Replication of Collaborative Filtering Generative Adversarial Networks on Recommender Systems

Discussion Paper

Fernando B. Pérez Maurera[1,2,*], Maurizio Ferrari Dacrema[1] and Paolo Cremonesi[1]

[1]Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy
[2]ContentWise, Via Simone Schiaffino 11, Milano, 20158, Milano, Italy

Abstract

CFGAN and its family of models (TagRec, MTPR, and CRGAN) learn to generate personalized and fake-but-realistic preferences for top-N recommendations by solely using previous interactions. The work discusses the impact of certain differences between the CFGAN framework and the model used in the original evaluation. The absence of random noise and the use of real user profiles as condition vectors leaves the generator prone to learn a degenerate solution in which the output vector is identical to the input vector, therefore, behaving essentially as a simple auto-encoder. This work further expands the experimental analysis comparing CFGAN against a selection of simple and well-known properly optimized baselines, observing that CFGAN is not consistently competitive against them despite its high computational cost. This work is an extended abstract of the paper presented in [1].

Keywords

Generative Adversarial Networks, Recommender Systems, Collaborative Filtering, Replicability

1. Introduction

Evaluation studies of previous works are fundamental to validate previous claimed progress. Several works have indicated the importance of such studies [2, 3, 4, 5, 6] for researchers and practitioners. This work presents an evaluation study of the most notable generative model applied in Recommender Systems: Collaborative Filtering GAN (CFGAN) [7].[1]

CFGAN [7] is a recommendation model based on Generative Adversarial Networks (GANs). It consists of two fully-connected feed-forward neural networks trained in an adversarial setting: a generator and a discriminator. Figure 1 illustrates the adversarial training of CFGAN. The *generator* learns to generate user profiles describing the preference of users toward items. The *discriminator* learns to distinguish between real user profiles and those created by the generator. Training of CFGAN converges when the generator creates *fake but realistic* user profiles. For a

IIR2022: 12th Italian Information Retrieval Workshop, June 29 - June 30th, 2022, Milan, Italy
*Corresponding author.
fernandobenjamin.perez@polimi.it (Fernando B. Pérez Maurera); maurizio.ferrari@polimi.it (M. Ferrari Dacrema); paolo.cremonesi@polimi.it (P. Cremonesi)
0000-0001-6578-7404 (Fernando B. Pérez Maurera); 0000-0001-7103-2788 (M. Ferrari Dacrema); 0000-0002-1253-8081 (P. Cremonesi)

[1]This work is an extended abstract of the work published in [1].

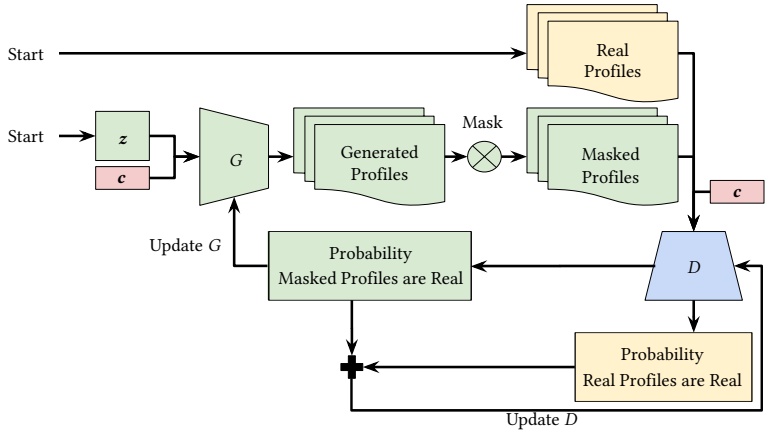

Figure 1: Training process of CFGAN. G, D, $\boldsymbol{z}$ and $\boldsymbol{c}$ are the generator network, discriminator network, random noise, and condition vectors, respectively. Real profiles are not masked.

given user, CFGAN constructs their recommendations by selecting the top-N items with the highest generated preference score.

This work presents and discusses the results of several experiments on CFGAN with the goal of addressing two research objectives. First, to describe inconsistencies found between the formulation of CFGAN and the implementation of it used in [7]. Second, to replicate the claimed progress made in [7] by measuring the CFGAN quality under a traditional top-N recommendation scenario against properly-tuned baselines. The discussions presented here are aligned with the exhortation given in [8]: research works should focus on understanding and analyzing the proposed models.

2. Inconsistencies in CFGAN

Figure 1 presents the architecture and training process of CFGAN as described in the reference of CFGAN [7]. From the figure, two vectors are part of the architecture of CFGAN: the random and condition vectors, denoted as $\boldsymbol{z}$ and $\boldsymbol{c}$, respectively. This work highlights inconsistencies between the implementation and the reference CFGAN presented in [7]: the use of user profiles as the condition vector and the absence of random noise for the empirical evaluation. These inconsistencies raise concerns about the model's ability to generalize and provide personalized recommendations.

First, the condition vector is used to provide *personalized recommendations*. To achieve this, this vector is encoded with users features, e.g., location, social information, identifiers, among others. Due to the collaborative nature of the datasets used in the experiments of the reference CFGAN [7], the *user profiles* are used as condition vectors, i.e., the data points that the generator and discriminator learn from. Using the user profiles as the condition vector makes both networks prone to learn trivial solutions. Essentially, the generator fundamentally becomes an auto-encoder and the discriminator may degenerate into learning a function that compares the condition with the real or generated profile.

Second, from a theoretical standpoint, the random noise is required on traditional GANs to

explore several points and to create a mapping between the random to the data spaces. The random noise vector is also part of the CFGAN reference and it serves the same purpose as for traditional GANs. However, the implementation of CFGAN in [7] removes the random noise from the model. Due to the absence of random noise, CFGAN is trained on highly sparse user profiles without the exploration of different input spaces. Furthermore, removing the random noise implicitly makes the assumption user profiles are static over time. As a consequence, CFGAN is less robust to evolving users preferences and *dataset shifts* [9].

3. Experimental Methodology

This work presents an evaluation study comprised of several experiments on CFGAN. The goal of this evaluation study is two-fold. First, to replicate the progress claims made in [7], where "replicability" is defined as in the ACM Artifact Review and Badging, version 1.1. [2] Second, to measure the effects in recommendation quality caused by the inconsistencies between CFGAN description and its implementation. The supplemental material provided in [7] solely contain the implementation of CFGAN and its data splitting, training, and evaluation. The details of the experimental methodology of the evaluation study is as follows:

Datasets and Splits: The experiments used the same open-source datasets and random holdout splits in [7], i.e., a sampled version of Ciao [10], and ML100K and ML1M versions of Movielens [11]. A validation split was created for hyper-parameter tuning purposes following the same split-creation steps as in [7].

Evaluation: All recommenders were evaluated on traditional accuracy and beyond-accuracy metrics [2] in the standard top-N recommendation scenario. Hyper-parameters were searched using bayesian search with 16 random cases, 50 total cases, and optimizing NDCG [2].

Baseline Recommenders: Neighborhood-based (ItemKNN and UserKNN) [2], graph-based (RP^3_β) [12], auto-encoders (SLIM ElasticNet [13] and EASE R [14]), and machine learning recommenders (PureSVD [15] and MF BPR [16]). The description of these recommenders, their hyper-parameters, and their ranges is found in [2].

CFGAN Recommenders: CFGAN as implemented in [7] was optimized. Two different variants were trained using the optimal hyper-parameters of the previous: CFGAN with random noise, and CFGAN using user identifiers as condition vectors.[3]

4. Results and Discussion

Table 1 shows accuracy and beyond accuracy metrics of baseline and CFGAN recommenders on the ML1M dataset.[4] Results on other datasets are consistent with this dataset except otherwise noted. From the table, it can be seen that at least two baselines have higher accuracy metric

[2] Available online at https://www.acm.org/publications/policies/artifact-review-and-badging-current.
[3] Due to space limitations, this work omits the list of hyper-parameters of CFGAN.
[4] Due to space limitations, only a subset of accuracy and beyond-accuracy metrics are shown.

Table 1
Accuracy and beyond-accuracy metrics for tuned baselines and CFGAN on the ML1M dataset at recommendation list length of 20. Higher accuracy values than CFGAN models reached by baselines in bold. ItemKNN and UserKNN use asymmetric cosine. CFGAN uses early-stopping. CFGAN UI is CFGAN with user identifiers as condition vector. CFGAN RN is CFGAN with random noise vector.

	PRECISION	RECALL	MRR	NDCG	COVERAGE ITEM
UserKNN	0.2891	**0.2570**	**0.6595**	**0.3888**	0.3286
ItemKNN	0.2600	0.2196	0.6254	0.3490	0.2097
RP3beta	0.2758	0.2385	**0.6425**	0.3700	0.3427
PureSVD	0.2913	0.2421	**0.6333**	0.0516	0.2439
SLIM ElasticNet	**0.3119**	**0.2695**	**0.6724**	**0.4123**	0.3153
MF BPR	0.2485	0.2103	0.5753	0.3242	0.3126
EASE R	**0.3171**	**0.2763**	**0.6795**	**0.4192**	0.3338
CFGAN	0.2955	0.2473	0.6222	0.3799	0.2167
CFGAN UI	0.1459	0.1118	0.3695	0.1831	0.0291
CFGAN RN	0.2915	0.2425	0.6211	0.3760	0.2021

than CFGAN. In particular, User KNN, SLIM ElasticNet, and EASE R have relative higher NDCG than CFGAN by $2.34\,\%$, $8.53\,\%$, and $10.34\,\%$, respectively. Furthermore, these more accurate baselines also trained faster than CFGAN, with differences in training time between two or three orders of magnitude. The results indicate that the progress claims made in [7] could not be replicated in the experiments of this evaluation study.

Regarding the absence of random noise, the results of the experiments are varied. Across datasets and variants, including random noise to CFGAN (CFGAN RN in Table 1) led to both relative increases or decreases in accuracy without a clear pattern.

Clear patterns resulted by changing the condition vector from user profiles to user identifiers (CFGAN UI in Table 1). Particularly, across datasets and variants, CFGAN UI consistently obtained relative lower accuracy metrics with respect to the base CFGAN. These results impose the following dichotomy. On one hand, using user profiles as condition vector may lead to both networks learn a trivial solution, as discussed in Section 2. On the other hand, using user identifiers as condition vectors when learning from pure collaborative data is possible on CFGAN at the cost of providing accurate recommendations.

Further studies are needed to address the recommendation quality of CFGAN and the inconsistencies presented in this work. For instance, a revision of the architecture of CFGAN can be addressed in future works. In this work, the results suggest that the current architecture does not work when changing the condition vectors to be the user identifiers. All these aspects are still open research questions and addressing will be beneficial for the maturity of this recommendation model.

References

[1] F. B. Pérez Maurera, M. Ferrari Dacrema, P. Cremonesi, An evaluation study of generative adversarial networks for collaborative filtering, in: Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part I, volume 13185 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 671–685. doi:10.1007/978-3-030-99736-6_45.

[2] M. Ferrari Dacrema, S. Boglio, P. Cremonesi, D. Jannach, A troubling analysis of reproducibility and progress in recommender systems research, ACM Trans. Inf. Syst. 39 (2021) 20:1–20:49. doi:10.1145/3434185.

[3] M. Ferrari Dacrema, P. Cremonesi, D. Jannach, Are we really making much progress? A worrying analysis of recent neural recommendation approaches, in: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019, ACM, 2019, pp. 101–109. doi:10.1145/3298689.3347058.

[4] J. Lin, The neural hype and comparisons against weak baselines, SIGIR Forum 52 (2019) 40–51. doi:10.1145/3308774.3308781.

[5] J. Lin, The neural hype, justified! a recantation, SIGIR Forum 53 (2021) 88–93. doi:10.1145/3458553.3458563.

[6] W. Yang, K. Lu, P. Yang, J. Lin, Critically examining the "neural hype": Weak baselines and the additivity of effectiveness gains from neural ranking models, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, ACM, 2019, pp. 1129–1132. doi:10.1145/3331184.3331340.

[7] D. Chae, J. Kang, S. Kim, J. Lee, CFGAN: A generic collaborative filtering framework based on generative adversarial networks, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018, ACM, 2018, pp. 137–146. doi:10.1145/3269206.3271743.

[8] Z. C. Lipton, J. Steinhardt, Troubling trends in machine learning scholarship, ACM Queue 17 (2019) 80. doi:10.1145/3317287.3328534.

[9] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, N. D. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009.

[10] J. Tang, H. Gao, H. Liu, mTrust: discerning multi-faceted trust in a connected world, in: Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012, ACM, 2012, pp. 93–102. doi:10.1145/2124295.2124309.

[11] F. M. Harper, J. A. Konstan, The MovieLens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (2016) 19:1–19:19. doi:10.1145/2827872.

[12] F. Christoffel, B. Paudel, C. Newell, A. Bernstein, Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015, ACM, 2015, pp. 163–170. doi:10.1145/2792838.2800180.

[13] X. Ning, G. Karypis, SLIM: sparse linear methods for top-n recommender systems, in: 11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011, IEEE Computer Society, 2011, pp. 497–506. doi:10.1109/ICDM.

2011.134.

[14] H. Steck, Embarrassingly shallow autoencoders for sparse data, in: The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, ACM, 2019, pp. 3251–3257. doi:10.1145/3308558.3313710.

[15] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, ACM, 2010, pp. 39–46. doi:10.1145/1864708.1864721.

[16] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: bayesian personalized ranking from implicit feedback, in: UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009, AUAI Press, 2009, pp. 452–461.