Data Generation via Latent Factor Simulation for Fairness-aware Re-ranking

ELENA STEFANCOVA, Comenius University Bratislava, Slovakia
CASSIDY ALL, Department of Information Science; University of Colorado, Boulder, USA
JOSHUA PAUP, Department of Information Science; University of Colorado, Boulder, USA
MARTIN HOMOLA, Comenius University Bratislava, Slovakia
NICHOLAS MATTEI, Department of Computer Science; Tulane University, USA
ROBIN BURKE, Department of Information Science; University of Colorado, Boulder, USA

Synthetic data is a useful resource for algorithmic research. It allows for the evaluation of systems under a range of conditions that might be difficult to achieve in real world settings. In recommender systems, the use of synthetic data is somewhat limited; some work has concentrated on building user-item interaction data at large scale. We believe that fairness-aware recommendation research can benefit from simulated data as it allows the study of protected groups and their interactions without depending on sensitive data that needs privacy protection. In this paper, we propose a novel type of data for fairness-aware recommendation: synthetic recommender system outputs that can be used to study re-ranking algorithms.

CCS Concepts: • **Information systems → Recommender systems**.

Additional Key Words and Phrases: recommender systems, fairness, simulation, synthetic data

ACM Reference Format:
Elena Stefancova, Cassidy All, Joshua Paup, Martin Homola, Nicholas Mattei, and Robin Burke. 2024. Data Generation via Latent Factor Simulation for Fairness-aware Re-ranking. 1, 1 (October 2024), 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Research in fairness-aware recommendation usually takes two paths: model-based, in which fairness objectives are integrated into the machine learning model itself, and post-processing or re-ranking, in which fairness objectives are applied to re-order the results of a recommender system that is not itself fairness-aware. The third option, pre-processing the input data to support fairness is less often applied in recommender systems. In this paper, we concentrate on supporting fairness-aware recommendation research that uses a post-processing approach.

Studying fairness-aware post-processing requires as input pre-computed recommendations that can be re-ranked and their fairness properties studied. While it is relatively straightforward to use existing recommendation data sets and frameworks to produce recommendations for this purpose, the set of available data sets with protected features around which fairness concerns might arise is relatively limited. Researchers often use a relatively limited repertoire of such data sets. In other cases, researchers may invent fairness concerns independent of detailed analysis of the fairness properties of the domain; for example, by designating some percentage of least popular items as protected. We believe such methodologies are valid and valuable (and part of our own experimental practice). However, in all such cases,

Authors' addresses: Elena Stefancova, elena.stefancova@fmph.uniba.sk, Comenius University Bratislava, , Bratislava, Slovakia; Cassidy All, cassidy.all@colorado.edu, Department of Information Science; University of Colorado, Boulder, , Boulder, Colorado, USA, 80309; Joshua Paup, joshua.paup@colorado.edu, Department of Information Science; University of Colorado, Boulder, , Boulder, Colorado, USA, 80309; Martin Homola, homola@fmph.uniba.sk, Comenius University Bratislava, , Bratislava, Slovakia; Nicholas Mattei, nsmattei@tulane.edu, Department of Computer Science; Tulane University, , New Orleans, Louisiana, USA, 70118; Robin Burke, robin.burke@colorado.edu, Department of Information Science; University of Colorado, Boulder, , Boulder, Colorado, USA, 80309.

2024. Manuscript submitted to ACM

researchers are limited by the characteristics of the data set and induced recommendations including its particular distribution of protected features.

In this paper, we describe our methodology for creating synthetic recommendation lists for studying fairness-aware re-ranking: LAtent Factor Simulation. As the name implies, we create synthetic data via first simulating the latent factor matrices that a matrix factorization model could produce and then generate sample ratings from these matrices. We show that it is possible to produce recommendation lists with characteristics similar to those that come from factorization models, and that we can adjust a number of dataset characteristics related to protected groups to evaluate fairness-aware re-rankers under a variety of conditions.

2 RELATED WORK

There have been a number of existing approaches to creating synthetic data for recommender systems. The first category is generation methods that use heuristics to generate data. These approaches leverage strong assumptions on the dataset distributions and properties. DataGenCARS [7] uses a description of user profiles to generate a dataset with specified properties. The designer of the algorithm manually designs and describes user "profiles". Another approach is to mine generic profiles from historical data by user clustering, as done in [15], and these profiles can be extended to include additional features and contexts as in [16]. In these works, the experimenter has to manually craft the user model, and make assumptions on how users behave. The quality of the resulting data will only be as good as these assumptions, however, and it is difficult to capture the texture of user-item interactions in their full detail.

An alternative to a manually-specified model is one that is learned from a data source. In unpublished work, Lin used a traditional matrix factorization model, and then used variational auto-encoders to generate new user vectors [13]. Researchers in reinforcement learning often use synthetic data generation for experimentation [22, 23], including latent vector models. However, these models are often unrealistic, or too simple. For instance, authors in [10] use 20-dimensional user latent vectors, which is very low (typical values in recommender systems range between 100 and 200). Authors of [22] generate a dataset that has 40% sparsity (40% of all possible combinations of users and items contain ratings), whereas industrial-grade datasets have sparsity orders of magnitude less ($\tilde{0}.1\%$). A method for generating realistically sparse datasets for recommendation is described in [3]. However, the Kronecker product expansion described here, a graph theoretic technique, creates both synthetic users and synthetic items, and does not generate item or user metadata, so cannot for studying fairness-aware recommendation.

Another approach to creating data that can be shared without compromising user privacy is that of data obfuscation. For example, [21] and [18] are examples of algorithms that use this approach. However, because the ratings matrix is modified but not replaced, each entry corresponds to a real user who identity might be compromised. Because LAFS is completely synthetic, no similar risks exist. A probabilistic approach to handling protected features was explored in [5], which used rating data properties to derive behavioral features to stand in for (unknown) demographic data.

As noted above, we concentrate in this paper on supporting research in fairness-oriented re-ranking. This approach represents a substantial and on-going research arc in fairness-aware recommendation. One of the advantages of post-processing is that the fairness objective can be adapted on the fly without relearning the recommendation model. It is also the case that specific guarantees relative to item distribution can be ensured by conducting batch reranking across an entire set of recommendations to be delivered. Such guarantees are not available if when fairness is included in part of the training objective. Re-ranking is also advantageous when the tradeoff between fairness objectives in inherently dynamic as in [2]. Re-ranking has been used to address a wide variety of fairness-aware recommendation challenges. There has been considerably investigation of greedy sublist optimization as a technique. In binary fairness

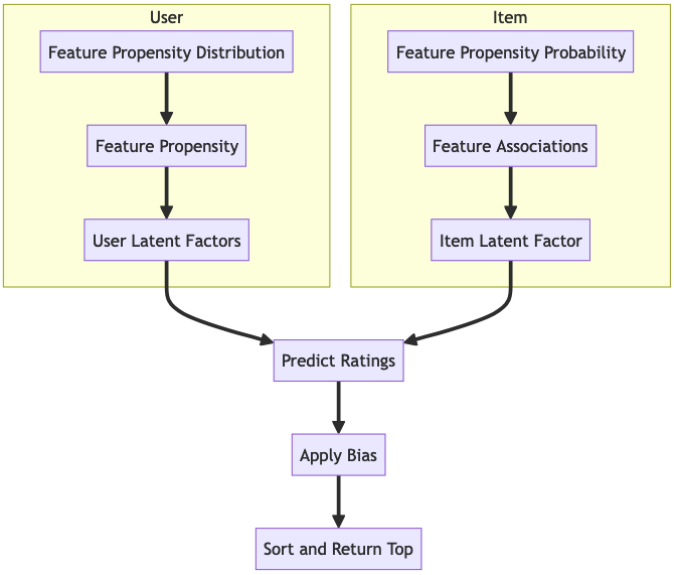

Fig. 1. Overview of the LAFS data generation process

situations, we note the algorithms in [1, 2, 6, 9, 11, 12, 14, 19]. The other main category of re-ranking solutions are batch-oriented approaches that consider the whole collection of recommendation lists at once and optimize across all lists aimed at all users. Examples of this technique include [4, 17, 20]. A detailed survey of approaches to fairness-aware recommendation can be found in [8].

3 LATENT FACTOR SIMULATION

The LAFS data generation process happens in several stages. See Figure 1.

3.1 Latent Factors

As shown in Table 1, U and V are the user and item latent factor matrices with k latent factors. We designate the first k_s of the latent factors as corresponding to protected features of items, and the remaining $k - k_s$ factors correspond to other aspects of the items. Thus, we can generate data with multiple protected features for studying intersection impacts of fairness-aware re-ranking.

We generate these latent factors for each (simulated) user and item in a two-step process. First, we generate two matrices of propensities that has a value for each user and each factor, and we do the same for each item and each factor. Then, from the propensities we generate a latent factor value for the corresponding U or V matrix. This two-step process avoids having the latent factors tied exactly to the user propensities, which would reduce the variance of the generated factors and make the recommendation lists too similar to each other.

More formally, the experimenter specifies a vector of d_i values, one for each factor, representing the probability of an item be associated with that factor. The assumption is that latent factors are closely tied to the presence or absence of a

n_i	The number of items
n_u	The number of users per propensity distribution
k	The number of latent factors
s	The number of sensitive factors ($s < k$)
σ_f	Standard deviation for factor generation
$d_u = [(\mu_{u1}, \sigma_{u1}), (\mu_{u2}, \sigma_{u2})..., (\mu_{uk}, \sigma_{uk})]$	Feature propensity distributions for users
$\Pi_u = [\pi_{u1}, \pi_{u2}, ..., \pi_{uf}]$	Feature propensities for users
$\Pi_U = [\Pi_u \forall u]$	The matrix of all user-feature associations
U	$< n_u \times n_f >$ matrix of user latent factors
$d_i = [d_{i1}, d_{i2}, ..., (d_{if}]$	Feature propensity probabilities for items
$\Pi_i = [\pi_{i1}, \pi_{i2}, ..., \pi_{if}]$	Feature associations for item i
$\Pi_I = [\Pi_i \forall i]$	The matrix of all item-feature associations
V	$< n_i \times n_f >$ matrix of item latent factors
l'	The size of the initial recommendation list
l	The size of the output recommendation list
$B = [(\mu_1, \sigma_1), (\mu_2, \sigma_2), ...(\mu_f, \sigma_s)]$	Bias generators for sensitive features

Table 1. Notation for LAFS data generation

feature for each item and so each π_i value is binary: the item has the feature or it doesn't. For each item i and factor j, each binary pi_{ij} propensity value is computed by treating the d_i value as a probability in a Bernoulli trial.

This binary propensity label for items has the additional advantage that it is easy to identify items as protected relative to the k_s sensitive features. We can control how many items will be protected relative to different sensitive by changing the d_i probabilities for these sensitive features.

Once we have the Π_I propensity matrix, we generate the item latent feature matrix V by sampling from a normal distribution centered around the corresponding propensity. That is, $v_{ij} = \mathcal{N}(\pi_{ij}, \sigma_f)$.

For user latent factors, the process is slightly more complex because we do not assume binary associations between users and latent factors. Instead of specifying the probability of the association for a Bernoulli trial, the experimenter specifies feature-specific parameters for a normal distribution: (μ_{u1}, σ_{u1}). The Pi_U matrix is filled by drawing from this distribution, and then, as with the V matrix, the U is filled by drawing from a normal distribution centered on the propensity: $u_{uj} = \mathcal{N}(\pi_{uj}, \sigma_f)$. This can be performed for users groups of different sizes and propensity distributions.

The latent factors for users and items are therefore created through similar two-step processes. First, propensities are drawn from experimenter-specified distributions. Second, latent factor values are drawn from a normal distribution centered on the prospensities.

Note that we do not require experimenters to specify the full co-variance matrix for their distributions. We assume that the propsensities and latent factor are independent. The values of the latent factors are not constrained and do not have to be in any particular range, similar to latent factors derived from unconstrained matrix factorization.

3.2 Rating Generation

Once we have the latent factor matrices U and V, the next step is to generate recommendation lists from them. For each user u, we select a set of l' items at random. For each item, we compute the rating by multiplying the U_u and I_i latent factor vectors.

To simulate bias against protected group items, we apply a randomly generated bias penalty (which could be zero) to the computed rating for each sensitive feature that it possesses. The penalty is drawn from an experiment-specified vector of bias distributions: B.

The final set of items and their adjusted ratings are then sorted and the top l items returned as the simulated recommendation list. This sorting process simulates the sorting that occurs when a recommender system is producing top-k recommendations for the user. We would expect that recommendation lists would contain items with higher rating predictions.

After all recommendations have been generated, an experimenter-specified min-max transformation is applied to normalize the scores from all of the generated recommendations (l' lists) to the experimenter's preferred range. This is not strictly necessary but helps make the recommendation output easier to understand.

3.3 User dynamics

For some of the experiments that we are conducting, we are interested in the dynamic properties of a recommender system with regard to fairness and in particular, how fairness outcomes may shift when the user population shifts. To support this kind of experiment, we have added an additional feature to LAFS, which allows for a sequence of user *regimes* in the recommendation output. A user regime is defined by a particular d_u, that is user feature propensity distribution. We wish to start with a user base that is very interested in items containing the protected feature and then move to a user base that is not.

To support these types of chronological shifts, we allow for a list of d_u distributions $[d_u^1, d_u^2, ..., d_u^r]$ where r is the desired number of regimes and a list of n_u user counts $[n_u^1, n_u^2, ...n_u^r]$. To generate the recommendation data, we iterate through the list of distributions, producing n_u^1 users using the distribution d_u^1 and then n_u^2 users with the next distribution, and so on, until all the users are generated.

The code for LAFS is available as open source on GitHub under the MIT License[1].

4 CONCLUSION

One thing that has been clear in this work is that there is a lack of an established methodology for evaluating data set simulation in recommender systems. Part of the reason is that, beyond basic measures like sparsity, we do not have strong predictive measures connecting data set properties with recommender system performance and it is therefore difficult to assess whether findings relative to a synthetic data set will be predictive of performance on a real-world one.

Clearly, additional research is needed on this and other aspects of simulated evaluation of recommender systems. As we continue to work with LAFS, we plan to explore a variety of visualizations, metrics and other techniques to compare its simulated output with recommendations from real algorithms and also to compare the synthetic latent factors with those derived from real-world data sets.

While the LAFS generation method has proved useful for our purposes (see [2]), there are additional improvements that we envision. Currently, we do not represent item popularity and so items in user's initial recommendation lists are selected uniformly at random.It would be more realistic to assign items to ranks along a long-tail distribution to better mimic the distribution of items in recommendation lists. Also, we have found that, in some data sets, there is correlation (and anti-correlation) between sensitive features, so, rather than assuming independence, it may be useful to allow experimenters to specify the co-variance matrix for these features. We plan to include these features in our next release of the tool.

[1] https://github.com/that-recsys-lab/lafs

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation under awards IIS-2107577 and IIS-2107505. This work was also supported by the Tatra Bank Foundation, and the national projects nos. VEGA-1/0621/22 and APVV-20-0353 awarded by Slovak VEGA and APVV research agencies.

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2012. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2012), 896–911. https://doi.org/10.1109/TKDE.2011.15

[2] Amanda Aird, Paresha Farastu, Joshua Sun, Elena Štefancová, Cassidy All, Amy Voida, Nicholas Mattei, and Robin Burke. to appear. Dynamic fairness-aware recommendation through multi-agent social choice. *ACM Transactions on Recommender Systems (TORS)* (to appear).

[3] Francois Belletti, Karthik Lakshmanan, Walid Krichene, Yi-Fan Chen, and John Anderson. 2019. Scalable realistic recommendation datasets through fractal expansions. *arXiv preprint arXiv:1901.08910* (2019).

[4] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. *arXiv preprint arXiv:1805.01788* (2018).

[5] Robin Burke, Jackson Kontny, and Nasim Sonboli. 2018. Synthetic Attribute Data for Evaluating Consumer-side Fairness, In Workshop on Responsible Recommendation (FATRec 2018. *arXiv preprint arXiv:1809.04199*.

[6] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. 2018. Ranking with Fairness Constraints. In *45th International Colloquium on Automata, Languages, and Programming (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 107)*, Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella (Eds.). Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany. https://doi.org/10.4230/LIPIcs.ICALP.2018.28

[7] María del Carmen Rodríguez-Hernández, Sergio Ilarri, Ramón Hermoso, and Raquel Trillo-Lado. 2017. DataGenCARS: A generator of synthetic data for the evaluation of context-aware recommendation systems. *Pervasive and Mobile Computing* 38 (2017), 516–541.

[8] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2022. Fairness in Information Access Systems. arXiv:2105.05779 [cs.IR]

[9] Michael D Ekstrand and Daniel Kluver. 2021. Exploring Author Gender in Book Rating and Recommendation. *User Modeling and User-Adapted Interaction* 31, 3 (2021). https://doi.org/10.1007/s11257-020-09284-2

[10] Arik Friedman, Shlomo Berkovsky, and Mohamed Ali Kaafar. 2016. A differential privacy framework for matrix factorization recommender systems. *User Modeling and User-Adapted Interaction* 26, 5 (2016), 425–458.

[11] David García-Soriano and Francesco Bonchi. 2021. Maxmin-Fair Ranking: Individual Fairness under Group-Fairness Constraints. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 436–446. https://doi.org/10.1145/3447548.3467349

[12] Elizabeth Gómez, Carlos Shui Zhang, Ludovico Boratto, Maria Salamó, and Mirko Marras. 2021. The Winner Takes it All: Geographic Imbalance and Provider (Un)fairness in Educational Recommender Systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1808–1812. https://doi.org/10.1145/3404835.3463235

[13] Kun Lin. 2020. Synthetic Data set Creation in Recommender Systems with Variational Autoencoder. (2020). Unpublished manuscript.

[14] Weiwen Liu and Robin Burke. 2018. Personalizing Fairness-aware Re-ranking. *arXiv preprint arXiv:1809.02921* (2018). Presented at the 2nd FATRec Workshop held at RecSys 2018, Vancouver, CA..

[15] Diego Monti, Giuseppe Rizzo, and Maurizio Morisio. 2019. All You Need is Ratings: A Clustering Approach to Synthetic Rating Datasets Generation. arXiv:1909.00687 [cs.IR]

[16] Marden Pasinato, Carlos Eduardo Mello, Marie-Aude Aufaure, and Geraldo Zimbrao. 2013. Generating synthetic data for context-aware recommender systems. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*. IEEE, 563–567.

[17] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. ACM, New York, NY, USA, 2219–2228. https://doi.org/10.1145/3219819.3220088

[18] Manel Slokom, M Larson, and Alan Hanjalic. 2019. Data masking for recommender systems: Prediction performance and rating hiding. In *RecSys 2019 Late-breaking Results*. CEUR, 5 pages.

[19] Nasim Sonboli, Farzad Eskandanian, Robin Burke, Weiwen Liu, and Bamshad Mobasher. 2020. Opportunistic Multi-Aspect Fairness through Personalized Re-Ranking. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization* (Genoa, Italy) *(UMAP '20)*. Association for Computing Machinery, New York, NY, USA, 239–247. https://doi.org/10.1145/3340631.3394846

[20] Özge Sürer, Robin Burke, and Edward C Malthouse. 2018. Multistakeholder recommendation with provider constraints. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 54–62.

[21] Udi Weinsberg, Smriti Bhagat, Stratis Ioannidis, and Nina Taft. 2012. BlurMe: Inferring and obfuscating user gender based on ratings. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 195–202.

[22] Sijin Zhou, Xinyi Dai, Haokun Chen, Weinan Zhang, Kan Ren, Ruiming Tang, Xiuqiang He, and Yong Yu. 2020. Interactive recommender system via knowledge graph-enhanced reinforcement learning. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 179–188.

[23] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, New York, 2810–2818.