List-IRGAN: A List-wise Learning to Rank Approach of IRGAN for Top-n Recommendation

Junjie Liang, Qian Sun, Chaoran Chen
College of Information Science and Technology,
Pennsylvania State University, IST Building,
University Park, PA 16802, USA

Abstract

The state-of-the-art GAN model in information retrieval, IRGAN, provides a unified account of two schools of thinking in information retrieval modeling: the generative model focusing on selecting relevant negative items from the item pool to mimic the true user preference to fool the discriminative model; the discriminative model focusing on distinguishing the negative items to best the generative model. However, the binary nature of the discriminative model might be limited due to the lack of utilization of more informative feedback, e.g., continuous ratings. In this project, we extend IRGAN to a list-wise learning-to-rank framework which enables the discriminative model to directly learn the top-n ranking order of the items using the continuous rating data. Empirical analysis demonstrates that our proposed method, List-IRGAN, is on part with the IRGAN on generator and performs better than it on discriminator.

Introduction

Nowadays, its common to gain access to the user feedback of the online e-commerce systems, which greatly motivate the implementation and refinement of modern recommender system. Advanced recommender system could increase the number of products sold. This is probably the most important function for a commercial recommender system. For example, a proper recommendation result could persuade users to buy an additional set of items (combination) instead of those usually sold without any kind of recommendation. This additional revenue could bring possible reward to all parts of the advertising business. Recommender system could also increase the user satisfaction and user fidelity. A properly designed human-computer interaction will help users to find relevant products and services. With the popularity of neural network models, increasing number of scholars are using deep learning models to help enhance the prediction accuracy of the system [1].

Generative Adversarial Nets (GAN [2]) are widely used in the fields of image generation, text generation, 3D object reconstruction, etc. There are two school of models in GAN: the discriminative model and the generative model. The goal of the discriminative model is to determine whether an example is drawn from the true data pool, while the goal of the generative model is to generate fake examples that mimic the ground-truth distribution of the data pool so to fool the discriminative model. Recently, GAN model is adapted to work on the field of Information Retrieval (IRGAN [3]). Contrary to the original GAN, IRGAN does not generate fake examples; rather, it selects negative examples from the true data pool that is similar to the positive ones so to escape the detection of the discriminator. Such practice transfers the original continuous generating problem to a discrete selecting problem, which is crucial to the application of recommender system. Extensive experimental results showcase that IRGAN outperforms several state-of-the-art models with respect to the area of Web Search, Item Recommendation and Question Answering.

However, IRGAN is designed to work on the binary classification data, the lack of usage of the ordinal or continuous data will limit the strength of the model. As in real-world item recommendation scenario, there are various types of user feedback available; even for a type of user feedback, e.g., user-item ratings, the value is continuous, and the exact value will provide extra knowledge for the system. On the other hand, end-users are usually presented with the top-n results of the computed item list, which means the sorting results of the item list might be more useful compared to the simplistic binary classification. Hence it is interesting to blend the Learning-to-Rank approach into IRGAN model to learn the specific ranking order of the item list. In this project, we wish to adapt the IRGAN model to the framework of list-wise learning-to-rank (List-IRGAN) and enhance the ability of IRGAN to learn non-binary data. Particularly, our proposed model, List-IRGAN, aims to maximize one of the most popular top-n ranking measure, DCG@N, which is a list-wise ranking approach that directly learn the ranking order of the item list. Through optimizing DCG@N, our method suits the specific application where only the top-n items are presented to the end-users. In this project, we will utilize the user feedback to build a recommender system based on the latent factor models, which is one of the most popular and effective models in the framework of Collaborative Filtering [4].

We perform the empirical analysis on the MovieLens dataset [5], which contains the user ratings about the movies

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

they have watched. In our experiment, we compare our model with IRGAN and the traditional binary classification model. Experiments show that the discriminator of our model is significantly better than that of IRGAN; whereas the performance of generator is almost identical.

Methodology

In this section, we first present the preliminaries of our model, and then present the formulations of List-IRGAN. Before we start, the basic notations are as follow:

Let $\mu = \{u_1, u_2, \ldots, u_n\}$ be the set of users, $\iota = \{i_1, i_2, \ldots, i_n\}$ be the set of items. The representation of observed data can be denoted as $R = \mu \times \iota$, where the entry r_{ij} represents that user μ rates item ι with score . We further denote the partially observed item set of μ as ι_u^+. For $i \in \iota_u^+$, we denote the rating of $i$as f_{ui} and the ranked position of i in us observed item set as R_u^+.

1. Preliminaries

1.1. Latent Factor Model Latent factor models are considered to be state-of-the-art in the CF framework since they often achieve attractive accuracy as well as scalability in practice [4]. The basic idea of latent factor models is to determine the fully-specified low-dimensional approximation of the original matrix to perform both reduced representation and effective prediction. Let θ be the set of latent factors such that θ^{user} is an $k \times n$ matrix with the u-th column $\theta_u^{user} \in R^k$denotes the latent factors of u and θ^{item} is an km matrix with the i-th column $\theta_i^{item} \in R^k$denotes the latent factors of i. The parameter k is known as the dimension of latent factor matrices which is much smaller than n and m. The rating for u to i is predicted by:

$$f_{ui} = b_i + \theta_u^{user^T} \theta_i^{item} = b_i + \sum_{t=1}^{k} \theta_{ut}^{user} \theta_{it}^{item} \quad (1)$$

where b_i is the bias term for item i.

1.2. Discounted Cumulative Gain The discounted Cumulative Gain (DCG) is widely used to evaluate the accuracy of the recommendation list, where the gain of the higher ranked items outweighs that of the lower ranked ones. Since we focus on the top-N recommendation scenario, its essential to emphasize higher influence on the higher positions, which makes DCG a proper objective function to quantify the ranking quality of a recommendation list. Let y_{ui} be a binary indicator to represent whether item i is relevant to a user u, then DCG of u is computed by:

$$DCG_u = \sum_{i=1}^{m} \frac{2^{y_{ui}-1}}{log(R_{ui}+2)} \quad (2)$$

Notice that the ranked position (start from zero) of item i can also be computed by comparing $f_u i$ to $\{f_{uj} | j \in m\}$, which is denoted as:

$$R_{ui} = \sum_{j=1}^{m} I(f_{ui} < f_{uj}) \quad (3)$$

where I(P) is an indicative function with I(P)=1 if P is true and otherwise I(P)=0.

1.3. IRGAN IRGAN is a unified framework for fusing generative and discriminative IR model in an adversarial setting. For a given user u, we have a set of preferred items. The underlying preferred item distribution can be expressed as conditional probability $p_{true}(i|u)$, which depicts the users preference distribution over the candidate items w.r.t. his observed item list. Given the observed item list as training data, we can construct two types of models:

Generative model. $p_\theta(i|u)$, which tries to select preferred items from the candidate pool for the given user u. Its goal is to approximate the true preference distribution $p_{true}(i|u)$ as much as possible.

Discriminative model. $f_\phi(u, i)$ which, as oppose to the generative model, tries to discriminative well-matched user-item pair (u, i) from ill-matched ones, where the goodness of matching depends on the ground truth. In other words, its simply a binary classifier and try to best the selected items given by the generative model.

Thus, inspired by the idea of GAN, the way to unify these two different types of IR models is to let them play a minimax game: the generative model tries to select relevant items that look like the ground truth and therefore could fool the discriminative model, whereas the discriminative model would try to distinguish between the ground-truth preference items and the selected ones made by its opponent generative model. Formally, the overall objective can be written as:

$$\begin{aligned} J^{G^*,D^*} = \min_\theta \max_\phi \sum_u \big(& E_{i \sim p_{true}(i|u)}[\log D(i|u)] \\ & + E_(i \sim p_\theta(i|u))[\log(1 - D(i|u))] \big) \end{aligned} \quad (4)$$

where the generative model G is written as $p_\theta(i|u)$, and the discriminative model D estimates the probability of item i being preferred by user u.

2. List-IRGAN Formulation

IRGAN can only be used in a binary classification setting in which only the preference label of user-item pair is considered. However, in real-life scenario, the user-item feedback could be ordinal or continuous. For example, in Netflix, users could give the movie a rating ranging from 1-5 with 1 representing dislike and 5 representing like [6]. With the actual rating scores, we could sort the rated items in the descending order and the corresponding order of the items can provide more abundant knowledge about the user preference. There are existing models using learning to rank methods to directly predict the ranking results of the given items [7], the experimental analysis of which usually showcase considerable improvement over the models that directly predict the item ratings. In this project, we try to fuse the learning to rank method into the IRGAN model so that the generative model and discriminative model are enhanced as follow:

Generative model. $p_\theta(i|u)$, which tries to select preferred items from the candidate pool for the given user u. Its goal is to challenge the discriminative model and select the not yet well-ordered item subset of the discriminative model. **Discriminative model.** $f_\phi(u, i)$ which, contrary to

the generative model, tries to learn the perfect top-ranking items from the ill-ordered items selected by the generative model. Unlike the discriminative model in IRGAN that train a binary classifier, here we train a list-wise ranking model that fits the ordinal or continuous nature of the data.

In our project, we use a list-wise learning to rank approach based on the popular ranking measure, Discounted Cumulative Gain (DCG) [8]. Similar to the IRGAN, we also let the generative model and the discriminative model to play a minimax game: the generative model tries to select relevant items that look like the ground truth and therefore could fool the discriminative model, whereas the discriminative model would try to rank the ground truth item and the selected ones based on the user preference. Our overall objective can be defined as:

$$J^{G^*,D^*} = \min_{\theta} \max_{\phi} \sum_{u} E_{x\sim p_\theta, y\sim p_{true}}[DCG@N(\phi, x, y)] \tag{5}$$

Since the user may only observe a small fraction of the item pool, to learn the complete order of the selected items may be biased due to the effect of the large number of long-tail items. However, we might have greater confidence to merely order the top-ranking items and lay less concern on the rest. Therefore, for our objective in Eq. (5), we design a minimax game aim at the cut-off ranking measure, DCG@N, for arbitrary user u, DCG@N of u is defined as:

$$DCG@N = \sum_{i=1}^{m} I(R_i < N) \cdot \frac{2^{f_i} - 1}{log(R_i + 2)} \tag{6}$$

where f_i is the rating value of item i which is computed as in Eq. (1) and R_i is the ranking index of i in the selected item list, which is computed as in Eq. (3). Since the indicative function $I(\cdot)$ is non-differentiable, one cannot adopt the off-the-shelf gradient descent to optimize the objective. A popular solution for this issue is to use the smooth function to substitute the non-differentiable indicative function. As sigmoid function is widely used in the existing literatures [9], [10], we adopt the sigmoid function to replace the indicative function, which yields:

$$I(f_i < f_j) = \sigma(f_j - f_i) \tag{7}$$

We further denote the smoothed version of DCG@N as sDCG@N.

2.1. Optimizing discriminative model The objective for the discriminator is to maximize the cut-off ranking measure so to learn the proper ranking order of the selected items. With the observed user feedback (ratings) of items as well as the ones sampled from the current challenger, the trained generative model $p_\theta(i|u)$, one can train the parameters for the discriminative model:

$$\phi^* = \arg\max \phi \sum_{u} E_(x \sim p_\theta, y \sim p_{true})[sDCG@N(\phi, x, y)] \tag{8}$$

The above objective can be solved by stochastic gradient descent.

2.2. Optimizing generative model The generative model $p_\theta(i|u)$ aims to sample relevant items from the item pool in order to fool the discriminative model. Therefore, it intends to minimize the cut-off ranking measure while keeping the discriminator $f_\phi(u, i)$ fixed after its maximization, we learn the generative model via performing its minimization:

$$\theta^* = \arg\min \theta \sum_{u} E_(x \sim p_\theta, y \sim p_{true})[sDCG@N(\phi, x, y)] \tag{9}$$

Similar to the analysis in [3], the sampling of item x is discrete. A common approach is to use policy gradient based reinforcement learning. The gradient is derived as follows:

$$\begin{aligned}
\bigtriangledown J^G(i|u) &= \bigtriangledown_\theta E_(x \sim p, y \sim p_{true})[sDCG@N(\phi, x, y)] \\
&= \sum_{x} \bigtriangledown_\theta p_\theta(x|u) E_(y \sim p_{true})[sDCG@N(\phi, x, y)] \\
&= \sum_{x} p(x|u) \bigtriangledown_\theta \log p_\theta(x|u E_(y \sim p_{true})[sDCG@N(\phi, x, y)] \\
&= E_(x \sim p_\theta) \bigtriangledown_\theta \log p_\theta(x|u E_(y \sim p_{true})[sDCG@N(\phi, x, y)] \\
&\approx \frac{1}{K} \sum_{i=1}^{K} \bigtriangledown_\theta \log p_\theta(x|u E_(y \sim p_{true})[sDCG@N(\phi, x, y)]
\end{aligned} \tag{10}$$

With the terminology of reinforcement learning, the term $\log p_\theta(x|u) E_(y \sim p_{true})[sDCG@N(\phi, x, y)]$ acts as the reward for the policy $p_\theta(x|u)$ taking an action x in the environment u.

The conditional distribution of $p_\theta(x|u)$ can be specified as:

$$p_\theta(x|u) = \frac{\exp(f_\theta(u, x)/\tau)}{\sum_x \exp(f(u, x)/\tau)} \tag{11}$$

where the computation of $f_\theta(u, x)$ is as Eq. (1).

The overall logic of our proposed List-IRGAN solution is summarized in Algorithm 1.

Algorithm 1 List-IRGAN

Input: generator $p_\theta(i|u)$; discriminator $f_\phi(u, i)$; training dataset S

1. Initialize $p_\theta(i|u), f_\phi(u, i)$ with random values θ, ϕ.
2. Pre-train $p_\theta(i|u), f_\phi(u, i)$ using S.
3. repeat
4. for g-steps do
5. $p_\theta(i|u)$ generates K items for each user u
6. Update generator parameters via policy gradient as Eq. (10)
7. end for
8. for d-steps do
9. Use current $p_\theta(i|u)$ to generate items and combine with observe items in S
10. Train discriminator $f_\phi(u, i)$ by Eq. (8)

11. end for
12. until List-IRGAN converges

Empirical Analysis

In the experiment, we conducted our experiments by using three different models – IRGAN, LIST-IRGAN and a simple binary classifier based on the Movielens datasets, which is a widely used collaborative filtering dataset. We record the Precision(P) score and Normalized Discounted Cumulative Gain(NDCG) to compute the predicted results of both of the generator and discriminator of the three methods in terms of top 3, 5, 10 items respectively. However, due to the simple binary classifier does not have the generator pair, so we keep the same scores to compare it with other 2 models.

Experimental Results

As we can see on table1, our model's scores of generators are close to the IRGAN model. And the scores of IRGAN are a little lower than that on original IRGAN paper, though we have the limitation of computational resources and didn't train the model into the best. But the setup of the three models are the same. When it comes to the results of discriminator, we find a wired thing that IRGAN's discriminator doesn't perform very well. And their scores of discriminator continuously decrease, which means the generator and discriminator don't develop equally. Same phenomenon happens in our model as well. Generally, our model has a higher score on discriminator and performs better than the IRGAN model.

Table 1: Results of Generator

Model	P@3	P@5	P@10
IRGAN	0.3845	0.3487	0.2991
LISTIRGN	0.3860	0.3434	0.2956
BASELINE	0.0980	0.0996	0.0906
	NDCG@3	NDCG@5	NDCG@10
IRGAN	0.3996	0.3769	0.3542
LISTIRGN	0.4004	0.3727	0.3513
BASELINE	0.0996	0.1006	0.0966

Table 2: Results of Discriminator

Model	P@3	P@5	P@10
IRGAN	0.0504	0.0526	0.0506
LISTIRGN	0.1403	0.1184	0.1015
BASELINE	0.0980	0.0996	0.0906
	NDCG@3	NDCG@5	NDCG@10
IRGAN	0.0471	0.0494	0.4942
LISTIRGN	0.1587	0.1370	0.1230
BASELINE	0.0996	0.1006	0.0966

Discussion and Future work

As shown in the empirical analysis section, our method, List-IRGAN, shows little improvement over IRGAN on the generative model; whereas the improvement on the discriminative model is significant. The reason behind the improvement on the discriminative model is intuitive. That is, the list-wise learning model indeed bring in significant possitive effect to help the discriminative model to learn more abundant information. However, as on the generative model, the experimental results showcase that ther performance of List-IRGAN is almost identical. The reason behind this is still unknown to us, we should do more experiments on it in the future.

One of the interesting observation on the learning process of both IRGAN and List-IRGAN is that, while the generative model keeps its performance almost unchanged as the epoch grows, the performance of the discriminative model keeps decreasing. This indicates that the generative model is dominating the minimax game, which is undesirable for both parts of the method. Perhaps the current settings on the overall method is unbalanced. Maybe parameter tuning is necessary for further experiments.

Notice that our work in this project is rather rudimentary, there are still a lot of additional works could be done. For example, the experimental settings of our project follow the same assumption of IRGAN. For IRGAN, all the selected items produced by the generative model is negative; whereas in our model, the ranking positions of selected items are not among the top-n list. However, its problematic to follow this assumption as we could not be sure whether the unobserved items are preferred by the user or not. A more appropriate model could be considered to predict the ranking positions of the unobserved items (selected items of the generative model) rather than merely precluding them from the top-ranking list. We leave this as the future work of our model.

Before we draw any conclusions about the performance our model, more experiments should be included. For example, more datasets should be tested so to understand the performance of our model in an unbiased way.

Conclusion

This study has provided enough analysis of current various model for recommender system including IRGAN and the traditional binary classification model. In this project, we sought to shed light on the List-IRGAN which can be a better recommendation model. Inspired by IRGAN, we made the generative model and the discriminative model work together by optimizing each of them alternately. Based on the MovieLens dataset we could get, several experiments had been conducted focused on the difference of P and NDGC aimed to compare the efficiency among IRGAN, List-IRGAN and simple binary classifier. Results shows that List- IRGAN works the same with IRGAN in the generator system, while List- IRGAN works better than IRGAN in the Discriminator system. This finding contributes to an overall understanding of List-IRGAN which is a list-wise learnig-to-rank framework using continues rating data. here are our term project web page. https://chosenchen95.wixsite.com/aitermproject.

References

[1] S. Zhang, L. Yao, and A. Sun, Deep learning based recommender system: A survey and new perspectives, ArXiv Prepr. ArXiv170707435, 2017.

[2] I. Goodfellow et al., Generative adversarial nets, in Advances in neural information processing systems, 2014, pp. 26722680.

[3] J. Wang et al., IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models, ArXiv Prepr. ArXiv170510513, 2017.

[4] C. C. Aggarwal, Model-based collaborative filtering, in Recommender Systems, Springer, 2016, pp. 71138.

[5] F. M. Harper and J. A. Konstan, The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. TiiS, vol. 5, no. 4, p. 19, 2016.

[6] J. Bennett and S. Lanning, The netflix prize, in Proceedings of KDD cup and workshop, 2007, vol. 2007, p. 35.

[7] A. Karatzoglou, L. Baltrunas, and Y. Shi, Learning to rank for recommender systems, in Proceedings of the 7th ACM conference on Recommender systems, 2013, pp. 493494.

[8] N. Ifada and R. Nayak, Do-Rank: DCG optimization for learning-to-rank in tag-based item recommendation systems, in Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2015, pp. 510521.

[9] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic, CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering, in Proceedings of the sixth ACM conference on Recommender systems, 2012, pp. 139146.

[10] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, and A. Hanjalic, xCLiMF: optimizing expected reciprocal rank for data with multiple levels of relevance, in Proceedings of the 7th ACM conference on Recommender systems, 2013, pp. 431434.