Adversarial Training for Review-Based Recommendations

Dimitrios Rafailidis
Maastricht University
Maastricht, The Netherlands
dimitrios.rafailidis@maastrichtuniversity.nl

Fabio Crestani
Faculty of Informatics
Università della Svizzera italiana (USI)
Lugano,Switzerland
fabio.crestani@usi.ch

ABSTRACT

Recent studies have shown that incorporating users' reviews into the collaborative filtering strategy can significantly boost the recommendation accuracy. A pressing challenge resides on learning how reviews influence users' rating behaviors. In this paper, we propose an Adversarial Training approach for Review-based recommendations, namely ATR. We design a neural architecture of sequence-to-sequence learning to calculate the deep representations of users' reviews on items following an adversarial training strategy. At the same time we jointly learn to factorize the rating matrix, by regularizing the deep representations of reviews with the user and item latent features. In doing so, our model captures the non-linear associations among reviews and ratings while producing a review for each user-item pair. Our experiments on publicly available datasets demonstrate the effectiveness of the proposed model, outperforming other state-of-the-art methods.

CCS CONCEPTS

• **Information systems → Collaborative and social computing systems and tools**.

KEYWORDS

Recommendation systems; adversarial training; neural models

ACM Reference Format:
Dimitrios Rafailidis and Fabio Crestani. 2019. Adversarial Training for Review-Based Recommendations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331313

1 INTRODUCTION

Nowadays, customers can write and share reviews on most e-commerce and social media platforms. Provided that reviews are informative and powerful in capturing user preferences, there has been immense interest in collaborative filtering systems that exploit users' reviews for improving recommendations. The main challenge is to learn how reviews influence the rating behaviour of users. For example, in the context of review-based recommendation Diao et al. [5] combine latent feature-based collaborative filtering and probabilistic topic modeling to provide a latent structure for users and items based on reviews. Collaborative Topic Regression learns the latent structure from user generated content so that probabilistic topic modeling can be integrated into collaborative filtering [16]. In the Hidden Factor Topics (HTF) model [8], the latent topic distribution of each item is learned by applying Latent Dirichlet Allocation on its reviews. Then by considering these distributions as the items' latent features, the users' latent features are estimated by optimizing the rating prediction accuracy with gradient descent. However, these models do not account for the non-linear associations of users' reviews and rating behaviors.

With the advent of deep learning strategies in recommendation systems, several neural architectures have been introduced to capture the non-linearity in users' reviews and ratings. For instance, the Collaborative Deep Learning mode of [17] jointly performs collaborative filtering for ratings and deep representation learning for the textual information. The Deep Cooperative Neural Networks (DeepCoNN) [20] model user-item interactions based on review texts by utilizing a factorization machine model on top of two convolutional neural networks. The TNET model [3] extends the DeepCoNN model by introducing an additional layer that is regularized by the latent representation of reviews, enforcing the regularized representation to be similar to the embedding of the actual target review. Shalom et al. [15] propose a CNN model to compute how compatible reviews are with users' ratings. Meanwhile, Recurrent Neural Networks (RNNs) have been applied to various Natural Language Processing (NLP) tasks with great success. For instance, bidirectional RNNs, including bidirectional Long Short-Term Memory and bidirectional Gated Recurrent Unit (GRU) can encode a target word with contextual and sequential information, that is encoding not just the target word but also the surrounding words when using sentences of documents as input sequences [19]. To exploit the computational power of the neural models in NLP tasks, Lu et al. [7] propose a Topical Attention Regularized Matrix Factorization (TARMF) model, which co-learns user and item information from ratings and reviews by optimizing matrix factorization and an attention-based GRU network. Nonetheless, what is missing from existing deep learning strategies is that they omit the fact that producing new (machine-generated) reviews while learning the influence of existing (groundtruth) user reviews on the rating behavior can significantly boost the recommendation accuracy.

Recently, IRGAN [18] was the first attempt to propose a minimax game framework of Generative Adversarial Networks (GANs) to train Information Retrieval systems, using matrix factorization for the discriminator and generator to produce recommendations. Chae et al. [4] introduce a vector-wise adversarial training model for collaborative filtering, while Lim et al. [2] jointly learn GANs

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331313

and RNNs for sequential modeling of user preferences. However, existing adversarial approaches ignore users' reviews while producing recommendations.

To overcome the shortcomings of existing methods, in this paper we propose an Adversarial Training approach for Review-based recommendations, namely ATR. Our contributions in this paper can be summarized as follows. (i) We propose an adversarial training strategy of sequence-to-sequence learning to compute the deep representations of reviews and produce a machine-generated review for each user-item pair. (ii) We formulate an objective function to jointly learn the deep representations of reviews and user and item latent features, by letting them adjust to each other via an alternating optimization strategy. Evaluated against baseline methods, our experiments demonstrate the superiority of our approach, showing the importance of our adversarial-based joint learning strategy.

2 THE PROPOSED MODEL

The proposed ATR model consists of a users' review generator and discriminator. The goal of the users' review generator is to produce machine-generated reviews that are likely to be written by a user u. The goal of the discriminator is to distinguish the adversarial examples, that is the machine-generated reviews from the groundtruth/authentic reviews that have been actually written by user u. Similarly, we also model an item review generator and discriminator, aiming to produce machine-generated reviews on item j. In our model users' and items' review generators and discriminators are learned via adversarial sequential learning as we will show in Section 2.3. In addition, to predict the ratings in our model we follow a coupling strategy of adversarial training with matrix factorization (Section 2.4).

Initially, based on users' reviews on items, we create a document d_u for each user u that contains all the groundtruth reviews of u and similarly a document d_j with all the reviews on item j. Given the rating matrix $R \in \mathbb{R}^{n \times m}$, with n and m being the numbers of users and items, the goal of the proposed model is to compute the factorized matrix $\hat{R} \in \mathbb{R}^{n \times m}$ with the missing ratings, by considering both the groundtruth and machine-generated reviews.

2.1 The Generator

For simplicity, in this Section we detail the users' review generator. In a similar spirit we model the items' review generator. The goal of the review generator G_ψ is to learn to create machine-generated reviews to confuse the discriminator D_θ, where ψ and θ are the parameters of the generator and discriminator, respectively. We introduce an adversarial sequence-to-sequence learning strategy. In particular, we design an encoder-decoder architecture to generate reviews based on groundtruth user reviews in documents d_u, with $u = 1, \ldots, n$. The goal of the encoder is to map d_u to a k-dimensional review vector $\overline{U}_u$, while the decoder tries to generate a review that is similar to the document d_u. Notice that $\overline{U}_u$, represents the textual features of the document d_u.

In the generator G_ψ, based on a lookup operation the encoder first maps each word w_t of a review r in the sequence $(w_1, w_2, \ldots, w_t)$ into a k-dimensional vector q_t. Then, following [19] we use a bidirectional GRU to compute two different types vector representations of a review r, that is the forward and backward activations, which are then concatenated into a unified representation h_r. Having computed all the review representations h_r of a document d_u, the output of the encoder $\overline{U}_u$ is calculated by averaging all the concatenated review representations h_r. Accordingly, given the item review document d_j, the output of the encoder of the item review generator is a k-dimensional review vector $\overline{V}_j$, with $j = 1, \ldots, m$.

Then, the k-dimensional review vector $\overline{U}_u$ is provided to the decoder, trying to make it consistent with the review document d_u. To achieve this, the decoder estimates the following conditional probability:

$$p(z_{u1}, \ldots, z_{ul} | \overline{U}_u) = \prod_{t=1}^{l} p(z_{ut} | \overline{U}_u, z_{u1}, \ldots, z_{ut-1}) \quad (1)$$

where $(z_{u1}, \ldots, z_{ul})$ is the predicted (machine-generated) review with length l. The decoder employs a single GRU that iteratively produces reviews word by word. In particular, at time step t the GRU first maps the output representation z_{ut-1} of the previous time step into a k-dimensional vector y_{ut-1} and concatenates it with $\overline{U}_u$ to generate a new vector y_{ut}. Finally, y_{ut} is fed to the GRU to obtain the hidden representation h_t, and then h_t is multiplied by an output projection matrix and passed through a softmax over all the words in the vocabulary of the document to represent the probability of each word. The output word z_{ut} at time step t is sampled from the multinomial distribution given by the softmax.

2.2 The Discriminator

Given a candidate review r, either authentic or machine-generated, and a target user u, the discriminator D_θ has to estimate the probability that the review r is authentic. To achieve this we employ a CNN. Each word of the review r is mapped to the corresponding word vector, which is then concatenated with a user-specific vector. Notice that the user-specific vectors are learned together with the parameters of the discriminator D_θ in the adversarial training of Section 2.3. The concatenated vector representations are then processed by a convolutional layer, followed by a max-pooling layer and a fully-connected projection layer. The final output of the CNN is a sigmoid function which normalizes the probability into the interval of $[0, 1]$, expressing the probability that the candidate review r is written by user u.

2.3 Adversarial Training

The objective of adversarial training in our model is to maximize the cumulative reward of the generator G_ψ in each epoch of the training, and maximize the reinforcement reward of the discriminator D_θ, following the training strategies of [4, 18]. In our implementation the discriminator D_θ is trained via gradient descent, by minimizing the probability of classifying adversarial (machine-generated) reviews to be authentic, while maximizing the probability of classifying users' groundtruth reviews as authentic ones. For each epoch of the adversarial training, we first draw the same number s of random samples from (i) users' groundtruth reviews and (ii) machine-generated reviews of the generator G_ψ, denoted by $X \sim P_{gr}$ and $X' \sim G_\psi$, respectively. The objective of adversarial training is formulated as follows:

$$\max_\theta L = E_{X \sim P_{gr}} \left[\log D_\theta(X) \right] + E_{X' \sim G_\psi} \left[\log(1 - D_\theta(X')) \right] \quad (2)$$

We learn the parameters θ of the discriminator D_θ via gradient ascent. To compute the parameters ψ of the generator G_ψ we apply a policy-gradient strategy [2], calculating the gradient of L of Equation 2 with respect to ψ as follows:

$$\nabla_\psi L = E_{X' \sim G_\psi} \nabla_\psi \log G_\psi(X')[\Phi(X') - b(X')] \quad (3)$$

with

$$E_{X' \sim G_\psi} \nabla_\psi \log G_\psi(X') = \frac{1}{s} \sum_{i=1}^{s} \nabla_\psi \log G_\psi(X'_i) \quad (4)$$

where X'_i denotes the i-th sample/review drawn from the generator G_ψ. In Equation 3, $\Phi(\cdot)$ is the reward function of the discriminator and $b(\cdot)$ is the baseline value to reduce the variance [18].

2.4 Coupling with Matrix Factorization

The objective function of matrix factorization tries to minimize the approximation error of the factorized matrix $\hat{R} \in \mathbb{R}^{n \times m}$ by computing the latent matrices $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{m \times k}$, where k is the number of latent dimensions. More formally, this is expressed by the following minimization problem: $\min_{U,V} \sum_{(u,j)} ||R_{uj} - U_u V_j^\top||^2$, which can be solved by gradient descent, that is computing the gradients with respect to matrices U and V and update the matrices/variables accordingly. In addition, at the same time we have to account for the inferred user preferences by matrix factorization, when learning users' and items' review documents. To achieve this, we consider the adversarial learning of the reviews of Section 2.3, by adjusting the k-dimensional review vectors $\overline{U}_u$ and $\overline{V}_J$ close to the k-dimensional latent vectors U and V. This is expressed by the approximation errors of $||U_u - \overline{U}_u||^2$ and $||V_j - \overline{V}_j||^2$, respectively. Thus, the joint objective function of our model is defined as follows:

$$\min \sum_{(u,j)} ||R_{uj} - U_u V_j^\top||^2 + \lambda \sum_u ||U_u - \overline{U}_u||^2 + \gamma \sum_j ||V_j - \overline{V}_j||^2 \quad (5)$$

where the regularization parameters λ and γ control the influence of the users' and items' latent matrices on the joint objective function of Equation 5, respectively. As the joint objective function is not convex with respect to the four variables/matrices we follow an alternating optimization strategy, that is update one variable while keeping the remaining fixed. Given the maximum number $maxIter$ of iterations/epochs, the steps of our alternating optimization strategy are summarized as follows:

$\forall$ epoch=1 $\leftarrow$ $maxIter$

-Step 1: update U_u via gradient descent, $\forall\ u = 1 \leftarrow n$
-Step 2: update V_j via gradient descent, $\forall\ j = 1 \leftarrow m$
-Step 3: update ψ_U and ψ_V via policy-gradient for the review generators of users' and items' review documents
-Step 4: update θ_U and θ_V via gradient ascent for the review discriminators of users' and items' review documents

where ψ_U and ψ_V are the model parameters of the generators of users' and items' review documents, respectively. Similarly, θ_U and θ_V are the model parameters of discriminators of users' and items' review documents. Notice that in each epoch the reviews' deep representations $\overline{U}$ and $\overline{V}$ and the latent matrices U and V regularize each other based on our alternating optimization strategy.

In our implementation, before updating the model parameters based on steps 1-4, we initialize the encoders of both users' and items' review generators with *word2vec* [10] to compute pre-trained word vectors and tune the parameters via backpropagation. Also, following [6], for each epoch before updating ψ_U and ψ_V via policy-gradient at step 3, we first provide the users' and items' review generators only with groundtruth reviews to initialize ψ_U and ψ_V, and then apply the policy-gradient strategy with both groundtruth and machine generated reviews as described in Section 2.3.

3 EXPERIMENTS

3.1 Setup

Our experiments were performed on the four datasets "Beauty", "Health", "Cell Phone" and "Clothing" from the publicly available Amazon Review Collection [9], having in total 823,534 reviews and 10,754,316 ratings at a 5-star scale. In our experiments we train the models on 50% of the ratings, 10% is used for cross-validation to tune the models' parameters and the remaining 40% is used as test set. In the Amazon Review Collection, the datasets are preprocessed in a 5-core fashion, that is each user and item have at least 5 reviews to be included. We tokenize the reviews using the NLTK API and retain words that appear at least ten times in the vocabulary. We would like to mention that when training the generators and discriminators to build user and item representations based on the respective review documents, all reviews that are associated with user-item ratings from the test set are not included.

Following the evaluation protocol of relevant studies [3, 7, 8], we measure the performance of the examined models in terms of Mean Squared Error (MSE) which is defined as follows:

$$\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} (\hat{R}_{ui} - R_{ui})^2 \quad (6)$$

where $\mathcal{T}$ is the set of ratings in the test set, $\hat{R}_{ui}$ the predicted rating by the model and R_{ui} is the groundtruth rating in the test set. We repeated our experiments five times and we report average MSE over the five trials.

In our implementation, we set λ=γ=1e-03 and fix the maximum number of iterations/epochs of our alternating optimization strategy to $maxIter$=40. The influence of the number of dimensions k of the latent matrices U and V, and the review deep representations of $\overline{U}$ and $\overline{V}$ on the proposed ATR model is presented in Figure 1.

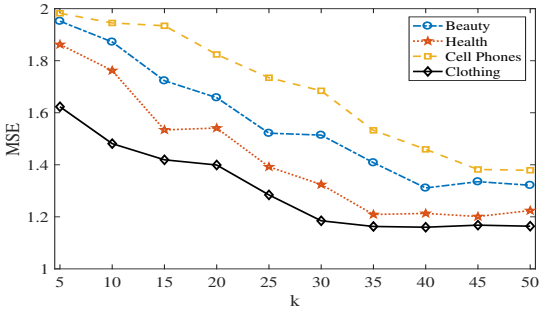

Figure 1: Effect on MSE by varying k.

We compare the proposed ATR model of Adversarial Training for Review-based recommendations with the baseline PMF model [14]

Table 1: Effect on MSE of the examined models on the four evaluation datasets. Bold values denote the best scores.

	Beauty	Health	Cell Phones	Clothing
PMF	1.950	1.882	1.972	2.396
HFT	1.552	1.415	1.606	1.535
DeepCoNN	1.435	1.299	1.524	1.322
TNET	1.404	1.249	1.438	1.197
TARMF	1.387	1.238	1.413	1.187
ATR	**1.311**	**1.203**	**1.382**	**1.163**

that ignores users' reviews on items, and the HFT model [8] that takes into account users' reviews when producing recommendations. We also compare our model with the review-based deep learning strategies of DeepCoNN [20], TNET [3] and TARMF [7] (Section 1) that try to capture the non-linearity among reviews and ratings based on different neural architectures.

3.2 Results

In Table 1, we report the effect on MSE of the examined models on the four evaluation datasets. We observe that all the models that exploit users' reviews achieve a lower MSE than the baseline PMF model, indicating that reviews can help improve the recommendations. The neural models of DeepCoNN, TNET, TARMF and ATR can better capture the non-linear associations between reviews and ratings, thus outperforming the baseline model of HFT. Clearly, the proposed ATR model beats all the competitors. We would like to highlight that both TARMF and the proposed ATR model follow a sequence-to-sequence learning approach when factorizing the rating matrix, which explains their superiority over the other baselines. Notice that sequence-to-sequence learning is a key factor in encoding reviews with contextual and sequential information [19]. On inspection of Table 1, we observe that the proposed ATR model achieves a lower MSE than the most competitive model of TARMF in all four evaluation datasets. This happens because not only ATR encodes the sequential information of words of users' reviews as TARMF, but ATR also tries to produce machine-generated reviews of each user-item pair according to our adversarial-based joint learning strategy, thus better adjusting the deep representations of reviews close to the user and item latent features. In doing so, the proposed ATR model better models the influence of reviews on users' behaviors than TARMF, thus significantly improving the recommendation accuracy.

Then, we evaluate the performance of the users' and items' review discriminators by measuring the accuracy to classify the reviews written by a user and on an item from other reviews. In this set of experiments we use as test-reviews those that are written for each user-item pair in the test set. The goal of the users' review discriminator is to predict whether a review is written by a user u, and similarly the items' review discriminator determines whether a review is written on a item j. In our experiments on all four evaluation datasets, the users' discriminator achieves an accuracy of 76.2%, while the items' discriminator classifies reviews correctly at 83.7%. The reason that the items' discriminator has higher classification accuracy than the users' discriminator is that the review document of an item is more semantically coherent than a review document of a user. This occurs because users can review various items/products that might be semantically different with each other, thus making the users' review documents less semantically coherent than the items' review documents.

4 CONCLUSIONS

We presented an adversarial training approach for review-based recommendations, where we jointly learn the latent features of users' ratings with the deep representations of reviews. By generating a review for each user-item pair and jointly factorizing user preferences, we demonstrated that our model can capture well the non-linearily in users' reviews and ratings. An interesting future direction is to extend the proposed model to exploit users' reviews as an effective means of bridging the gap of heterogeneous items on different social media platforms or e-commerce sites to produce cross-domain recommendations [1], by taking into account users' preference dynamics [13] and social relationships [11, 12].

REFERENCES

[1] Mohammad Aliannejadi, Dimitrios Rafailidis, and Fabio Crestani. 2017. Personalized Keyword Boosting for Venue Suggestion Based on Multiple LBSNs. In *ECIR*. 291–303.

[2] Homanga Bharadhwaj, Homin Park, and Brian Y. Lim. 2018. RecGAN: recurrent generative adversarial networks for recommendation systems. In *RecSys*. 372–376.

[3] Rose Catherine and William W. Cohen. 2017. TransNets: Learning to Transform for Recommendation. In *RecSys*. 288–296.

[4] 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 *CIKM*. 137–146.

[5] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *KDD*. 193–202.

[6] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial Learning for Neural Dialogue Generation. In *EMNLP*. 2157–2169.

[7] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Co-Evolutionary Recommendation Model: Mutual Learning between Ratings and Reviews. In *WWW*. 773–782.

[8] Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*. 165–172.

[9] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*. 43–52.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*. 3111–3119.

[11] Dimitrios Rafailidis and Fabio Crestani. 2016. Collaborative Ranking with Social Relationships for Top-N Recommendations. In *SIGIR*. 785–788.

[12] Dimitrios Rafailidis and Fabio Crestani. 2017. Learning to Rank with Trust and Distrust in Recommender Systems. In *RecSys*. 5–13.

[13] Dimitrios Rafailidis, Pavlos Kefalas, and Yannis Manolopoulos. 2017. Preference dynamics with multimodal user-item interactions in social media recommendation. *Expert Syst. Appl.* 74 (2017), 11–18.

[14] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*. 1257–1264.

[15] Oren Sar Shalom, Guy Uziel, Alexandros Karatzoglou, and Amir Kantor. 2018. A Word is Worth a Thousand Ratings: Augmenting Ratings using Reviews for Collaborative Filtering. In *ICTIR*. 11–18.

[16] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*. 448–456.

[17] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *KDD*. 1235–1244.

[18] 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 *SIGIR*. 515–524.

[19] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *NAACL HLT*. 1480–1489.

[20] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM*. 425–434.