Realistic but Non-Identifiable Synthetic User Data Generation

Isaac Noble[1], Ivan Vendrov[2], Xin Xu[1] and Deepak Ramachandran[1,†]

[1]*Google Research*
[2]*Anthropic*

Abstract

The release of large-scale synthetic datasets would greatly accelerate research in recommender systems. However, there is a fundamental tension between the requirements of high quality recommendations needing close knowledge of user behavior, and privacy, which is endangered by excessive disclosure of real user behavior. We propose to resolve this tension by introducing two novel metrics to compare synthetic recommender datasets: *Identifiability*, which measures susceptibility to multiple different attacks on privacy and *Realism*, which is a comparative (as opposed to an absolute) measure of recommender performance (between real and synthetic datasets). We do an extensive evaluation of 7 data generation algorithms for movie and song recommendations in 28 different settings, from which we construct Pareto frontiers of Realism vs Identifiability. This reveals multiple insights into the performance of different synthetic data generation methods along different points on this curve. We discuss how these insights can guide future research on synthetic data generation.

1. Introduction

Recent revolutions in Computer Vision and Natural Language Processing are often credited to the use of massive datasets such as ImageNet [1] and Common Crawl [2], which dramatically improved performance across a multitude of tasks, and completely changed the algorithmic landscape by elevating data-intensive overparameterized models. In contrast, academic recommender systems have seen much less algorithmic progress at this scale, with recent work [3] showing that linear methods invented decades ago are still competitive. One reason for this gap is the scarcity of equivalently-sized public recommender datasets; while industrial recommender systems are often trained on trillions of user interactions, much academic research is still evaluated on datasets like MovieLens [4] with tens of millions of interactions or fewer.

The key limiting factor is the requirement of user privacy: unlike image and language datasets, releasing datasets of user interactions could lead to private information about individual users being leaked. Simply stripping any identifying information from the dataset before sharing publicly is insufficient: Narayanan and Shmatikov [5] describe an attack using side-channel information that de-anonymizes 99% of users in the Netflix Prize dataset. Attacks like these

2nd Edition of EvalRS: a Rounded Evaluation of Recommender Systems, August 6 - August 10, 2023, Long Beach, CA, USA

[†]Senior author.

isaacn@google.com (I. Noble); ivendrov@gmail.com (I. Vendrov); xxujasmine@google.com (X. Xu); ramachandrand@google.com (D. Ramachandran)

are a major reason for the relative of large-scale user interaction datasets released since the Netflix Prize challenge, although efforts such as the SIGIR eCOM Challenge [6] and the Recsys Challenge [7] have remedied this somewhat.

The most promising approach in our view is to create datasets of *synthetic* user interactions [8, 9] satisfying two properties:

Private : Very little information about individual real users should be derivable from the synthetic datasets. We translate this into measures of robustness against three possible attacks recognized in the literature: *membership* [10, 11], *de-anonymization* [12, 5], and *attribute inference* [13, 14, 15]. We propose an *Identifiability* metric for each attack method that allows for comparison across models and datasets.

Realistic : In many cases, the goal of organizations that release synthetic datasets is not to have external researchers develop a state-of-the-art recommendation system for their needs, but instead to spur research on effective recommendation systems. In this scenario, the absolute performance achieved on the synthetic dataset is not of central importance. Instead, we want to ensure that the conclusions of empirical analyses on synthetic data (e.g. relative performance of two offline recommendation algorithms) are likely to hold on real data. Thus, we introduce a new metric called *Realism* for synthetic datasets that measure whether such conclusions can be reliably transferred from synthetic to real data.

We propose that synthetic datasets and synthetic dataset generation algorithms (in the recommendation domain) should be evaluated by a combination of Identifiability and Realism metrics. By contrast, evaluation using a combination of privacy metrics and absolute recommendation performance would be less informative because these 2 objectives are to a significant degree contradictory: Achieving high absolute recommendation performance (for real users) by training on synthetic data alone correlates strongly with an understanding of real user behavior, implying a certain amount of privacy leakage.

Realism can also have a similar tension with Identifiability but as our empirical investigations show, different synthetic data generation algorithms can make different tradeoffs along these axes, and a Pareto frontier emerges connecting Pareto-optimal tradeoff points representing various synthetic generation algorithms and hyperparameter settings.

The primary goal of our work is to understand the shape of this Realism-Identifiability Pareto frontier and to push it forward so that increasingly realistic synthetic datasets can be safely released publicly, narrowing the gap between industrial and academic practice. To that end, this paper makes the following contributions:

1. For any synthetic data generation algorithm for recommendation systems, we propose 3 separate Identifiability metrics that quantify robustness against each of the 3 privacy attacks we address in this paper, and a measure of Realism that captures transferability of empirical conclusions from synthetic to real data.
2. We perform a comprehensive set of experiments comparing 7 synthetic user generation methods across 4 recommender algorithms using 7 recommender metrics, for 2 domains (movies and songs); which we use to construct Pareto frontiers of synthetic data generation algorithms. A number of insights can be derived from these curves, such as the surprising

weakness of VAE methods and the general robustness of specific GAN architectures such as R-GAN [16].

3. We introduce a new approach to synthetic data generation, *IdentityGAN* (a variant of TimeGAN [8]) which is Pareto-optimal on these metrics along many sections of the Pareto frontier and may be treated as the preferred starting point for future research on synthetic data generation. [1]

Due to space constraints, related work and additional experimental evaluations are presented in the Appendix.

2. Preliminaries

We start by defining our recommendation system setting and formalizing the synthetic dataset generation problem. We assume a finite universe of d items that may be consumed by users. A *user* in our framework is simply identified by a binary vector u of dimension d where a 1 indicates that the user has rated that item positively. In our setting, we are provided a *real user dataset* D of users u, split into train and test sets $D_{\text{train}}, D_{\text{test}} \subset D$. We are expected to construct a *synthetic user dataset* $\hat{D}$ of synthetic users $\hat{u} \notin D$, that allows us to study the behavior of recommendation algorithms on D, without leaking information about individual users (we define these objectives precisely in Sections 3 and 4).

As a strawman metric, we will define $\text{AbsPerf}_{R,M}$ to be the (absolute) performance of recommendation algorithm R on some metric M (e.g. Recall@10) when trained on $\hat{D}$ and evaluated on D_{test}. See Sections 5.1 and B.3 for the full list of recommenders and metrics we consider.

3. Measuring Realism

Our criteria for realism is that the results of empirical analyses of synthetic data are very likely to hold for the real data. The most important type of empirical analysis commonly conducted by recommender systems researchers is a *relative performance comparison* of two or more recommendation algorithms on some dataset using some offline metric, yielding the conclusion that one algorithm significantly outperforms the other. So we seek to operationalize the realism of synthetic dataset $\hat{D}$ as the *probability that if a recommender algorithm outperforms another on $\hat{D}$, it will also outperform on D.*

More formally, let $\mathscr{R}$ be a distribution over recommender algorithms R, and $\mathscr{M}$ a distribution over recommender metrics, where a metric $M \sim \mathscr{M}$ is a mapping $M(R, D_{\text{train}}, D_{\text{test}}) \to \mathbb{D}(\mathbb{R})$, from a recommender and dataset split to a distribution over scalars. A distribution is needed to account for the inherent stochasticity of the measurement process due to inference-time dropout etc. We set up a hierarchical Bayesian model as follows to sample recommenders, metrics and measurements:

$$R_1, R_2 \sim \mathscr{R}$$
$$M \sim \mathscr{M}$$

[1] All code, models, and reproducibility data will be shared along with the camera-ready version.

$$m_i \sim M(R_i, D_{\text{train}}, D_{\text{test}})$$
$$\hat{m}_i \sim M(R_i, \hat{D}_{\text{train}}, \hat{D}_{\text{test}})$$

A preliminary measure of realism might be given by $P(m_1 > m_2 \mid \hat{m}_1 > \hat{m}_2)$ i.e. the probability that the relative performance on synthetic data is preserved on real data. Note that this is equivalent to the Kendall Rank Correlation between performance on D and $\hat{D}$ linearly rescaled to the range $[0, 1]$. To provide a stronger measure and reduce noise we only care about large and statistically significant differences in performance, so we conduct two-sample one-tailed Student's t-tests without assuming equal variance to determine if results are significant:

$$T_\sigma(m_1, m_2) = \begin{cases} 1, & \text{p-value of hyp. } [m_1 > m_2] \text{ is } < \sigma \\ 0, & \text{otherwise} \end{cases}$$

This leads to our precise definition of *Realism*, which is the *fraction of measurements where a significant result on synthetic data is preserved on real data*:

$$Realism_\sigma(\hat{D}, D) = \frac{\mathbb{E}_{\substack{m_1, m_2 \\ \hat{m}_1, \hat{m}_2}} [T_\sigma(m_1, m_2) \cdot T_\sigma(\hat{m}_1, \hat{m}_2)]}{\mathbb{E}_{\hat{m}_1, \hat{m}_2} [T_\sigma(\hat{m}_1, \hat{m}_2)]} \tag{1}$$

For the rest of the paper, we assume a uniform distribution over a finite set of algorithms (treating hyper-parameter variations as separate algorithms) and metrics and use $\sigma = 0.01$ in all our experiments.

Operationally, in order to estimate the variance of the different algorithms, each uniquely parameterized algorithm is trained N times on both D_{train} and $\hat{D}_{\text{train}}$, and the results evaluated on D_{test} and $\hat{D}_{test}$. An all-pairs comparison is done using the N samples to compute the Student t-statistic, all statistically significant results on the synthetic data are then checked for significance on the real data. For all experiments, N was set at 10.

4. Measuring Identifiability

A key factor restraining commercial entities from widely sharing datasets of user behavior, is the threat of releasing private or sensitive user data. Even in many cases where attempts are taken to anonymize or remove sensitive data before widely sharing, it has been shown that the identities or sensitive data can be re-constructed by adversarial attacks with minimal side information – most notably in the case of the Netflix challenge [5]. By releasing synthetic datasets, these concerns could potentially be significantly alleviated. However, to be useful for recommendation research, the synthetic data must have some relationship to the behavior of real users, and this represents an opportunity for sophisticated adversaries to extract privacy-threatening information about real users.

Recommender systems researchers have generally favored rigorous definitions of privacy that come with provable guarantees such as differential privacy [17, 18], which ensures that the output of a computation does not allow inference of any record from the original input. However, practitioners have argued that these are in some cases too strong and limits the usage of many useful algorithms and applications [19], "requiring unconditional privacy guarantees against

computationally unbounded adversaries" [20]. We hold that such arguments are particularly valid for the release of derivative synthetic datasets, since these have an extra level of indirection from real data.

In the rest of this section, we outline 3 different privacy-compromising attacks that have been described in the literature, based on strong (but not unbounded) adversary models. These attacks were originally designed in settings where either a database or ML model was available to the adversary, but we adapt them to our setting where only synthetic data is available. For each, we derive a corresponding metric that measures the susceptibility of a synthetic dataset to that kind of attack. We refer to this collection of metrics as *Identifiability*.

This catalog of attacks and corresponding identifiability metrics are not intended to be provably exhaustive of the possibilities, but we believe we have covered the most commonly-reported and well-studied scenarios. If new attacks are proposed, then it should be reasonably straightforward to extend our framework by defining a corresponding identifiability metric, adding it to the evaluation suite, and re-assessing our conclusions on the robustness of synthetic data generation algorithms.

4.1. Membership-Identifiability

The first identifiability measure addresses the threat of discovering a particular user's presence in a particular dataset:

Definition 1 (Membership Attack [10, 11])**.** *We assume an adversary who has perfect knowledge of all the attributes of a real user u (i.e. their ratings for each item). A membership attack is an attempt to infer from the synthetic dataset $\hat{D}$, whether $u \in D$.*

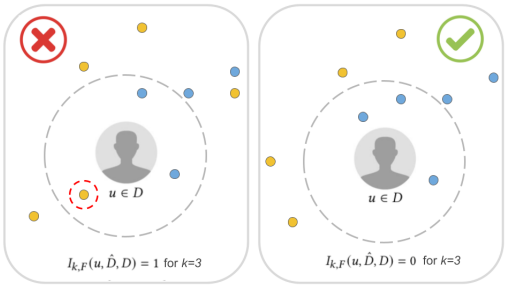


Figure 1: Illustration of Membership-Identifiability. Blue nodes represent users in the real dataset D, and yellow nodes users in the synthetic dataset $\hat{D}$. The k-identifiability (k=3) of the user $u \in D$ in the left figure is 1 because there is one user in $\hat{D}$ among its k-nearest neighbors. In the right figure, all 3 nearest neighbors of u are from D, so $I_{k,F}(u, \hat{D}, D) = 0$.

This can be a privacy threat if the very existence of a user in a particular dataset can be used to infer privacy-compromising information about them, e.g. if a dataset is known to have been constructed in a way that includes or excludes certain protected categories of users. Membership attacks were originally introduced in the context of deriving membership information about users in a dataset from Machine Learning models trained on them [10]; here we extend the definition to synthetic datasets as a source.

For our Membership-Identifiability measure, we expand upon the notion of ϵ-Identifiability from Yoon et al. [21], although the precise justification using Membership attacks is novel. Assume a distance metric F between users, and a nearest neighbor function $N_{k,F}(u, D)$ which returns the nearest (under F) set of k users in D to user u. Then, the *k-identifiability* of a user $u \in D$ w.r.t. $\hat{D}$ is defined as:

$$I_{k,F}(u, \hat{D}, D) = \begin{cases} 1, & \text{if } N_{k,F}(u, D \cup \hat{D}) \cap \hat{D} \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$I_{k,F}$ is an indicator function that is 1 when there exists a $\hat{u} \in \hat{D}$ that is among the k nearest neighbors of $u \in D$ (See Fig 1). A user who is not k-identifiable for a sufficiently high value of k can be regarded as safe from a membership attack, since an adversary who knows all the user's attributes (i.e. item ratings) cannot distinguish him from k other users who may or not be present in the dataset.

Finally, the *Membership-Identifiability of dataset* D at k under F and w.r.t. $\hat{D}$ is:

$$\text{Identifiability}_{k,\hat{D},F}(D) = \frac{\sum_{u \in D} I_{k,F}(u, \hat{D}, D)}{|D|} \tag{3}$$

This gives the fraction of users $u \in D$ that have synthetic users $\hat{u} \in \hat{D}$ within their k nearest neighbors.

4.2. Deanonymization-Identifiability

Definition 2 (Deanonymization Attack [12, 5])**.** *We assume an adversary who has oracle knowledge that a particular synthetic user, $\hat{u} \in \hat{D}$ is close to a real user $u \in D$. A de-anonymization attack is an attempt to gain maximum information about u's attributes (i.e. ratings) from $\hat{u}$.*

We base our metric for measuring Deanonymization-Identifiability on the Mutual Information between the ratings of u and $\hat{u}$. Concretely, we use the generative model in Figure 2a to cast the ratings of u and $\hat{u}$ as random variables. We assume some distribution over items $I \sim \mathcal{I}$, and then the ratings $r_{u,I}$ and $r_{\hat{u},I}$ that users u and $\hat{u}$ assign to the item I are deterministic functions of I. *Deanonymization-Identifiability* is defined as the mutual information between the marginal distributions of $r_{u,I}$ and $r_{\hat{u},I}$:

$$MI(r_{u,I}; r_{\hat{u},I}) = \sum_{r_{u,I} \in \{0,1\}} \sum_{r_{\hat{u},I} \in \{0,1\}} p(r_{u,I}, r_{\hat{u},I}) \log \left(\frac{p(r_{u,I}, r_{\hat{u},I})}{p(r_{u,I}) p(r_{\hat{u},I})} \right)$$

These marginals can be computed by the sum rule:

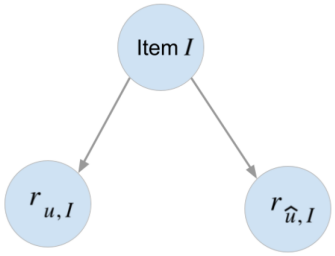
$$p(r_{u,I} = a, r_{\hat{u}} = b) = \sum_{\text{item } i} p(r_{u,I} = a, r_{\hat{u}} = b | i) P(i)$$

$p(r_{u,I} = a, r_{\hat{u}} = b|i)$ is now fully specified as a function of i: it is 1 if the ratings by u and $\hat{u}$ are a and b, and 0 otherwise.

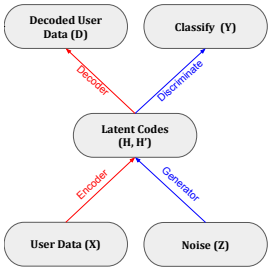
Finally, we must specify the (prior) item distribution. The simplest (i.e. uninformative) approach is to use the uniform distribution over items, but instead we propose (similar to section 4.1) to use:

$$P(i) \propto \frac{1}{log(ItemFreq)}$$

again, because items that are more rare impart more information.



(a) The probabilistic graphical model used to define the mutual information between ratings of the real and synthetic user (for Deanonymization-Identifiability). Items are selected by some prior distribution. Ratings of real and synthetic user are deterministic functions of I, but $r_{u,I}$ and $r_{\hat{u},I}$ are random variables.



(b) The TimeGAN model for generating synthetic data is a combination of a Discrete Conditional Encoder that embeds users into a Latent space and a GAN which distinguishes real data points from synthetic ones.

In practice, we calculate a real user's Deanonymization-Identifiability by averaging the mutual information w.r.t. the 50 synthetic users that share the most items in common with the real user. We found this metric to be more stable than using the maximum mutual information from any single synthetic user.

4.3. Attribute-Identifiability

Definition 3 (Attribute Attack [13, 14, 15])**.** *We assume an adversary who has oracle knowledge of some attributes (i.e. ratings) of user $u \in D_{train}$. An attribute attack is an attempt to gain the maximum possible information about u's remaining attributes (i.e. ratings) from $\hat{D}_{train}$.*

Observe that an attribute attack on recommendation datasets is equivalent to solving the recommendation problem for a specific subset of users, namely the ones in the original dataset, D_{train}. Therefore, our metric for Attribute-Identifiability will reduce to a measure of *recommendation effectiveness for users in D_{train}, when trained on users in $\hat{D}_{train}$*. In particular, we choose to use Recall@20 (one of the simplest and most popular metrics) over all $u \in D_{\text{train}}$ using EASER (a state-of-the-art recommender; See sections 5.1 and B.3 for more details).

Attribute-Identifiability can be viewed as analogous to a measure of overfitting in traditional Machine Learning, but instead of being computed directly on the original training set, there is an extra level of indirection from the synthetic user training set $\hat{D}_{\text{train}}$ to the real user training set D_{train} from which $\hat{D}_{\text{train}}$ was derived.

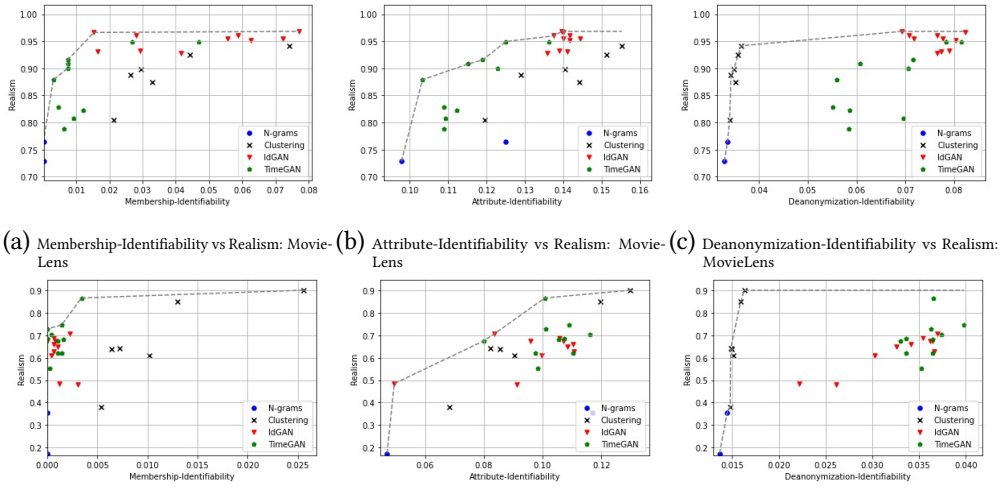
5. Experimental Evaluation

We conduct a comprehensive large-scale evaluation of 6 major families of synthetic data generation algorithms (described below) in 2 different recommendation domains, MovieLens [22] and MillionSongs [23]. Also, for computing the realism metric, we need multiple recommender algorithms and recommendation performance metrics. For details, see Appendix B.

5.1. Synthetic Data Generation Algorithms

To capture the Pareto frontier between realism and privacy, we evaluate 4 different synthetic data generation algorithms that cover the range from simple statistical models to deep generative models with millions of parameters:

1. **Unigrams**: Our simplest generation algorithm ignores all correlational structure in the dataset. We compute the distribution over items in the dataset, and sample each user history from this distribution without replacement.
2. **Bigrams**: Here, we model pairwise correlations between items by computing the conditional probabilities $P(j|i)$ of a user's history containing item j conditional on containing item i, then generating synthetic users by recursively sampling items $i_{n+1} \sim P(\cdot|i_n)$
3. **Clustering**: Following Monti et al. [9], we generate synthetic data by clustering users using K-means with Euclidean distance between unweighted users. This is in contrast to the identifiability calculation which weights items based on frequency. We then compute the empirical distribution of items for each cluster and sample users from each cluster in proportion to the cluster size and cluster item distribution. By treating K as a hyperparameter, we generate a family of synthetic datasets with different behaviors on Identifiability vs Realism (see sec 5.2). We experiment with $K = [5, 10, 15, 20, 50, 100]$.
4. **VAE**: A natural approach to generating synthetic user data is to use the popular Variational Autoencoder (VAE) model [24]. Following Liang et al. [25], the VAE consists of an encoder and decoder, and is trained to compress the input information (i.e. the user's vector of item ratings) into a constrained multivariate latent distribution $\hat{X}$ (encoding) from which the input can be reconstructed with the smallest reconstruction error (decoding). Crucially, the latent representation $\hat{X}$ actually defines a distribution over users. We can then sample from this distribution and use the decoder to generate synthetic users that should approximate the training distribution.
5. **TimeGAN**: We implement a simplified TimeGAN model [8], originally proposed for creating synthetic medical records. This model uses an auto-encoder to build a latent representation of the real data and a generator to generate samples in this latent space while a discriminator separates real and generated latent samples. The encoder takes a sequence of integers X representing items the user has interacted with, and embeds them with a multi-layer RNN into a latent code H. A decoder then uses a multi-layer RNN to expand the H into a sequence of logits D. Cross-entropy loss between D and X is used to train the encoder and decoder. The generator is a simple network that takes a noise vector Z and uses a sequence of dense layers to form a synthetic latent code $\hat{H}$. The discriminator takes H and $\hat{H}$ and uses a sequence of dense layers to classify them as real or fake. We follow the approach of Lucic et al. [26], in automatically searching a wide range of GAN architectures, training methods and hyperparameter settings along with random restarts. After training the model, the generator and decoder networks are used to generate a set of synthetic latent codes $\hat{H}$ and then a dataset of synthetic users $\hat{D}$.
6. **IdentityGAN**: We further tested a novel and simpler GAN architecture that replaces the input item sequence X with a single token representing the user's unique id. The encoder is reduced to a sequence of dense layers embedding the user id into a latent code H. The rest of the architecture is the same as the TimeGAN implementation above.



(a) Membership-Identifiability vs Realism: MovieLens (b) Attribute-Identifiability vs Realism: MovieLens (c) Deanonymization-Identifiability vs Realism: MovieLens

(d) Membership-Identifiability vs Realism: MillionSongs (e) Attribute-Identifiability vs Realism: MillionSongs (f) Deanonymization-Identifiability vs Realism: MillionSongs

Figure 3: Pareto curves of Identifiability vs Realism.

7. **Fractal Expansion**: We follow the fractal expansion model from Belletti et al. [27] which uses Kronecker Graph expansion adapted to binary vectors. This technique re-introduces patterns observed in D_{train} into each block of local interactions of the synthetic user/item matrix in an entirely non-parametric way.

5.2. Results

Figures 3a-3f shows the results for various hyper-parameter settings of the algorithms. For extensive discussion on the conclusions that can be drawn from these curves, and further experiments please refer to Appendix C, omitted here due to lack of space.

6. Conclusions and Future Work

In this paper, we present a set of metrics that capture a fundamental trade-off in the creation of synthetic recommender datasets. Identifiability gives a operational evaluation of resilience against multiple privacy attack vectors, while Realism captures the degree to which we may rely on the results of recommendation algorithms research built on synthetic data. The Pareto curve analysis shows contributions from multiple algorithm families with no single one dominating.

We hope our contribution will spur research on synthetic data generation to expand the Pareto frontier (without over-optimizing for specific metrics) similar to how Reclist [28] expanded the role of behavioral testing in the rounded evaluation of recommender systems. We make no strong claim that our work should immediately enable organizations with stewardship of critical data to share access through synthetic data generation. Caution is warranted in this regard, and applications must be evaluated on a case-by-case basis.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[2] Common crawl dataset, https://commoncrawl.org, ????

[3] M. Ferrari Dacrema, P. Cremonesi, D. Jannach, Methodological issues in recommender systems research (extended abstract), 2020, pp. 4648–4652. doi:10.24963/ijcai.2020/642.

[4] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (2015) 19:1–19:19. URL: http://doi.acm.org/10.1145/2827872. doi:10.1145/2827872.

[5] A. Narayanan, V. Shmatikov, Robust de-anonymization of large sparse datasets, in: Proceedings of the 2008 IEEE Symposium on Security and Privacy, SP '08, IEEE Computer Society, USA, 2008, p. 111–125. URL: https://doi.org/10.1109/SP.2008.33. doi:10.1109/SP.2008.33.

[6] J. Tagliabue, C. Greco, J.-F. Roy, F. Bianchi, G. Cassani, B. Yu, P. J. Chia, Sigir 2021 e-commerce workshop data challenge, in: SIGIR eCom 2021, 2021.

[7] P. Knees, Y. Deldjoo, F. B. Moghaddam, J. Adamczak, G.-P. Leyson, P. Monreal, Recsys challenge 2019: session-based hotel recommendations, in: RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, p. 570–571. URL: https://doi.org/10.1145/3298689.3346974.

[8] J. Yoon, D. Jarrett, M. van der Schaar, Time-series generative adversarial networks, in: Advances in Neural Information Processing Systems, volume 32, 2019. URL: https://proceedings.neurips.cc/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf.

[9] D. Monti, G. Rizzo, M. Morisio, All you need is ratings: A clustering approach to synthetic rating datasets generation, ArXiv abs/1909.00687 (2019).

[10] R. Shokri, M. Stronati, V. Shmatikov, Membership inference attacks against machine learning models (2016). URL: https://arxiv.org/abs/1610.05820. arXiv:1610.05820.

[11] H. Hu, Z. Salcic, G. Dobbie, X. Zhang, Membership inference attacks against machine learning models (2021). URL: https://arxiv.org/abs/2103.07853. arXiv:2103.07853.

[12] L. Sweeney, Weaving technology and policy together to maintain confidentiality, J. of Law, Medicine and Ethics 25 (1997).

[13] M. Kosinski, D. Stillwell, T. Graepel, Private traits and attributes are predictable from digital records of human behavior, Proceedings of the National Academy of Sciences 110 (2013) 5802–5805.

[14] U. Weinsberg, S. Bhagat, S. Ioannidis, N. Taft, Blurme: Inferring and obfuscating user gender based on ratings, in: In RecSys, Dublin, Ireland, 2012.

[15] N. Z. Gong, B. Liu, Attribute inference attacks in online social networks, ACM Transactions on Privacy and Security 21 (2018) 1–30.

[16] K. Roth, A. Lucchi, S. Nowozin, T. Hofmann, Stabilizing training of generative adversarial networks through regularization, Advances in Neural Information Processing Systems 30 (2017).

[17] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation methods for the web, in:

Proceedings of the Tenth International World Wide Web Conference (WWW-01), ACM, Hong Kong, 2001, pp. 613–622.
[18] M. S. Daskin, Network and Discrete Location: Models, Algorithms, and Applications, John Wiley & Sons, Hoboken, NJ, 2011.
[19] A. Groce, J. Katz, A. Yerukhimovich, Limits of computational differential privacy in the client/server setting, in: In Proceedings of the Theory of Cryptography Conference, 2011.
[20] H. Asi, J. Duchi, O. Javidbakht, Element level differential privacy: The right granularity of privacy, arXiv preprint arXiv:1912.04042 (2019).
[21] J. Yoon, L. N. Drumright, M. van der Schaar, Anonymization through data synthesis using generative adversarial networks (ads-gan), IEEE Journal of Biomedical and Health Informatics 24 (2020) 2378–2388. doi:10.1109/JBHI.2020.2980262.
[22] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (2015). URL: https://doi.org/10.1145/2827872. doi:10.1145/2827872.
[23] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, P. Lamere, The million song dataset, in: Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011), 2011.
[24] D. P. Kingma, M. Welling, Auto-encoding variational bayes, in: 2nd International Conference on Learning Representations, ICLR, Banff, AB, Canada, 2014.
[25] D. Liang, R. G. Krishnan, M. D. Hoffman, T. Jebara, Variational autoencoders for collaborative filtering, 2018. arXiv:1802.05814.
[26] M. Lucic, K. Kurach, M. Michalski, S. Gelly, O. Bousquet, Are gans created equal? a large-scale study, arXiv preprint arXiv:1711.10337 (2017).
[27] F. Belletti, K. Lakshmanan, W. Krichene, Y.-F. Chen, J. Anderson, Scalable realistic recommendation datasets through fractal expansions, 2019. arXiv:1901.08910.
[28] P. J. Chia, J. Tagliabue, F. Bianchi, C. He, B. Ko, Beyond ndcg: Behavioral testing of recommender systems with reclist, WWW '22 Companion, Association for Computing Machinery, New York, NY, USA, 2022, p. 99–104. URL: https://doi.org/10.1145/3487553.3524215. doi:10.1145/3487553.3524215.
[29] M. Slokom, Comparing recommender systems using synthetic data, in: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 548–552. URL: https://doi.org/10.1145/3240323.3240325. doi:10.1145/3240323.3240325.
[30] M. Slokom, M. Larson, A. Hanjalic, Partially synthetic data for recommender systems: Prediction performance and preference hiding, 2020.
[31] J. Williams, A. Raux, D. Ramachandran, A. Black, The dialog state tracking challenge, in: Proceedings of the SIGDIAL 2013 Conference, Association for Computational Linguistics, Metz, France, 2013, pp. 404–413. URL: https://aclanthology.org/W13-4065.
[32] M. F. Dacrema, S. Boglio, P. Cremonesi, D. Jannach, A troubling analysis of reproducibility and progress in recommender systems research, CoRR abs/1911.07698 (2019). URL: http://arxiv.org/abs/1911.07698. arXiv:1911.07698.
[33] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, Association for Computing Machinery, New York, NY, USA, 2010, p. 39–46. URL: https://doi.org/10.1145/1864708.1864721. doi:10.1145/1864708.1864721.

[34] M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms, ACM Transactions on Information Systems 22 (2004) 143–177. URL: https://doi.org/10.1145/963770.963776.
[35] H. Steck, Embarrassingly shallow autoencoders for sparse data, CoRR abs/1905.03375 (2019). URL: http://arxiv.org/abs/1905.03375. arXiv:1905.03375.
[36] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein gans, arXiv preprint arXiv:1704.00028 (2017).

A. Related Work

Slokom [29] and Slokom et al. [30] use CART to create synthetic Movielens-100k and Goodbook datasets, and show that this preserves the relative performance of 3 recommenders, although this is a somewhat easy test because the 3 recommenders they select have huge relative gaps in performance (2-10x in terms of Recall@5) making it easy to preserve relative ordering even with low quality synthetic data. Typically, new methods in recommender systems demonstrate recall improvements on the order of 1-2%. Their setting also differs from ours in that they generate *partially* synthetic data (some original ratings and items are kept) and the attack they defend against is whether an adversary looking at the synthetic data can reliably determine the value of summary statistics like a user's favorite actor, director, or author.

Monti et al. [9] proposes a simple and flexible method of generating synthetic data by splitting users into clusters and learning summary statistics for each cluster. They evaluate 5 recommenders on both real and synthetic data and find that the relative order of performance is preserved in "almost all" cases although this is not quantified. They do not explicitly consider privacy guarantees or threat models.

To our knowledge, there has been no work that measures the Realism of synthetic data in terms of comparative performance across multiple metrics on real data. Williams et al. [31] conducts an aggregate analysis of multiple metrics on a Dialog State Tracking task, but the details differ significantly from ours, apart from its use of the Kendall-Tau coefficient.

B. Experimental Methodology

Here we give further details of the components used for the Realism vs. Identifiability measurements:

B.1. Datasets

To ensure our results generalize across domains, we performed separate sets of measurements on both Movie and Song recommendations datasets. For movies, we use the MovieLens 1 Millions dataset [22], truncated (to speed up training for the GAN RNN) to at most 20 items per user by randomly sampling. The resulting dataset consists of 4822 users over 3252 movies. For songs, we use the MillionSongs dataset [23], filtered down to users that listened to at least 70 songs more than 3 times and songs that had at least 50 different users. This was then further

truncated to 20 items per user by randomly sampling. This left 1752 songs and in order be consistent with MovieLens, 5000 users were then sampled.

B.2. Recommenders

To get robust estimates of realism, we need a diverse set of recommendation algorithms to compare. From the exhaustive benchmarking done by Dacrema et al. [32], we identified a subset of 4 diverse algorithms that perform well in different settings:

1. **TopPopular**: The popularity baseline of Cremonesi et al. [33].
2. **ItemKNN**: A simple nearest neighbors baseline based on item similarities [34].
3. **SVD**: A popular matrix factorization model.
4. **EASER**: A recently proposed linear autoencoder [35] that achieves SotA results on the Movielens-20M and Million Songs datasets.

For each of (2,3,4) we use 4 different hyperparameter sets (since comparisons between different hyperparameter settings are as important to researchers as comparisons between different methods), yielding a total of 13 recommender algorithms.

B.3. Metrics

We use the user-based offline ranking evaluation procedure of Liang et al. [25]:

1. Each recommender algorithm was trained on 80 percent of users (from either D_{train} or $\hat{D}_{\text{train}}$), with the remaining 20 percent used for validation.
2. For each of the validation users, we use the trained recommender to predict a held out 20 percent of items, with the other 80 percent as input.

We evaluate the quality of the predictions with three standard ranking metrics at different cutoffs:

1. Precision@1, @5, @10
2. Recall@1, @5, @10
3. Mean Reciprocal Rank (MRR) which does not have a cutoff and is more sensitive to rankings at lower positions.

This gives a total of 7 metrics that can be compared. Along with the 13 recommenders in Section 5.1 this gives a total of $1092 = 7 * P(13, 2)$ possible comparisons for computing Realism using a one-sided t-test.

B.4. More details

Each synthetic data generation algorithm from section 5.1 was used to generate synthetic datasets of 50000 users each, on which all 3 Identifiability metrics and the Realism metric were computed by sub-sampling separate sets of 4822 users for MovieLens and 5000 users for Million Songs. Identifiability was averaged over 3 trials with different synthetic user samples, while Realism was calculated once, but each recommender was trained 10 times with different synthetic user samples to perform the Student's t-test.

C. More Experimental Evaluations

We present additional experimental and qualitative analyses conducted with the tools we introduced in this paper.

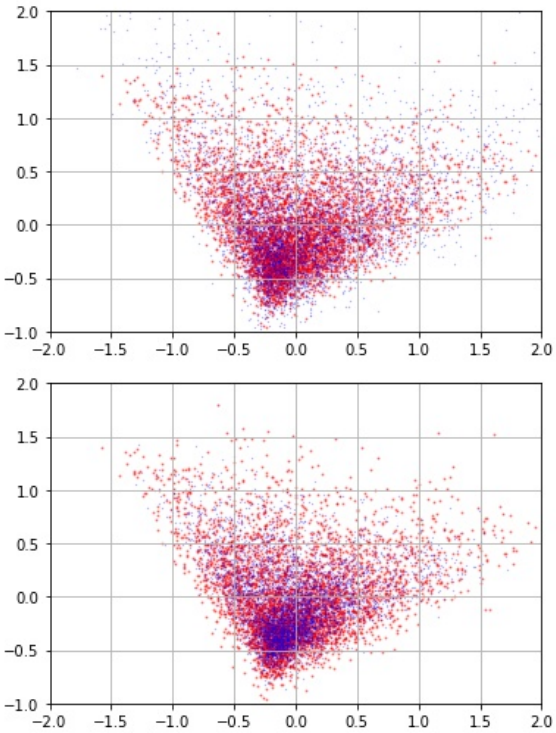


Figure 4: Top 2 Principal Components of synthetic (blue) and real user data (red). IdentityGAN (top) reproduces the real data distribution better than Clusters-50 (bottom) which is weighted too close to the mean.

C.1. Discussion of Pareto curves

The most significant conclusions we draw from Figures 3a-3f are regarding the composition of different sections of each Pareto curve. Each point along the curve represents a Pareto-optimal tradeoff between metrics that may be suitable for a particular application. Other points (in the interior) may be considered as being dominated by points on the curve, but studying their behavior can still give useful insights about the behavior of various algorithms and may be relevant to future research.

1. **N-Grams Methods**: For all the identifiability measures, these models occupy the regions of low identifiability and low realism for both domains.
2. **Clustering Methods**: These methods populate the interior of the Pareto curve for both Membership and Attribute Identifiability in both domains. They are generally outperformed by GAN-based models, but when the simplicity of implementation is considered,

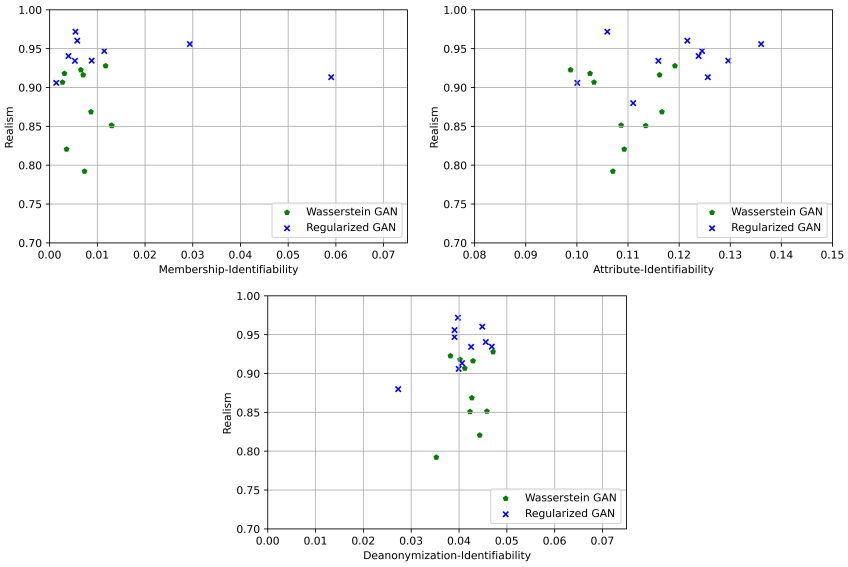


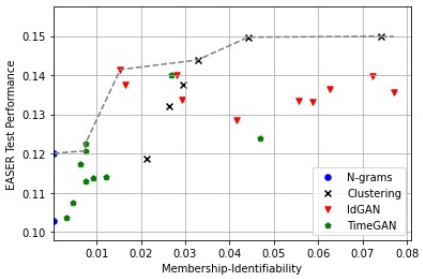
Figure 5: Plot of Wasserstein GAN (W-GAN) vs Regularized GAN (R-GAN) showing a clear clustering based on training method that spans hyperparameters and architectures.

may still be regarded as a viable approach. For De-anonymization Identifiability, clustering in fact defines a significant portion of the intermediate region of the curve. While not labeled on the graph for simplicity, increase in Realism and Identifiability is achieved by increasing the number of clusters (see Appendix for more details).

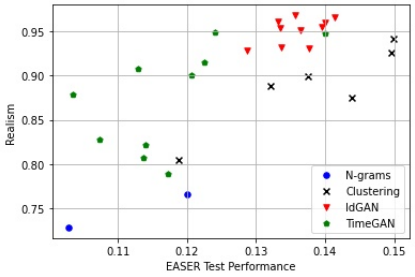
3. **GAN-based Models**: The TimeGAN and IdGAN models contribute to different sections of the Pareto curve for Realism vs Membership-Identifiability and Realism vs Attribute-Identifiability in both domains. They are less competitive compared to other methods (esp. clustering algorithms) on Realism vs Deanonymization-Identifiability, perhaps due to the hyper-parameters being optimized to minimize the distance between item distributions.
4. **Fractal Expansion and VAE**: These algorithms produced synthetic datasets with low realism and high identifiability (across all measures). In both cases, the situation is exacerbated by there being much fewer statistically significant results to be included in the realism measure (See Table 1 in the appendix). For fractal expansion, this is not surprising in hindsight, because the method was designed with a specific objective that is not recommender model independent, and without privacy considerations since it was meant to run on pre-sanitized public data. For VAE, the results are surprising given the competitiveness of the algorithm on recommendation problems in general [25], and may warrant further analysis of the latent distribution of users learned by the VAE.

C.2. Comparison with Absolute Performance

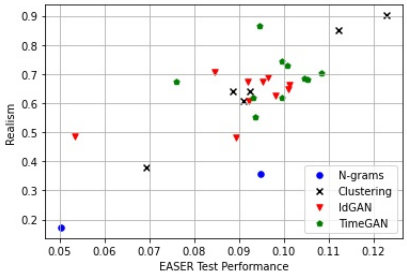
In section 3 we motivated the use of the Realism metric versus measures of absolute performance. It is natural to ask how interchangeable these metrics are. In Figures 6b and 6c, we show a comparison between Realism vs the performance of EASER on D_{test}. While there is some



(a) Membership-Identifiability vs Easer: MovieLens



(b) Realism vs EASER (Test): MovieLens



(c) Realism vs EASER (Test): Million-Songs

correlation, there is still a high degree of variance between them. To further illustrate the difference, we re-do the comparison of Realism vs Membership-Identifiability for MovieLens (i.e. Fig. 3a) with EASER Test performance instead (see Fig. 6a). This shows a clear quantitative and qualitative difference e.g. Clustering methods now dominate the top right region of the Pareto curve.

C.3. Discussion

The most significant conclusions we draw from Figures 3a-3f are regarding the composition of different sections of each Pareto curve. Each point along the curve represents a Pareto-optimal tradeoff between metrics that may be suitable for a particular application. Other points (in the interior) may be considered as being dominated by points on the curve, but studying their behavior can still give useful insights about the behavior of various algorithms and may be relevant to future research.

1. **N-Grams Methods**: For all the identifiability measures, these models occupy the regions of low identifiability and low realism for both domains.
2. **Clustering Methods**: These methods populate the interior of the Pareto curve for both Membership and Attribute Identifiability in both domains. They are generally outperformed by GAN-based models, but when the simplicity of implementation is considered, may still be regarded as a viable approach. For De-anonymization Identifiability, clustering in fact defines a significant portion of the intermediate region of the curve. While not labeled on the graph for simplicity, increase in Realism and Identifiability is achieved by increasing the number of clusters (see Appendix for more details).

3. **GAN-based Models**: The TimeGAN and IdGAN models contribute to different sections of the Pareto curve for Realism vs Membership-Identifiability and Realism vs Attribute-Identifiability in both domains. They are less competitive compared to other methods (esp. clustering algorithms) on Realism vs Deanonymization-Identifiability, perhaps due to the hyper-parameters being optimized to minimize the distance between item distributions.
4. **Fractal Expansion and VAE**: These algorithms produced synthetic datasets with low realism and high identifiability (across all measures). In both cases, the situation is exacerbated by there being much fewer statistically significant results to be included in the realism measure (See Table 1 in the appendix). For fractal expansion, this is not surprising in hindsight, because the method was designed with a specific objective that is not recommender model independent, and without privacy considerations since it was meant to run on pre-sanitized public data. For VAE, the results are surprising given the competitiveness of the algorithm on recommendation problems in general [25], and may warrant further analysis of the latent distribution of users learned by the VAE.

C.4. Qualitative Analysis

We performed a PCA analysis to visualize the qualitative difference in the synthetic user data generated by Clustering (on MovieLens with $K = 50$) vs a GAN model with high realism (Specifically, the highest realism IdentityGAN model). As Figure 4 shows, the GAN reproduces the distribution of the real data more accurately than Clustering, with more coverage of the tail distributions, while the clustering method over-represents the region near the mean user.

C.5. GAN Training Analysis

One of the hyperparameters used in the GAN training (see Appendix) was a binary variable selecting either a Wasserstein GAN (W-GAN) [36] or Regularized GAN (R-GAN) [16] training method. Inspection of the results (Figure 5) showed a consistent pattern w.r.t this variable that spanned other hyperparameter settings and even the 2 network architectures, IdentityGAN and TimeGAN. R-GAN consistently outperformed the W-GAN in terms of realism. However, on average the W-GAN was competitive with R-GAN on all the Identifiability metrics. We present this as an example of the kind of fine-grained analysis that our Realism-Identifiability framework affords.

C.6. Membership-Identifiability

Figure 7 shows how Membership-Identifiabilty changes as K is changed. For low K values the GAN outperforms the clustering methods but under performs for high k values.

D. Algorithm Implementations

Several of the algorithms tested either used or were modeled on the following implementations.
TimeGAN: https://github.com/jsyoon0823/TimeGAN
VAE: https://github.com/dawenl/vae_cf

Table 1
The number of recommender metric comparisons on synthetic users that were statistically significant and contributed to the Realism metric computations (averaged over hyper-parameters in each algorithm category).

Algorithms	MovieLens	MillionSongs
N-grams	764	669
Clustering	738	591
IdGAN	784	671
TimeGAN	749	641
Fractal Expansion	365	50
VAE	6	1

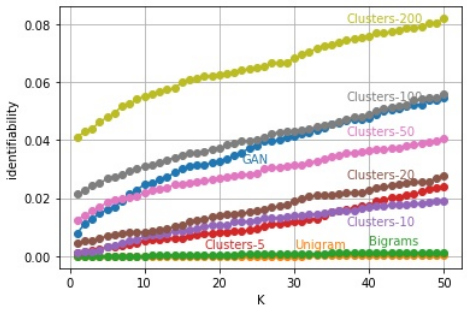


Figure 7: The change in Membership-Identifiablity as k is increased. The cluster based synthetic users grow very linearly while the GAN has a non-linear growth

Fractal Expansion: https://github.com/mlcommons/training/tree/master/data_generation/fractal_graph_expansions