian Weiß. 2013. The Differences between Recommender Technologies in their Impact on Sales Diversity. In *Procs. Association for Information Systems ICIS.*

[34] Sheshera Mysore, Andrew Mccallum, and Hamed Zamani. 2023. Large Language Model Augmented Narrative Driven Recommendations. In *Proceedings of the 17th ACM Conference on Recommender Systems.*

[35] MEJ Newman. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics* (2005).

[36] Qiyao Peng, Hongtao Liu, Hongyan Xu, Qing Yang, Minglai Shao, and Wenjun Wang. 2024. Review-LLM: Harnessing Large Language Models for Personalized Review Generation. arXiv:2407.07487 [cs.CL] https://arxiv.org/abs/2407.07487

[37] Oumaima Reda and Ahmed Zellou. 2023. Assessing the quality of social media data: a systematic literature review. *Bulletin of Electrical Engineering and Informatics* (2023).

[38] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence.*

[39] Manoel Horta Ribeiro, Veniamin Veselovsky, and Robert West. 2023. The Amplification Paradox in Recommender Systems. In *Proceedings of the 17th International AAAI Conference on Web and Social Media, ICWSM.*

[40] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. 2003. Trust Management for the Semantic Web. In *The Semantic Web - ISWC 2003.* 351–368.

[41] Sven Schmit and Carlos Riquelme. 2017. Human Interaction with Recommendation Systems: On Bias and Exploration. *ArXiv* (2017).

[42] Oren Sar Shalom, Shlomo Berkovsky, Royi Ronen, Elad Ziklik, and Amihood Amir. 2015. Data Quality Matters in Recommender Systems. *Proceedings of the 9th ACM Conference on Recommender Systems* (2015).

[43] Manel Slokom, Martha A. Larson, and Alan Hanjalic. 2020. Partially Synthetic Data for Recommender Systems: Prediction Performance and Preference Hiding. *CoRR* (2020).

[44] Matthew Smith, Laurent Charlin, and Joelle Pineau. 2017. A Sparse Probabilistic Model of User Preference Data. In *Proceedings of the 30th Canadian Conference on Artificial Intelligence.*

[45] Zhu Sun, Di Yu, Hui Fang, Jie Yang, et al. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison *(RecSys '20).* 23–32.

[46] Karen H. L. Tso and Lars Schmidt-Thieme. 2006. Evaluation of Attribute-Aware Recommender System Algorithms on Data with Varying Characteristics. In *Advances in Knowledge Discovery and Data Mining.*

[47] Thanh Vinh Vo and Harold Soh. 2018. Generation meets recommendation: proposing novel items for groups of users. In *Proceedings of the 12th ACM Conference on Recommender Systems.*

[48] Qinyong Wang, Hongzhi Yin, Hao Wang, Q. V. Hung Nguyen, et al. 2019. Enhancing Collaborative Filtering with Generative Augmentation. In *Proceedings of the 25th ACM SIGKDD KDD.*

[49] Wenlin Wang. 2021. Learning to Recommend from Sparse Data via Generative User Feedback. In *AAAI Conference on Artificial Intelligence.*

[50] Xiao Wang, Ruijia Wang, Chuan Shi, Guojie Song, and Qingyong Li. 2020. Multi-Component Graph Convolutional Collaborative Filtering. In *EAAI of 35th Conference on Artificial Intelligence AAAI.*

[51] Jiahao Wu, Wenqi Fan, Shengcai Liu, Qijiong Liu, et al. 2023. Dataset Condensation for Recommendation.

[52] Qi Zhang, Longbing Cao, Chongyang Shi, and Liang Hu. 2021. Tripartite Collaborative Filtering with Observability and Selection for Debiasing Rating Estimation on Missing-Not-at-Random Data. In *EAAI 2021 of 35th Conference on Artificial Intelligence AAAI.*

[53] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, et al. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM.* 4653–4664.

[54] Xiangyu Zhao, Long Xia, Lixin Zou, Hui Liu, et al. 2021. UserSim: User Simulation via Supervised GenerativeAdversarial Network. In *Proceedings of the Web Conference 2021.*

[55] Dong Hong Zhu, Yawei Wang, and Ya Ping Chang. 2018. The influence of online cross-recommendation on consumers' instant cross-buying intention: The moderating role of decision-making difficulty. *Internet Res.* (2018).

[56] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-Aware Tensor-Based Recommendation. In *ACM International Conference on Information and Knowledge Management.*

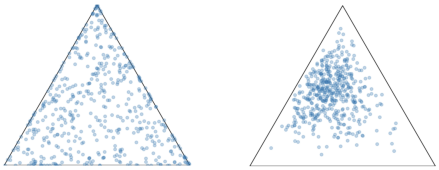


Figure 10: User profiles generated (in a three-dimensional feature space) through multiple prior sampling (left) as opposed to single prior sampling (right). Each point represents a sample in the hierarchical sampling process. By enabling a specific prior μ_u^ρ for each user, the resulting samples are evenly distributed over the whole latent space. By contrast, a single prior μ^ρ results in a concentration within specific regions of the space.

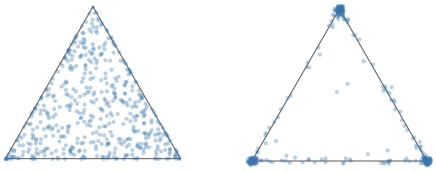


Figure 11: User (left) and item (right, with jitter) representations within a three-dimensional feature space, according to the proposed User-Item Matching model. Users are evenly distributed within the latent space. By contrast, item samples tend to concentrate along specific subsets of the latent features, representing edges/corners in the feature space.

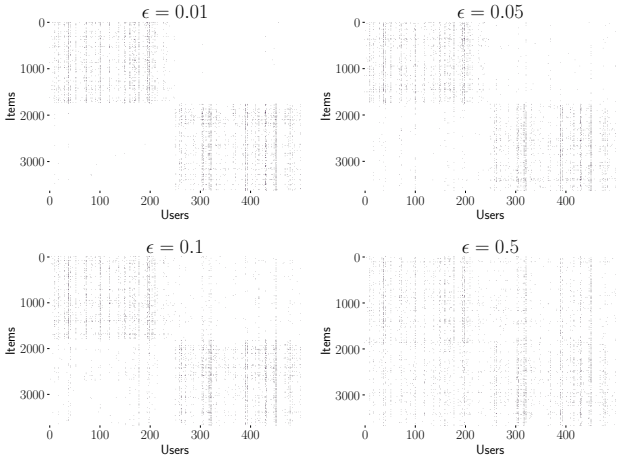


Figure 12: Visualization of the user-item interaction matrix by varying the ε parameter. The X-axis reports the users, while the Y-axis represents the items. A dot in the position (u, i) indicates the user u interacted with item i.

A User-Item Interaction

Figures 10 and 11 show how users and items latent factors are distributed by employing the hierarchical sampling process described in Section 3.

Figures 12 depicts the impact of the ε parameter over the user-item interaction matrix. Specifically, as ε increases, the interaction matrix becomes gradually denser, eventually reaching full homogeneity.

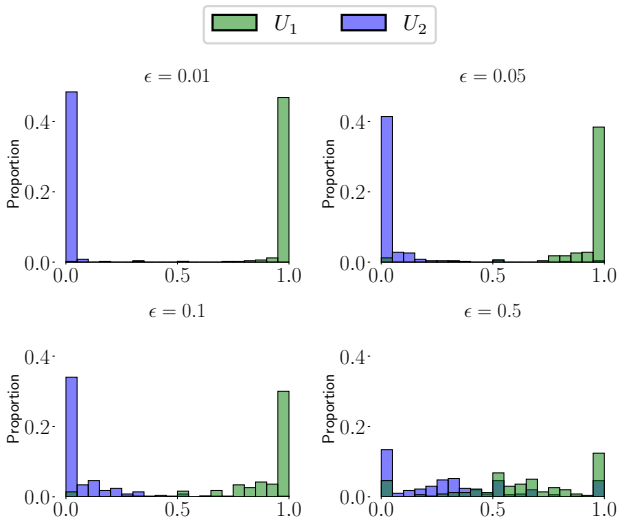


Figure 13: Histograms of user interactions with a specific topic of interest. The X-axis represents the percentage of items in I_1 within the users history. The Y-axis shows the proportion of users having that percentage.

As a complementary analysis, Figure 13 presents the proportion of items in I_1 in the interaction history of each communities (U_1 and U_2). We show that, when ε approaches 0, the histograms are totally clustered; vice-versa, when it is close to 1, the interactions overlap.

B Hyper-parameter tuning

We here report some additional experiments concerning the impact of the weighting factors ζ, ξ, and λ over the generated degree distributions, as shown in Figures 14, 15, and 16, respectively. As we can see from the plots, fixed ξ (resp. ζ) to 1, when ζ (resp., ξ) increases, the distributions of users (resp., items) become more sharpened, conversely following a Normal pattern when the control value is low, due to the Bernoullian sampling process. Regarding λ, the higher the value, the higher the minimum degree of users and items, hence the density of the generated dataset.

Symbol	Meaning
$U = \{1, \ldots, n\}$	Set of users
$I = \{1, \ldots, m\}$	Set of items
$U = \{U_1, U_2, \ldots, U_c\}$ and $I = \{I_1, I_2, \ldots, I_g\}$,	Populations of users and items that exhibit strong internal homophily
$\mathcal{D}$	Dataset of user-item interactions
$r \in \{0, 1\}$	interaction occurrence (relative to (u, i): if (u, i) exists then $r = 1$)
ν	Parameters governing the preferences
$z_u = \lvert\{i : (u, i) \in \mathcal{D}\}\rvert, \varrho_u$	Engagement level and relative likelihood for user u
$y_i = \lvert\{u : (u, i) \in \mathcal{D}\}\rvert, \varphi_i$	Popularity frequency and likelihood for item i
$\pi_k, k \in \{1, .., K\}, P_{\theta_k}(z_u)$	User engagement distribution and probability governed by the parameters of π
$\psi_h, h \in \{1, .., H\}, P_{\vartheta_h}(y_i)$	Item popularity distribution and probability governed by the parameters of ψ
ρ_u	User latent factors
μ_u^ρ	User parameters for sampling ρ_u
α_i	Item latent factors
μ_i^α	Item parameters for sampling α_i
$\hat{\mathcal{D}} = \{\mathcal{D}, \mathbf{z}, \mathbf{y}, \nu\}$	Estimated dataset
$Q(h, k \mid \hat{\mathcal{D}}, \Theta)$	Proposal probability distribution

Table 3: Notation table.

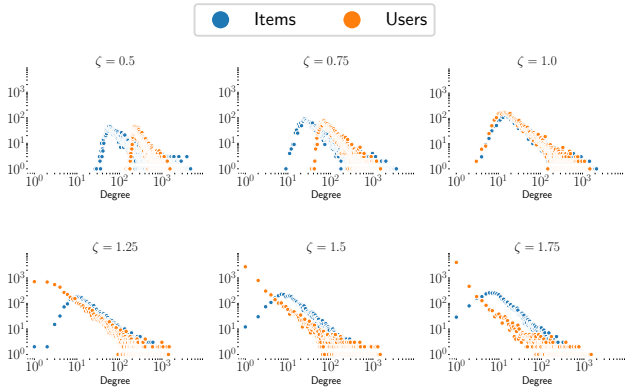


Figure 14: Effects of ζ on the generated data distributions.

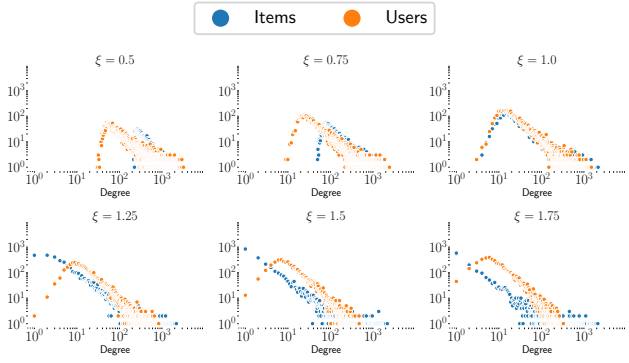


Figure 15: Effects of ξ on the generated data distributions.

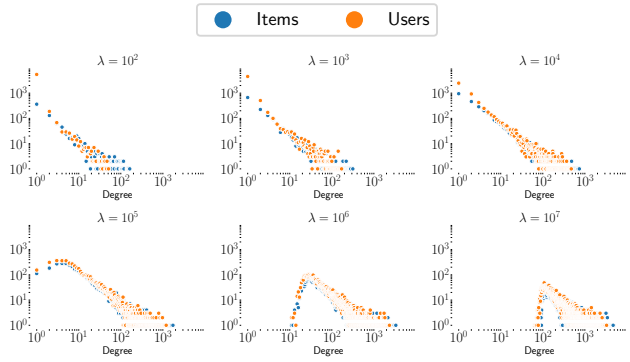


Figure 16: Effects of λ on the generated data distributions.

C Synthetic samples parameters

For the original user/item distributions, we determined that the best fits are provided by Log-Normal/Stretched Exponential distributions for Movielens, Power-Law/Log-Normal distributions for Yahoo-R3, and Power-Law/Power-Law for Amazon. Using these estimated parameters, we regenerated the datasets artificially:

- Movielens-1M: $x_u \sim$ Log-Normal($\mu = 4.61, \sigma = 1.2$), $y_i \sim$ StretchedExponential($\lambda = 0.025, \beta = 0.691$), $\zeta = 1.0$, $\xi = 1.4$ and $\lambda = 55k$.
- Yahoo-R3: $x_u \sim$ Power-Law($\alpha = 3.8$), $y_i \sim$ Log-Normal($\mu = -0.98, \sigma = 2.44$), $\zeta = 1.9$, $\xi = 1.0$ and $\lambda = 650k$.
- Amazon: $x_u \sim$ Power-Law($\alpha = 2.6$), $y_i \sim$ Power-Law($\alpha = 1.9$), $\zeta = 1.1$, $\xi = 0.95$ and $\lambda = 20k$.