Info-flow Enhanced GANs for Recommender

Yuan Lin
zhlin@dlut.edu.cn
WISE Lab., Faculty of Humanities and Social Sciences, Dalian University of Technology
Dalian, Liaoning, China

Zhang Xie
xzreal_dlut@163.com
WISE Lab., Faculty of Humanities and Social Sciences,Dalian University of Technology
Dalian, Liaoning, China

Bo Xu*
xubo@dlut.edu.cn
Information Retrieval Lab., School of Computer Science and Technology, Dalian University of Technology
Dalian, Liaoning, China

Kan Xu*
xukan@dlut.edu.cn
Information Retrieval Lab., School of Computer Science and Technology, Dalian University of Technology
Dalian, Liaoning, China

Hongfei Lin*
hflin@dlut.edu.cn
Information Retrieval Lab., School of Computer Science and Technology, Dalian University of Technology
Dalian, Liaoning, China

ABSTRACT

Recommendation systems can help users process large amounts of information, and generative adversarial networks (GANs) show great potential in recommendation systems. In this paper, we propose a new GAN model to enhance the information flow within the generator based on the information flow between the original generator and discriminator. Our experimental results indicate that our model reduces the discrepancy between the generator and the discriminator. Both the generator and discriminator yield considerable performance improvements compared to other strong baselines. The improvements by NDCG@3 and MRR are significant, which can reach 30.98% and 30.17%, respectively.

CCS CONCEPTS

• **Information systems → Collaborative filtering**.

KEYWORDS

Generative adversarial networks, Recommendation system, Information flow

ACM Reference Format:
Yuan Lin, Zhang Xie, Bo Xu, Kan Xu, and Hongfei Lin. 2021. Info-flow Enhanced GANs for Recommender. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463009

*Corresponding Author

SIGIR '21, July 11–15, 2021, Virtual Event, Canada.

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463009

1 INTRODUCTION

A typical recommendation system contains two roles, namely, users and items. A recommendation system calculates the user's preferences for items and provides preferred items for users. Some recommendation models have achieved good results. For example, the BPR model [17] is used to form BPR-Opt, which is derived from the maximum posterior estimator for optimal personalized rankings. NCF [13] combines matrix factorization and a multilayer perceptron, which extracts features from both low dimensions and high dimensions, thus achieving good recommendation results.

The framework of a GAN model can be divided into two parts: the generator and the discriminator. The generator constantly fits the real data distribution and generates fake data to deceive the discriminator. The discriminator learns to distinguish fake data from real data through the generator's repeated adversarial training [1, 3, 9–11, 16]. Considering recent development of recommenders and GANs, researchers have applied GANs to recommendation systems. Due to the particularity of a recommendation system, the generator does not generate new things as usual, but rather it processes an information flow in the form of generating hard negative samples. The generator does not generate new things, but rather it samples existing items for the discriminator [2, 15, 18, 22, 24]. Recently, the popularity of applying GANs to recommender systems has improved. The PDGAN [22] was proposed to better capture users' personal preferences for both individual items and the diversity of a set of items. The IRGAN [20] is applied to multiple semisupervised IR tasks and can properly handle implicit feedback. IRGAN for recommender implements classical collaborative filtering as one of the uncluttered scoring functions for user preferences. However, due to the nonequivalence of the training results between the generator and discriminator, the degree of improvement obtained by the generator is much lower than that of the discriminator. To more effective train the model, our research aims to avoid the initial random sampling process in the generator. We propose the concept of the info-flow as all information interactions that exist in the following: (1) the interior information flow of the generator and (2) the information flow between the generator and discriminator. These two information flows are hereinafter referred to as the inner info-flow and the outer info-flow. We add the inner info-flow to remedy the flaws caused by the existence of only the outer info-flow. The process of the inner info-flow can be analogized to the process of the blood flow among organs in the human body below.

The general process of the external information flow is that the generator generates hard pairs for the discriminator to train, and the discriminator takes the reward loss to the generator. In this case, due to the discrepancy in the performances of the generator and the discriminator, a poor sample may occur [21]. Similar to the abovementioned process, in the human body, the heart receives blood from the veins and delivers blood to other organs in the body through muscle contractions. If the heart's blood supply capacity is not balanced with the blood required by the organs in the body (e.g., insufficient blood supply caused by vigorous exercise), which is similar to the problem of the discrepancy between the generator and the discriminator, then the heart is no longer able to take on this heavy work, like the performance flaw that occurs for the generator. As a result, various problems will emerge. What the blood flow process among organs tells us is that if we install a pacemaker inside the heart to help the heart supply sufficient blood through the stimulation of electrical pulses, this symptom can be greatly relieved, and thus the human body functions normally. Hence, similarly, we make improvements inside the generator to enhance the info-flow within the generator (inner info-flow), thereby stimulating the generator to sample more difficult examples. Consequently, both the generator and discriminator will be better trained.

In order to make the generator better capture the connections between users and items, we pretrained the network on a dataset and constructed a 54-dimensional feature vector that represents the connections between one user and all items to focus on the connections between users and items. In other words, each dimension of the 54-dimensional feature vector can be regarded as u's scoring of i. Then, the training after inputting the 54-dimensional feature vector into the adversarial network is similar to a learning-to-rank system, and the loss function is also consistent with the learning-to-rank paradigm. These features are input to the generator, which greatly improves the generator. This process can be understood as extending the structure of the GAN to enhance the inner info-flow process within the generator.

2 THE PROPOSED MODEL

2.1 Model overview

We divide the generator into two parts, namely, the level one generator and level two generator. Specifically, we add LambdaRank [4, 5] to the level one generator as the input of the GAN to determine the sampling possibility of examples, thus determining the waiting list of examples in the inner info-flow. We obtain this intermediate waiting list before the sampling process to sample examples directly. The superiority of this method is that by selecting examples scored by LambdaRank, the probability that difficult examples are discriminated will be improved by the discriminator.

Level one generator: $g_{\lambda prob}(j|u,r)$ tries to fit the underlying relevance distribution and generates the item given a relevant score and a user. In other words, level one generator not only aims to improve its own performance, but it also continuously selects hard negative samples for the level two Generator.

Level two generator: $g_{\theta prob}(i|u,r)$ not only models $p_{true}(i|u_n,r)$ to generate samples for the discriminator, but also learns a discriminative score for each user-item pair to give direct feedback to itself. For the level one generator, the level two generator acts as both a student and a mate: as a student, it receives batches of hard samples to train; as a mate, it constantly gives feedback to the level one generator in the form of gradients. As a result, both generators more efficiently explore the gradient space in collaboration. Meanwhile, for the discriminator, the level two generator acts as both a competitor and partner: as a competitor, it constantly generates hard pairs for the discriminator to train; as a partner, it obtains the response from the competitor and guides its training.

In summary, the generator is trained to fit the scoring functions $g_{\lambda score}(i,u)$ and $g_{\theta score}(i,u)$ and generate the generative probabilities $g_{\lambda score}(i,u)$ and $g_{\theta score}(i,u)$, respectively. The generator not only aims to improve its own performance, but it also continuously selects hard negative samples for itself. In this setting, generator G acts as an information hub, allowing generative information to be passed forth to the inner info-flow and discriminative information to be passed back to the outer info-flow. The discriminator is trained to fit the score function $d_{\Phi score}(i,u)$, and constantly discriminate the hard samples from the generator to improve its performance.

2.2 Overall objective

We apply adversarial losses for the generator-discriminator info-flow (outer info-flow). We improve the performance of discriminator D by maximizing the cross-entropy loss and train the true samples to 1 and the fake (negative) samples to 0. The negative samples are sampled by the generator, which estimates the generative probability $p_\theta(i|u_n,r)$. Our objective is as follows:

$$\begin{aligned} J^{G^*,D^*} = \min_{\theta} \max_{\phi} \sum_{n=1}^{N} (E_{i\sim p_{true}(i|u_n,r)})\,[logD(i|u_n)] \\ + E_{i\sim p_\theta(i|u_n,r)})\,[log(1-D(i|u_n))] \end{aligned} \tag{1}$$

For inner Info-flow, the generator estimates the generative probability $p_\lambda(i|u_n,r)$ so that it can sample negative samples for itself. We improve the performance of the generator G by maximizing the cross-entropy loss, train the true samples to 1 and the fake(negative) samples to 0. We design our objective as follow:

$$\begin{aligned} J^{G^*} = \min_{\lambda} \max_{\theta} \sum_{n=1}^{N} (E_{i\sim p_{true}(i|u_n,r)})\,[logG(i|u_n)] \\ + E_{i\sim p_\lambda(i|u_n,r)})\,[log(1-G(i|u_n))] \end{aligned} \tag{2}$$

2.3 Training discriminator

The discriminator is trained to fit the scoring function $D_{\Phi score}(i,u)$. The objective of the discriminator is to maximize the sampled fake positive items and minimize the log likelihood prediction.

$$\begin{aligned} \phi^* = \max_{\phi} \sum_{n=1}^{N} (E_{i\sim p_{true}(i|u_n,r)})\,[logD(i|u_n)] \\ + E_{i\sim P_\theta(i|u_n,r)})\,[log(1-D(i|u_n))] \end{aligned} \tag{3}$$

where $p_{true}(i|u_n,r)$ is the underlying relevance distribution, which shows the true probability. $p_\theta(i|u_n,r)$ is the estimated generation probability that is produced by the generator. Through the interaction of these parameters, ϕ can be made more reasonable and better.

We use the RankNet [6] model as the discriminator and pair loss as its loss function. The input of the function is a pair of items [5, 7]:

$$Pairloss = logD(i|u_n) = \sum_{n=1}^{K} log\sigma(f_{\phi}(pos_n) - f_{\sigma}(neg_n)) \tag{4}$$

The negative items are sampled from $p_\theta(i|u_n, r)$ by the level two generator, and the level one generator learns from $p_\lambda(i|u_n, r)$. $p_\lambda(i|u_n, r)$ is learned by the level one generator from the raw data.

2.4 Training Generator

Level one Generator We use the LambdaRank model as level one Generator to help the level two Generator. By minimizing lambda loss and generating NDCG probability $g_{\lambda prob}(i|u, r)$, the level one Generator is trained to fit the score function $g_{\lambda score}(i, u)$.

For the discontinuity of the sampling, we further deduce the formula and approximate the gradient. For simplicity, we express $E_{i \sim P_\lambda(i|u_n,r)})[log(1 - D(i|u_n))]$ as $J^{G_\lambda}(u_n)$. Due to the fact that our neural network is a nonconvex function, we use the gradient descent method to obtain the maximum point. Although it is difficult to calculate the gradient of $J^{G_\lambda}(u_n)$, the strategy gradient is applied to obtain an approximate gradient:

$$\begin{aligned} \lambda^* &= \underset{\lambda}{argmax} \sum_{n=1}^{N} (E_{i \sim p_\lambda(i|u_n,r)})[log(1 - G_\lambda(i|u_n)] \\ &= \underset{\lambda}{argmax} \sum_{n=1}^{N} J^{G_\lambda}(u_n) \end{aligned} \tag{5}$$

$$\begin{aligned} \Delta_\lambda J^{G_\lambda}(u_n) &= \Delta_\lambda E_{i \sim p_\lambda(i|u_n,r)})[log(1 - G_\lambda(i|u_n)] \\ &= E_{i \sim p_\lambda(i|u_n,r)})[\Delta_\lambda log p_\lambda(i|u_n, r) log(1 - G_\theta(i|u_n)] \\ &\approx \frac{1}{K} \sum_{i=1}^{K} \Delta_\lambda [\underbrace{log p_\lambda(i|u_n, r)}_{Listloss} \underbrace{log(1 - G_\theta(i|u_n))}_{Reward}] \end{aligned} \tag{6}$$

The core idea of the equation is to transform the gradient of expectation to the expectation of gradient. In the above formula, K represents the item number of a specific user.

$$\lambda = N\left(\frac{1}{1 + e^{s_i - s_j}}\right)(2^i - 2^j)\left(\frac{1}{ln(1+i)} - \frac{1}{ln(1+j)}\right) \tag{7}$$

where N is the reciprocal max DCG for the user. All λ functions were designed with the $NDCG$ cost function.

Level two Generator. Level two generator uses the fitting scoring function $g_\theta score(i, u)$ and generates the probability $g_\theta prob(i|u, r)$. It receives a batch of hard samples from the level one generator, and provides false samples for the discriminator to cheat the discriminator. Having noticed that the conventional generator often times provides unsatisfactory results, we add a level one generator inside the generator to make it perform better.

$$\begin{aligned} \theta^* &= \underset{\theta}{argmax} \sum_{n=1}^{N} (E_{i \sim p_\theta(i|u_n,r)})[log(1 - D(i|u_n)]) \\ &\approx \underset{\theta}{argmax} \sum_{n=1}^{N} \frac{1}{K} \sum_{n=1}^{K} [\Delta_\theta \underbrace{log p_{\theta(i|u,r)}}_{Pointloss} \underbrace{log(1 - D(i|u_n))}_{Reward}] \end{aligned} \tag{8}$$

where $p_{true}(i|u_n, r)$ is the underlying relevance distribution, namely, the true probability. $p_\lambda(i|u_n, r)$ is the estimated generation probability by level one Generator. Through the info-flow of generators,

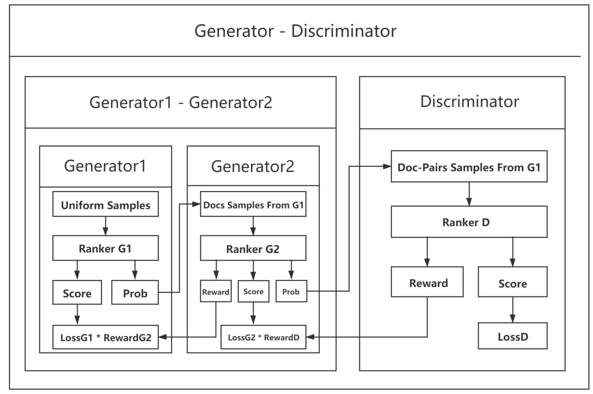

Figure 1: The workflow of the IFGAN.

Algorithm: IFGAN

Require: The level one generator $p_\lambda(i|u_n, r)$ and $g_{\lambda score}(i, u)$;
The level two generator $p_\theta(i|u_n, r)$ and $g_{\theta score}(i, u)$;
Discriminator $d_{\phi score}(i, u)$;
Dataset $S = \{Movielens100K\}$;
Initialize g1, g2 and d with random weights λ, θ and ϕ.

repeat: The generator cycles
Train the level one generator via the policy gradient ***(ListLoss∗ Reward)***
$p_\lambda(i|u_n, r)$generates hard items for the level two generator
Train the level two generator via policy gradient using examples from $p_\lambda(i|u_n, r)$
repeat: The generator-discriminator cycles
Train the level two generator via the policy gradient ***(PointLoss∗ Reward)***using examples from $p_\lambda(i|u_n, r)$
$p_\theta(i|u_n, r)$ generates hard item pairs for the discriminator
Train the discriminator $d_{\phi score}(i, u)$using the ***pair loss***
Until IFGAN converges

we achieve the optimal parameter θ, thus improving the effect of Generator. Here is the definition of the G function.

$$Pointloss = \sum_{n=1}^{K} log\sigma(f_\theta(item_n)) \tag{9}$$

By modifying the traditional generator, the conventional generator is divided into the level one generator and the level two generator.

2.5 Workflow

Figure 1 shows the total workflow of our model. The level one generator teaches the hard items to the level two generator, and the level-two generator gives a feedback reward to the level one generator, which is directly multiplied by the list loss.The level two generator generates hard pairs to the discriminator, and the discriminator gives a discriminative reward to the level two generator, which is directly multiplied by the point loss.In order to better show the algorithm flow, The following algorithm shows the overall training process.

3 EXPERIMENTS

3.1 Experimental Setting

Dataset We conducted experiments with the MovieLens (100k) dataset. The MovieLens dataset consists of user ratings for movies (1 to 5). It contains 943 users, 1683 items, and 18 explicit categories;

Table 1: Experimental results on the Movielens dataset.

Model	R@5	R@10	MAP	N@3	N@5	N@10	MRR
BPR	0.1181	0.1936	0.2376	0.4078	0.3889	0.3632	0.5952
CDAE	0.0708	0.1128	0.1374	0.2169	0.2115	0.2047	0.3923
NCF	0.1284	0.2066	0.2527	0.4458	0.4183	0.3912	0.6229
CML	0.1435	0.2324	0.2813	0.4797	0.4508	0.4198	0.6612
IRGAN	-	-	0.2418	0.4222	0.4009	0.3723	0.6082
LRML	0.0906	0.1473	0.1724	0.3040	0.2820	0.2639	0.4967
CF-GAN	0.152	-	-	0.4760	-	-	
LightGCN	0.1569	0.2305	0.2833	0.4582	0.4329	0.3701	0.6403
IFGAN	0.1658	0.2407	0.2895	0.6283	0.4928	0.4241	0.8607
Improve	5.67%	3.57%	2.19%	30.98%	9.32%	1.02%	30.17%

each movie belongs to multiple categories. We consider the 5 ratings of the MovieLens dataset as positive feedback.

Implementation The input data are pretrained into an equal dimension feature vector, and fed into a two-layer neural network. The first linear layer is activated by a tanh function. The second layer is activated by the sigmoid function. The generative probability $prob(i|u,r)$ on modeling the data distribution is calculated by the softmax function. We set the optimizer of the generator as the stochastic gradient descent with learning rate 1e-6, set the optimizer of the generator and discriminator as Adam with learning rate 1e-5, betas of (0.9, 0.999) and no weight decay.

Baselines and Metrics We compare our model with the following baselines: BPR [17], CDAE [23], NCF [13], CML [14], IRGAN [20], LRML [19], CFGAN [8], LightGCN [12]. These baselines explore the information flow between user and item more or less.

We use multiple measures to evaluate the methods: Ranker Precision metrics (R@5 R@10, MAP, MRR), Normalized Discounted Cumulative Gain (NDCG@3, NDCG@5, NDCG@10, short for N@k), which are widely used metrics in evaluation.

3.2 Experimental Result

We report the experimental results in Table 2. Our model outperforms the baseline by approximately 5.69% and 13.77% by the average performance of precision and the NDCG, respectively. Moreover, considering that the quality of the top-ranked items is important in real-world scenarios, the maximum improvement of NDCG@3 is significant, and it reaches up to 30.98%. Other metrics are also improved.

3.3 Ablation experiment

In order to further verify the effectiveness of the info-flow proposed in this article, we conducted two ablation experiments. The first ablation experiment only uses LambdaRank, and the second ablation experiment uses an adversarial network without the inner info-flow (IFGAN-outer), which means that generator 2 is removed.

Table 2 compares the best results of the NDCG indicators of the two ablation experimental models and the original model to verify the effectiveness of the original IFGAN model. Figure 2 compares the performance of the discriminator in the training process of ablation experimental model 2 and the original model. Obviously, generator 1 can effectively help the negative sampling process of generator 2 because the performance gap in the discriminators of the two models is relatively obvious. The results of the ablation

Table 2: Results of ablation experiments.

Model	NDCG@3	NDCG@5	NDCG@10
LambdaRank	0.2241	0.2028	0.1935
IFGAN-outer	0.4512	0.4260	0.3774
IFGAN	0.6283	0.4928	0.4241
Improvement	39.25%	15.68%	12.37%

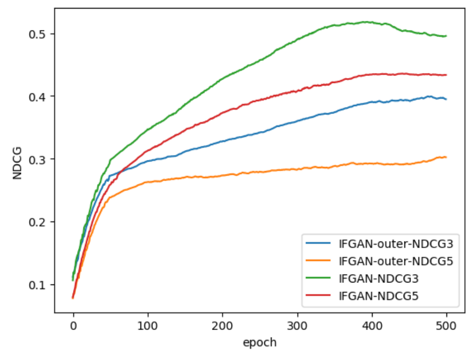

Figure 2: Algorithm: IFGAN

experiment further verify the effect of inner info-flow on the overall performance. In addition, in the observation of the discriminator, we also found a similar conclusion. In fact, the two info-flows can greatly improve the performance of the generator2. During the interactive training process between the generator2 and the discriminator, the discriminator receives the sample pairs from the second generator, and the reward of its own feedback is more accurate, which further promotes the overall training process.

4 CONCLUSION

In this paper, we proposed a novel GAN framework to enhance the adversarial network by adding two types of info-flow processes. The key feature of our model is that its two-level generator networks work for two types of information flows and play discriminative and generative roles. Our proposed model takes advantage of the insightful inner info-flow, LambdaRank, which gives more reasonable sampling. We use a pairwise model, RankNet, which plays an important role in the outer info-flow. We conducted experiments on the MovieLens dataset and validated the effectiveness of our model. Our work encourages additional research on recommendation systems with GANs. We will also exploit ways to apply our method to other research fields, such as term ranking in query expansion.

5 ACKNOWLEDGEMENTS

The Natural Science Foundation of China (No.61976036, 62006034), the Ministry of Education Humanities and Social Science Project (No.19YJCZH199), and the Fundamental Research Funds for the Central Universities (No.DUT21RC(3)015). The authors would like to thank Chengxuan Ying, Yanhao Li, Fangzhou Mou, Zixiong Ye and Haifeng Liu for the discussion, which helped determine the direction of this work.

REFERENCES

[1] Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 214–223. http://proceedings.mlr.press/v70/arjovsky17a.html

[2] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. 2017. CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2764–2773. https://doi.org/10.1109/ICCV.2017.299

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=B1xsqj09Fm

[4] Christopher J. C. Burges. 2010. From Ranknet to Lambdarank to Lambdamart: An Overview. *Learning*. 11 (2010), 81.

[5] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. 2006. Learning to Rank with Nonsmooth Cost Functions. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, Bernhard Schölkopf, John C. Platt, and Thomas Hofmann (Eds.). MIT Press, 193–200. https://proceedings.neurips.cc/paper/2006/hash/af44c4c56f385c43f2529f9b1b018f6a-Abstract.html

[6] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. Learning to rank using gradient descent. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005 (ACM International Conference Proceeding Series, Vol. 119)*, Luc De Raedt and Stefan Wrobel (Eds.). ACM, 89–96. https://doi.org/10.1145/1102351.1102363

[7] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007 (ACM International Conference Proceeding Series, Vol. 227)*, Zoubin Ghahramani (Ed.). ACM, 129–136. https://doi.org/10.1145/1273496.1273513

[8] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. 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*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 137–146. https://doi.org/10.1145/3269206.3271743

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2672–2680. https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

[10] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5767–5777. https://proceedings.neurips.cc/paper/2017/hash/892c3b1c6dccd52936e27cbd0ff683d6-Abstract.html

[11] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long Text Generation via Adversarial Training with Leaked Information. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 5141–5148. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16360

[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 639–648. https://doi.org/10.1145/3397271.3401063

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 173–182. https://doi.org/10.1145/3038912.3052569

[14] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge J. Belongie, and Deborah Estrin. 2017. Collaborative Metric Learning. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 193–201. https://doi.org/10.1145/3038912.3052639

[15] Salman H. Khan, Munawar Hayat, and Nick Barnes. 2018. Adversarial Training of Variational Auto-Encoders for High Fidelity Image Generation. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. IEEE Computer Society, 1312–1320. https://doi.org/10.1109/WACV.2018.00148

[16] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.06434

[17] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *CoRR* abs/1205.2618 (2012). arXiv:1205.2618 http://arxiv.org/abs/1205.2618

[18] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 3483–3491. https://proceedings.neurips.cc/paper/2015/hash/8d55a249e6baa5c06772297520da2051-Abstract.html

[19] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 729–739. https://doi.org/10.1145/3178876.3186154

[20] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 515–524. https://doi.org/10.1145/3077136.3080786

[21] Qiang Wu, Christopher J. C. Burges, Krysta M. Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Inf. Retr.* 13, 3 (2010), 254–270. https://doi.org/10.1007/s10791-009-9112-1

[22] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. 2019. PD-GAN: Adversarial Learning for Personalized Diversity-Promoting Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 3870–3876. https://doi.org/10.24963/ijcai.2019/537

[23] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, Paul N. Bennett, Vanja Josifovski, Jennifer Neville, and Filip Radlinski (Eds.). ACM, 153–162. https://doi.org/10.1145/2835776.2835837

[24] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2242–2251. https://doi.org/10.1109/ICCV.2017.244