Synthetic data and simulators have the potential to markedly improve the performance and robustness of recommendation systems. These approaches have already had a beneficial impact in other machine-learning driven fields. We identify and discuss a key trade-off between data fidelity and privacy in the past work on synthetic data and simulators for recommendation systems. For the important use case of predicting algorithm rankings on real data from synthetic data, we provide motivation and current successes versus limitations. Finally we outline a number of exciting future directions for recommendation systems that we believe deserve further attention and work, including mixing real and synthetic data, feedback in dataset generation, robust simulations, and privacy-preserving methods.

Additional Key Words and Phrases: synthetic data, simulators, dataset fidelity, privacy preservation

ACM Reference Format:
Adam Lesnikowski, Gabriel de Souza Pereira Moreira, Sara Rabhi, and Karl Byleen-Higley. 2021. Synthetic Data and Simulators for Recommendation Systems: Current State and Future Directions. In SimuRec 2021: Workshop on Simulation Methods for Recommender Systems at ACM RecSys 2021, October 2nd, 2021, Amsterdam, NL and online. ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

Synthetic data generation and simulation techniques have been popular and successful in machine learning areas such as computer vision [25] [17] and robotics [24] [14], but have not been broadly explored for recommender systems, with [1, 21, 22] being notable exceptions. These techniques have the potential to address problems such as the lack of publicly available large datasets for research outside of industry. This problem is motivated by companies’ concerns about user privacy, and the possibility of revealing strategic internal KPIs, such as the level of user engagement and their recurrence in the service, or the growth of the company’s item catalog over time. Synthetic data, when representative of real data, can enable researchers to benchmark, evaluate their methods on datasets of the scale and complexity used in commercial applications. The question remains however, to what extent can simulated data effectively balance the trade-off between being close enough to the real data to act as an effective surrogate, while not being close enough to the real data to leak sensitive personal information. Simulators can potentially generate an infinite amount of synthetic data at very little cost. They bring opportunities for design decisions on recommender systems and algorithms. An algorithm

*Work for this paper, and paper submission, done while first author at NVIDIA.
designer could simulate different patterns of user behaviour to evaluate and compare recommendation algorithms, or could emulate the feedback loop between users and recommendations.

The complexity of simulators proposed for recommendation systems varies from simple to complex, depending on its purpose, whether for algorithm comparison, framework development, or simulating the feedback loop between recommendations and user interactions. We explore and motivate these uses, and highlight exciting future directions, in this work.

1.1 Past Successes of Synthetic Data and Simulators

Domains such as computer vision outside of recommendation systems have benefited markedly from applying machine learning on synthetic or simulated data [17][14]. Synthetic data in computer visions tasks such as object detection and segmentation has proven to be an effective strategy to increase model validation set performance [24], and in particular, domain randomization has been surprisingly effective [25].

Domain randomization is a technique where simulator parameters are sampled from values which are known to be unrealistic. For example training with domain randomization in a physical simulator would include training with uncommon or extreme values for friction, gravity, and object density. This approach might a priori seem detrimental to final model performance by unduly focusing model performance on cases not expected to be encountered in the intended application. Nonetheless this approach of domain randomization has been shown to be effective [24] [25].

One explanation for the effectiveness of this technique is that the benefit from a large increase of training diversity outweighs any negative effects from data-distribution shifts towards physically unrealistic scenarios, while another explanation is that the training curriculum is purposefully made more difficult than the intended application, so that the intended application is an easier, more simpler sub-problem than that encountered during the totality of training. In recommendation systems, domain randomization might include setting simulator parameters for users that are known to be unrealistic, such as browsing time, budgets, or spending habits, in order to increase training diversity, so long as validation set performance is improved. We believe this past success is a cause of optimism for the use of simulators and synthetic data in recommendation systems.

2 FIDELITY VERSUS PRIVACY TRADE-OFF

2.1 Fidelity

Synthetic data should share some statistical properties of real data. There have been a number of past approaches to capture real data fidelity in synthetic data. For instance [1] build synthetic clickstreams through graph walks that explicitly remain faithful to transition probabilities between items and co-occurrence probabilities of items appearing in the same clickstream. The authors measure synthetic dataset fidelity by performance on downstream tasks, such as training recommendation systems on their synthetic data and analyzing the performance of these trained models. Past work [2] evaluate their data-generation method on MovieLens 20M by comparing the item-wise and user-wise rating sums, as well as the singular values, of their generated data matrix versus the real data matrix, in addition to baselines such as histograms of movie ratings between real and generated data. Past work [11] evaluates dataset fidelity of generated time-series data by auto correlation metrics, distribution of generated labels and categorical event types, as well as the prediction of baseline recommendation system algorithm rankings by training and testing on synthetic data. In general it is important that features have similar distributions, but this requirement is sensitive to the intended usage of synthetic data. When synthetic data is used to test recommender systems framework and tools
based on neural networks, the cardinality and frequency distribution of categorical features is especially important, as they are represented by embeddings in the model. High-cardinality categorical features result in very large embedding tables that may exceed the capacity of a single GPU memory. That poses engineering challenges, like distributing those huge embedding tables to multi-GPU and minimizing the inter-communication between GPUs by caching the embeddings of popular categorical values, issues addressed by the HugeCTR framework for example. On the other hand, these requirements are much more strict for the purpose of comparing different algorithms, as they in general learn patterns from the conditional dependency among the features and the prediction target. Hence ignoring these conditional probabilities between features would be far from a realistic scenario. We discuss this requirement more in Section 3 below.

2.2 Privacy

Another research direction of synthetic data generation focuses on using statistical disclosure control techniques to transform original data by hiding specific information. The scale of the resulting data remains the same, but personalized information about user’s preferences is masked to protect his privacy. These past works can be classified into two categories: attack modelling and differential privacy. Attack modelling aims to identify all types of threats, which can take at least three forms: identity disclosure, attribute disclosure and inferential disclosure. Once one or multiple of these threats are identified, synthesized data is generated to prevent attacks. The main limitation of such methods is the requirement of identifying beforehand the attacker’s capabilities and goals, a requirement which is very challenging in practice. In RecSys, previous studies have focused on the classification of recommender attack models and designed simple proof-of-concept models to evaluate if the relative performance of algorithms when trained and tested on the synthesized data matches with the relative performance of algorithms trained and tested on the original data. By contrast differential privacy aims to prevent attackers from gaining information about their targets, even if the attacker has knowledge about the dataset. One approach towards differential privacy consists of injecting probabilistic noise in the original data while maintaining the same probability distributions. In RecSys differential privacy has been applied to matrix factorization (MF) at different levels of modelling: input perturbation of user-item matrices, private MF optimizers via gradient descent or alternating least squares solvers, and output perturbation. Previous work conducts experiments to evaluate the trade-off between the noise perturbation approaches and MF algorithm accuracy and demonstrated that input perturbation ensures the highest performance. However the authors point out that high degree of noise motivated by ensuring high-level privacy directly impacts the relative ranking of models’ performances. Most recent works are extending differential privacy methods to complex deep recommender systems such as wide and deep architectures and collaborative bandits learning.

3 PREDICTING ALGORITHM RANKINGS

3.1 Motivations

There are very few publicly available high-quality datasets, in terms of size and diversity of available features, likely due to companies concerns on having user privacy or public data that can leak internal company metrics. This scenario limits the research advances in the RecSys field, as scientists outside popular online service companies cannot assess and compare their proposed algorithms on large datasets. Synthetic generation can be an approach for companies

---

1https://github.com/NVIDIA/HugeCTR
to release data which is similar to its large real data but does not leak this sensitive information, so that third-party researchers can evaluate their proposed algorithms.

3.2 Successes and Limitations

Past works have shown that one can successfully predict what model performance ranking on datasets are, given model performance rankings on simulated and synthetic datasets. For instance [11] show a successful prediction of the relative performance rankings of various recommendation systems algorithms trained and tested on real data, obtained by training and testing on synthetic data. Here the algorithm ranking of the authors’ proposed GAN-based data-generation method on two different datasets is perfectly aligned, with a correlation ranking of 1.00, with the actual performance among five other algorithms trained and tested on real data. Similarly [21] provides successful results on the prediction of real algorithm rank orderings from synthetic algorithm rank orderings among three other recommendation system algorithms trained and tested on real data. On the other hand, [1] provide inconclusive or contradictory evidence that algorithm rank orderings may be successfully predicted for click-stream algorithms, at least for the probabilistic graph walk dataset generation method that the authors propose in this past work.

4 FUTURE DIRECTIONS FOR SIMULATORS AND SIMULATED DATA

4.1 Data Augmentation, Mixing Synthetic and Real Data for Recommendation Systems

Data augmentation techniques have been shown to outperform purely synthetic or purely real data in machine learning. In computer vision, techniques like random cropping, image mirroring, and color shifting have helped models to generalize better and to achieve improved accuracy [20]. Similar strategies have been proposed for raw signals and audio spectograms, such as perturbation and noise injection [10, 13], as well as in NLP [28]. However augmentation techniques for recommender systems have been largely unexplored, with [6, 27, 29] being some notable exceptions. We believe mixing synthetic and real data can be a promising direction for domains or recommender systems deployments, especially early stages of data collection and small dataset size scenarios. For research scientists outside large online services companies, it would be very helpful for synthetic data generation to augment real small data, allowing an accurate emulation of algorithm behavior on datasets larger than currently available.

4.2 Feedback in Dataset Generation

In production recommendation systems, there is often a back-and-forth process between dataset generation and model training [19]. In particular a model is likely to be trained on data that a previous model iteration solicited by providing some action that the model selected, like recommending a particular item to a user. This feedback cycle may be positive for performance, in selecting data that future iterations of model training find useful for increasing performance, or this feedback cycle may be pernicious, in either halting or reversing model performance [18]. This latter phenomenon may occur by selecting data points which are repetitions in the existing dataset, or more broadly, by focusing on short-term utility rather than promoting dataset coverage or diversity [5]. One concrete example of this latter phenomenon is when a recommendation system recommends a small number of highly popular items, and hence fails to build diverse datasets for future model training iterations. We believe that this back-and-forth process that occurs in commercial applications of recommendation systems, but typically does not in the academic or open-source study of recommendation systems, should be more placed at a higher priority for future recommendation systems research.
4.3 Robust Simulations

Generating high-fidelity synthetic data from real data may not be fully feasible, due partly to biases introduced in the real data by the policies under which the data was collected, and partly due to our imprecise understanding of what dataset properties state-of-the-art models model. We believe that robust simulators are a promising approach for these concerns. For exploring what patterns models are capable of capturing, one can generate synthetic data with known, but not necessarily realistic, properties, towards understanding how the feedback loop between users and algorithms would evolve over time, and which recommendation algorithms would perform better in such scenarios. Most public datasets do not include the information necessary to tell apart the actual preferences of the user base and the biasing effects of what the recommender presented to users, which makes unbiased evaluation of new models difficult. By evaluating on MovieLens and other public datasets, RecSys as a field has overfit to whatever policy was used during data collection for the MovieLens and other canonical RecSys datasets. In the robust simulators that we envision, we can make explicit assumptions about what the distribution of true preferences in the user population are, select a particular known logging policies, and generate a set of observations with known properties. Simulation allows system administrators to model recommender system dynamics over time.

Simulators can be used to test how generalizable are the proposed recommendation algorithms with respect to edge cases not frequent in real datasets. Simulator interpretability allows us to better understand the effect of model parameters we want to test. One can choose a distribution of true user interests and an observation sampling policy such that we end up with a dataset that has comparable statistics to MovieLens for example, then expand the simulated data to whatever large size is desired. We do not think RecSys has yet achieved this vision for robust simulators, but simulation frameworks such as RecoGym\[16\] and RecSim\[9\] are promising approaches for this vision.

4.4 Privacy-Preserving Methods

Motivated by preserving users sensitive data and global statistics related to business KPIs, large companies have not shared large-scale datasets with external communities. Privacy-preserving methods discussed in section 2.2 constitute a promise for guaranteeing privacy while releasing large scale datasets for research development. However it is still unclear how these noisy aggregated data impact learning effective recommender system models that maintain the relative ranking of different approaches for performance comparison and model selection. A recent Criteo challenge organized in collaboration with the CAP21 conference aims to benchmark models defined using private constrained training data to explore the trade-off between privacy level and prediction performance. In particular individual data is transformed through an embedded, anonymized, and compact representation. Then machine learning models are trained and tested on two objectives: the privacy attacks protection and the outcome prediction task. If such privacy functions are demonstrated to lead to high performance machine learning models, large companies may generate large anonymized synthetic data using a given privacy function, and more openly share it with the RecSys community.

5 CONCLUSION

In this paper, we motivate and state the uses of synthetic data and simulators for recommendation systems. The success of these approaches in other machine learning fields provide promise for these methods. For approaches that use real data, we identify a key trade-off between data fidelity and privacy. The important use case of predicting algorithm rankings using synthetic data is well-motivated, and has had both successes and limitations. Finally there are a number

\[^{2}\text{https://medium.com/criteo-engineering/criteo-cap21-privacy-preserving-ai-challenge-9cf9cd88e54}\]
of exciting and promising future directions we believe the field should invest in, including data augmentation that mixes real and synthetic data, feedback in dataset generation in production systems, robust simulators, and privacy-preserving methods.

REFERENCES

[1] Nino Antulov-Fantulin, Matko Bošnjak, Vinko Zlatić, Miha Grčar, and Tomislav Šmuc. 2014. Synthetic Sequence Generator for Recommender Systems—Memory Biased Random Walk on a Sequence Multilayer Network. In International Conference on Discovery Science. Springer, 25–36.

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

[3] Arnaud Berloz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovski. 2015. Applying Differential Privacy to Matrix Factorization. (RecSys ’15). Association for Computing Machinery, New York, NY, USA.

[4] R. Burke, B. Mobasher, Roman Zabicki, and Runa Bhaumik. 2004. Identifying Attack Models for Secure Recommendation.

[5] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhard. 2018. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In Proceedings of the 12th ACM Conference on Recommender Systems. 224–232.

[6] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Juklir Sarla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in your inbox: Product recommendations at scale. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 1809–1818.

[7] Huifeng Guo, Wei Guo, Yong Gao, Ruiming Tang, Xiaqiang He, and Wenzhi Liu. 2021. ScaleFreeCTR: MixCache-based Distributed Training System for CTR Models with Huge Embedding Table. Proceedings of SKIR’21.

[8] Udit Gupta, Carole-Jean Wu, Xiaodong Wang, Maxim Naumov, Brandon Reagen, David Brooks, Bradford Cottel, Kim Hazelwood, Mark Hempstead, Bill Jia, et al. 2020. The architectural implications of facebook’s dnn-based personalized recommendation. In 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 488–501.

[9] Eugene Ie, Chih-wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutiller. 2019. Recsim: A configurable simulation platform for recommender systems. arXiv preprint arXiv:1909.04847 (2019).

[10] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In Sixteenth annual conference of the international speech communication association.

[11] Zinan Lin, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. 2020. Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions. In Proceedings of the ACM Internet Measurement Conference (Virtual Event, USA). (IMC ’20). Association for Computing Machinery, New York, NY, USA, 464–483. https://doi.org/10.1145/3419394.3423643

[12] Frank McSherry and Bya Mironov. 2009. Differentially Private Recommender Systems: Building Privacy into the Netflix Prize Contenders (KDD ’09).

[13] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zooph, Ekin D Cubuk, and Quoc V Le. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. arXiv preprint arXiv:1904.08779 (2019).

[14] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. 2019. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In 2019 International Conference on Robotics and Automation (ICRA). IEEE, 7249–7255.

[15] Fatemeh Rezaimehr and Chitra Dadikeh. 2021. A survey of attack detection approaches in collaborative filtering recommender systems. Artificial Intelligence Review (2021).

[16] David Rohde, Stephen Bonner, Travis Dunlop, Flavian Vastile, and Alexandros Karatzoglou. 2018. Recogym: A reinforcement learning environment for the problem of product recommendation in online advertising. Proceedings of the REVEAL workshop at the Twelfth ACM Conference on Recommender Systems (RecSys ’18).

[17] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. 2018. Learning from synthetic data: Addressing domain shift for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3752–3761.

[18] Sven Schmit and Carlos Riquelme. 2018. Human interaction with recommendation systems. In International Conference on Artificial Intelligence and Statistics. PMLR, 862–870.

[19] Burr Settles. 2009. Active learning literature survey. (2009).

[20] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. Journal of Big Data 6, 1 (2019), 1–48.

[21] Manel Slokom. 2018. Comparing recommender systems using synthetic data. In Proceedings of the 12th ACM Conference on Recommender Systems. 548–552.

[22] Manel Slokom, Martha Larson, and Alan Hanjalic. 2020. Partially Synthetic Data for Recommender Systems: Prediction Performance and Preference Hiding. arXiv preprint arXiv:2008.03797 (2020).

[23] Manel Slokom, Martha A. Larson, and Alan Hanjalic. 2020. Partially Synthetic Data for Recommender Systems: Prediction Performance and Preference Hiding. CoRR abs/2008.03797 (2020). arXiv:2008.03797 https://arxiv.org/abs/2008.03797
[24] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, et al. 2018. Domain randomization and generative models for robotic grasping. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 3482–3489.

[25] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. 2018. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 969–977.

[26] Huazheng Wang, Qian Zhao, Qingyun Wu, Shubham Chopra, Abhinav Khaitan, and Hongning Wang. 2020. Global and Local Differential Privacy for Collaborative Bandits. In Fourteenth ACM Conference on Recommender Systems (RecSys ’20). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3383313.3412254

[27] Qinyong Wang, Hongzhi Yin, Hao Wang, Quoc Viet Hung Nguyen, Zi Huang, and Lizhen Cui. 2019. Enhancing collaborative filtering with generative augmentation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 548–556.

[28] Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019).

[29] Matthias Wölbitzch, Simon Walk, Michael Goller, and Denis Helic. 2019. Beggars can’t be choosers: Augmenting sparse data for embedding-based product recommendations in retail stores. In Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization. 104–112.

[30] Huanyu Zhang, Ilya Mironov, and Meisam Hejazinia. 2021. Wide Network Learning with Differential Privacy. CoRR abs/2103.01294 (2021). arXiv:2103.01294 https://arxiv.org/abs/2103.01294