

RESEARCH ARTICLE

Graph Fusion in Reciprocal Recommender Systems

LUWEI ZHANG¹ (Graduate Student Member, IEEE), XUETING WANG^{1,2}, (Member, IEEE), AND TOSHIHIKO YAMASAKI¹, (Member, IEEE)

¹Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan

²AI Laboratory, CyberAgent, Inc., Tokyo 150-0042, Japan

Corresponding author: Luwei Zhang (zhangluwei@cvm.t.u-tokyo.ac.jp)

This work was supported by the Institutional Review Board (IRB) of Graduate School of Information Science and Technology, The University of Tokyo, under Grant UT-IST-RE-210415.

ABSTRACT Unlike traditional user-item recommendation tasks (e.g., movie or consumer-product recommendation), reciprocal recommender systems (RRSs) (e.g., online dating services and job-recruitment sites) must consider the interests of both two users. Pair matching prediction can improve the efficiency with which RRSs match potential partners. Graph Neural Networks (GNNs) are powerful models for learning representations of attributed graphs and information circulation between nodes. GNNs greatly facilitate link prediction in the area of user-item recommender systems but have not been extensively applied to RRS. In this study, we present a novel method for pair matching prediction that learns the reciprocal information circulation between users: not only side information about them but also structural information about their behavior histories. In contrast to earlier RRSs, which focus on response prediction, ours predicts both send and reply signals. Moreover, we introduce negative sample mining to explore the effect of different types of multiple samples on recommendation accuracy in real applications. Testing our method on data provided by an online dating service, we achieved an AUC of 73.15% (an absolute improvement of over 3.20% point above baseline) and an AP of 26.01% (an absolute improvement of over 2.79%) on send prediction; an AUC of 68.95% (an absolute improvement of over 1.74%) and an AP of 23.02% (an absolute improvement of over 0.70%) on reply prediction; an AUC of 71.26% (over 4.35% point absolute improvement) and an AP of 23.95% (over 0.30% point absolute improvement) on fusion reciprocal prediction.

INDEX TERMS Reciprocal recommender systems, online dating, graph neural networks, social media.

I. INTRODUCTION

Reciprocal Recommender Systems (RRSs) [1] recommends people or people-like items (e.g., corporations in the case of an employment site) to other people; a successful recommendation only occurs when the needs of both persons in the pair are met. Figure 1 illustrates the difference in data construction between user-item recommendation and reciprocal recommendation. In user-item recommendation, inputs are features of users and items; labels indicate whether the users clicked on the items. In reciprocal recommendation, inputs are features of two users; labels indicate whether the users

made match. The difference in labels causes the difference in the recommendation policy. In typical recommendation tasks, users are potentially interested in items, but not vice versa: a recommendation can be made once it is established that the item meets the user's needs. The process is unidirectional and the recommendation list is for the user only. However, in RRS tasks, success is determined by both parties and detailed profiles of both are required.

An RRS also needs to consider additional factors that are not present in conventional user-item recommender systems. Difficult situations may result from recommending a popular user to an unpopular one, or one passive user to another [2]. Therefore, RRSs are inherently more complicated to design and build than conventional recommenders. Although RRSs

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina ¹.

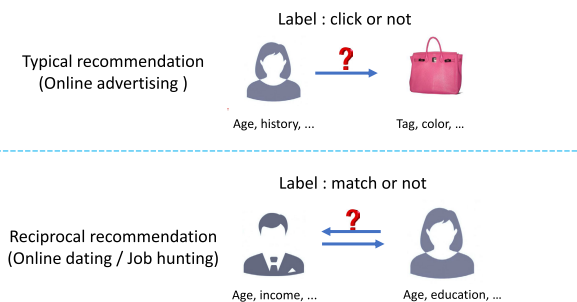


FIGURE 1. Comparison of traditional user-item recommendation and reciprocal recommendation.

have been developed for job recruitment [3], [4] and social networks [5], the most common application has been to online dating services.

In this work, we design reciprocal recommendation methods for online dating. Figure 2 shows the formulation of the pair matching prediction. A number of collaborative filtering-based RRSs for online dating services [6], [7] have been proposed in the literature; there are also categorical content-based RRSs such as RECON [8]. In contrast, Zhang et al. [9] treated pair matching prediction as click-through rate (CTR) prediction. Feature interaction based methods in [9] are content-based methods that employ Factorization Machines (FMs) [10] and their variants combined with deep neural networks (DNNs).

Embedding, which is also called representation learning, is a proven and widely used machine learning technique [11]. Because of its massive success in other domains such as computer vision (CV) and natural language processing (NLP), it has been an active research topic in the information retrieval community and recommender system industry as the next generation technology [12] as well. Embedding (sometimes called semantic embedding in that it can often learn semantics.) represents the sparse vector corresponding to a categorical feature as a dense feature vector. In particular, in recommender systems, categorical features are the main components of the dataset in use. Therefore, a proven business-specific embedding method is essential.

Collaborative filtering-based methods are widely and successfully employed for embedding in recommender systems; the structure information in a user’s behavior history network is highly likely to enhance the rate of predictive success. There have recently been great advances in the application of graph neural networks (GNNs) to recommendation tasks [13], [14], [15]; GNNs offer as a message passing method which learns both the rich side information included in the user profile and the structural information in the user’s historical interactions.

Previous RRSs performed like response prediction simply by predicting the final status label. In real applications, however, both directions of user preference should be considered, which means the initial interaction should also be predicted. Therefore, we decompose the final status labels (considered

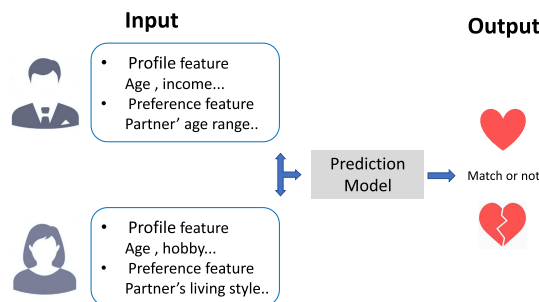


FIGURE 2. Formulation of pair matching prediction in reciprocal recommenders. The inputs are users’ profile such as age, residence, hobby, income, living habits (smoke or not/drink or not etc.) and so forth. The outputs are the prediction results of two users’ matching.

as reciprocal signals) into several kinds of links/edges in the graph formulation. The reciprocal recommendation task is also transferred into both send prediction and reply prediction simultaneously. We evaluate the performance of models based on this formulation on a real dataset, using two types of metrics.

Negative sample mining (also called training data mining in Embedding-based Retrieval (EBR)) is gaining more attention in information retrieval [16]. In this approach, the failure of a user to click is not the only way in which a sample can be classified as negative. They propose the idea of easy negative/hard negative sample grading: easy negative samples refer to those not impressed items/products; hard negative ones refer to those impressed but not clicked ones. The same perspective on negative sample diversity is used in the Airbnb recommendation [17] and Mobious [18]. It is vital to keep the offline training data distribution consistent with the data distribution of the actual online application by introducing the easy/hard negative sample grading to enhance the negative sample diversity. Otherwise, the real recommendation results will exhibit disorderliness because the trained model has not seen huge amounts of random samples in real business.

In this paper, we make the following technical contributions:

- We reformulate the pair matching prediction task in RRSs into simultaneous send prediction and reply prediction (rather than predicting reciprocal signals only).
- We propose employing GNNs to capture high-order message passing flows between users in a reciprocal recommendation: this enables learning both the side information from user profiles and structural information about historical interactions.
- We apply negative sample mining (or named training data mining) to keep the offline training data distribution consistent with that of the actual online application, thus enhancing the diversity of samples and the reliability of the model in real business.
- The experimental results on the real-world dataset provided by a collaborating corporation demonstrate that our model outperforms baseline models based on

feature interaction. For send prediction, the absolute improvements in terms of AUC and AP are over 3.20% point absolute improvement and over 2.79% point absolute improvement, respectively; for reply prediction, over 1.74% point absolute improvement and over 0.70% point absolute improvement; for fusion (i.e. reciprocal prediction) by over 4.35% point absolute improvement and by over 0.30% point absolute improvement.

The remainder of this paper is organized as follows. Section II introduces some prior studies that are related to our work. Then, in Section III, we start with dataset reformulation to present our proposal. Section IV introduces the experiments for evaluating the effectiveness of the proposed framework and summarizes the experimental results. Finally, Section V presents our conclusions.

II. RELATED WORKS

In this section, we review existing works on feature interaction based recommendation, graph based recommendation, and negative sample mining.

A. FEATURE INTERACTION BASED METHODS

With its powerful expressive ability and flexible network structure, deep learning has made major breakthroughs in many fields such as NLP, image, and voice processing [19]. In the field of recommendation, however, the feature space is too large and sparse to be modeled only by DNNs or Multilayer Perceptrons (MLPs). Therefore, recent research on DNN-based recommendation has revolved around projecting the high-dimensional categorical vector of DNNs onto a low-dimensional dense input through the embedding layer. Wide & Deep [20] is an example of a network that has established the mainstream framework. The network consists of two components, a wide (shallow) part and a deep part (stack layer). The following works mainly make improvements based on it. The wide part of Wide & Deep is logistic regression and deep part is embedding plus MLP. Wide & Deep is an end-to-end model but the wide part do not share the embedding result. There have numerous attempts to improve Wide & Deep. DeepFM [21] is an end-to-end model that performs automatic feature interaction. DeepFM applies FM as a wide part to capture low-order feature information; the deep part shares the embedding output with the FM part. Neural factorization machine (NFM) [22] seeks to improve the embedding output or the input of the DNN part; there is a bi-interaction layer between the embedding layer and the DNN as a pooling function. For comparison, DeepFM does not have bi-interaction layer so it concatenates all the output of embedding layer so the dimension of DNN input is $f * k$ (f is the number of field and k is the dimension of embedding size). The advantage of NFM is that the network parameters are directly compressed from n to k (which is less than the $f * k$ of DeepFM), which reduces the network complexity and accelerates the training of the network to obtain the model. However, at the same time, this method

may also cause larger information loss. The FM-based deep model mentioned above allows different feature vectors to directly make feature intersections. The assumption is that the contribution of each feature interaction to the prediction result is the same, which is actually unreasonable. The idea is somehow the same as FFM. Attentional Factorization Machines (AFM) [23] apply the attention mechanism that has been successful in the fields of NLP, image, and voice in recent years to solve this problem. Several other deep learning models for CTR recommendation are proposed based on FMs. FNN [24] is a proposal which applies a pre-trained FM before the embedding layer to gain good performance so it is not an end-to-end model and cannot capture low-order features very well. PNN [25] provides an idea of product layer to identify feature interactions with inner product or outer product.

Besides the mentioned methods, many other models have good performance in CTR recommendation tasks, such as tensor-based model [26], support vector machine based model [27], and Bayesian based model [28]. There are several tasks other than CTR prediction and pair matching prediction. Sedhain et al. [29] and Wang et al. [30] propose to enhance collaborative filtering via deep learning. Chen et al. [31] develop a deep network in multimedia area which use both image features and user features in display advertising. Covington et al. [32] applied a deep network for YouTube video recommendation.

B. GRAPH BASED RECOMMENDATION METHODS

Recently, there is increasing interest in graph convolutional networks (GCNs) [33] because of the generality and effectiveness on graph data. Due to the marvelous performance of GCNs in graph data analysis, GCNs are adopted into many fields with graph-structured data to learn the correlation between target objects, such as hyperspectral image classification in CV [34], [35], [36], [37], [38], natural language sentences matching [39] and question answer matching [40] in NLP.

Recommendation is a perfect field for GCNs in particular. Early efforts such as ItemRank [41], [42] exploit the user-item interaction graph to explore high-order proximity. HOP-Rec [43] combines graph-based and embedding-based methods, using random walk method to improve the recommendation results. NGCF [44] follows the same idea with HOP-Rec of taking advantage of graph embedding for user-item recommendation. GC-MC [45] employs GCN [33] on the user-item bipartite graph first but only one convolutional kernel is exploited; therefore high-order message passing flows are not embedded into representation learning. Pinsage [46] is one of the first industrial solution to employ graph convolutional operators on item-item graphs for Pinterest recommendation. It combines efficient random walks with graph convolution but the collaborative filtering (CF) signals are captured only for item relations, not user historical behaviors. SpectralCF [47] proposes a spectral graph

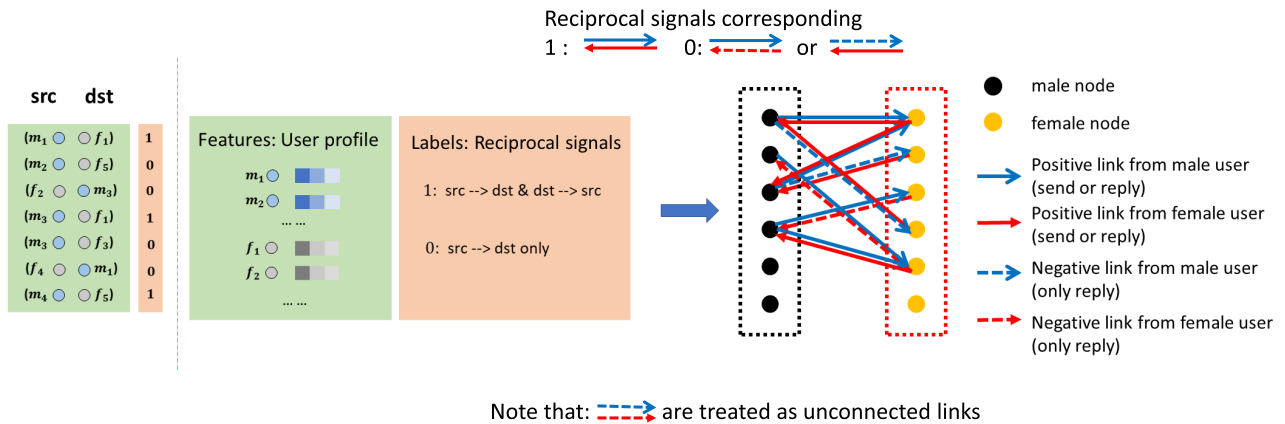


FIGURE 3. The dataset illustration (left) and its reformulation in graph conceptual view (right). As we introduce above, the dataset consist of users interactions based on source (src) side and destination (dst) side. The label is binary which we called reciprocal signal to represent the interaction status of sending like or not to each other. When both sides send “like”, the reciprocal signal would be set to 1; and when only src side initials the interaction (i.e send “like” to dst side) but gets no reply, it would be 0. Therefore, according to such formulation in the service, we can split the binary reciprocal signal into four user behaviors shown in the right and treat the problem as a bipartite graph link prediction.

convolution operator to excavate all possible link between nodes in spectral domain. However, the eigen-decomposition causes high computational complexity. Therefore, the method is quite time-consuming and not able to support applications on real-world large scale datasets.

C. NEGATIVE SAMPLE MINING

Negative sampling is a well-known technique in the word2vec method in the NLP community [48], [49]. Word2vec can also be seen as a recall problem, where the context word is recalled from the center word throughout the lexicon. Embedding-based Retrieval (EBR) [16] formulates the search retrieval task as a recall optimization problem. They claim that you cannot (only) take non-clicked impressions (those impressed but not clicked results) as negative samples. They propose the idea of easy negative/hard negative sample grading and they also propose to enhance the idea of hard negative. Airbnb recommendation [17] and Mobious [18] share the same perspective on negative sample diversity with EBR. The difference is that Airbnb recommendation selects hard negative samples according to their business logic. They employ rooms ‘in the same city with the positive samples but not accepted’ as negative samples to enhance the similarity between positive and negative samples in terms of geography and apply rooms ‘rejected by the owner’ as negative samples to enhance the similarity of positive and negative samples in terms of users’ interest matching, which increases the learning difficulty of the model. On the other hand, when it comes to the situation that the business logic is not so obviously signaled, it is up to the model to excavate proper negative samples. Both EBR and Mobious use the previous version of the recall model to filter out the less similar pairs as additional negative samples to train the next version of the model.

III. METHODOLOGY

In this section, we present our graph fusion reciprocal recommender (GFRR) prediction models.

A. DATASET AND GRAPH CONSTRUCTION

The dataset we used for our experimental evaluation contained only interactions between senders and receivers. This is consistent with previous works on RRS [6], [7], [8], [9].

However, previous studies generally only focus on response prediction, predicting only the final status label. In real applications, both user preference directions must be considered, which means the send behavior should be predicted as well. If the send like behavior is also predicted as positive, the whole reciprocal signal can be treated as positive. Therefore, we decompose the final status labels which are considered as reciprocal signals into several kinds of link/edge in graph formulation. The reciprocal recommendation task is also transferred into both send prediction and reply prediction simultaneously.

The dataset illustration is visualised in Figure 3. As we propose a graph neural network based method, we also reformulate the data in a bipartite graph conceptual view. As mentioned previously, we distinguish the send and reply interaction as different types of edges in the bipartite graph; both send and reply edges can be further classified as positive or negative. We formulate the link prediction task on the bipartite graph as the reciprocal recommendation. In our online dating case, the bipartite graph consist of two domains with male nodes and female nodes because the service our collaborating corporation provides does not consider the situation of homosexual and bisexual persons. In other cases, the graph structure is optional without doubt. Even in our data construction, the interactions between the same gender nodes can be used to train the model (though there is no such interaction).

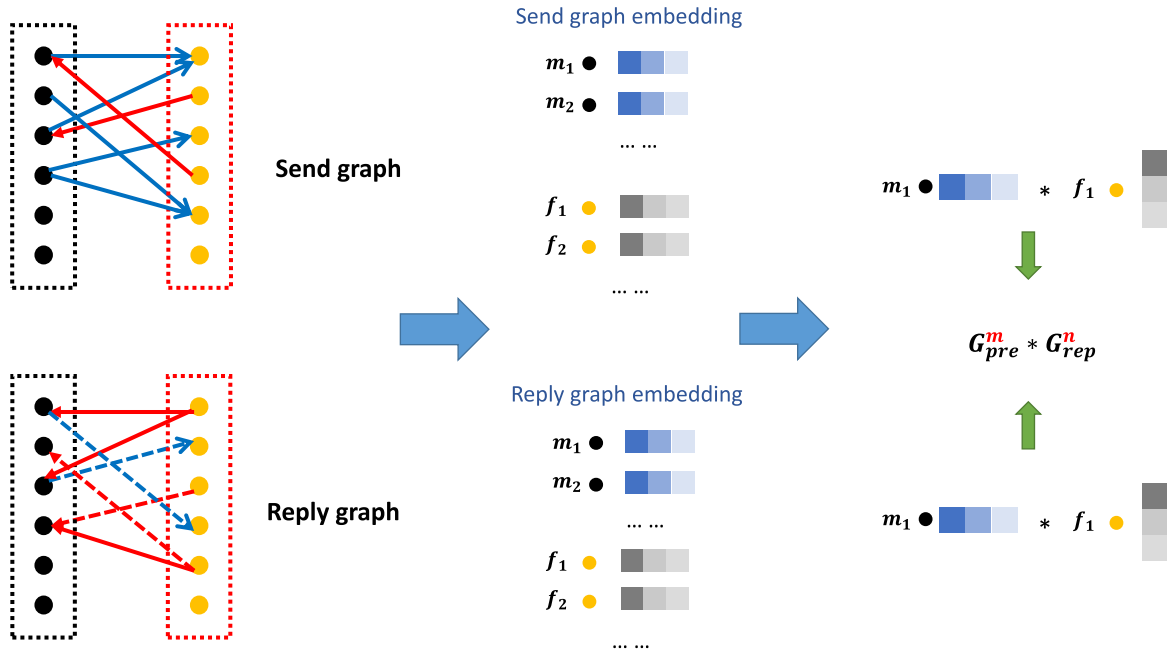


FIGURE 4. The procedure of our reciprocal recommender by conducting send/reply prediction simultaneously and finally fusion operation. The send graph and reply graph are sampled from the original bipartite graph according to the edge type. Both them contain all user nodes same with original one. The user node embeddings gain from our graph model represent different semantic information (send or reply behavior). We apply cosine similarity to get the final prediction results. Here, we conduct the fusion operation by setting the product of the send and reply prediction results with additional exponential parameters as the reciprocal result. More details are provided in Section IV.

TABLE 1. Graph construction notation.

Notation	Description
G	A bipartite pair matching network
V	The set of user nodes
E^{src}	The set of send links
E^{dst}	The set of reply links
E^{non}	The set of unconnected links

Table 1 shows the notation used in the graph construction. Consider a social network $G = (V, E^{src}, E^{dst})$, where V is the set of users with their own attributes such as age, height, and income, and E^{src}, E^{dst} are the sets of edges of send edges and reply edges, respectively. Consider a pair of user nodes such that both parties show interest in each other, i.e., for which the elements of both E^{src} and E^{dst} joining the nodes are positive: such a pair is consider to be positive linked. Unconnected node pairs are defined as null relationships, denoted as E^{non} . Note that, although we use dotted line to represent negative reply links, such links are treated as null relationships, the same with unconnected node pairs, in model training. The pair matching prediction problem to be addressed is predicting whether the node pairs in E^{non} will be connected by a positive edge.

B. SEND/REPLY GRAPH AND FUSION OPERATION

To distinguish the two parts of reciprocal recommendation (send and reply), two graphs are constructed: one containing only send edges and the other only reply edges. We divide

the original reciprocal signals that represent the final status of pair matching into two graphs, each containing all user nodes in the dataset. Because of the data sparsity in the real application, both graphs contain numerous null relationship node pairs; this problem is more critical in the reply graph.

Figure 4 shows the procedure of our reciprocal recommender. Send and reply results are predicted simultaneously; then, the fusion operation is used to predict the reciprocal result as well.

C. GRAPH CONVOLUTION

In this work, we employ the graph convolution to learn both side information of user nodes and the message passing flow of user historical behaviors. Figure 5 shows the flowchart of the graph convolution in our reciprocal recommender. Take the send graph as example, we first treat the bipartite graph in a normal conceptual view of graph. GCNs are demonstrated as successful methods to learn the first-order and high-order propagation between nodes in the graph. Considering the computational inefficiency of spectral GCN models [33], [47], we employ the graph convolutional kernel proposed in [14], which can be expressed as:

$$h_v^{(k)} = \sigma \left(W_k \sum_{u \in \mathcal{N}_v} \frac{h_u^{(k-1)}}{|\mathcal{N}_v|} + B_k h_v^{(k-1)} \right), \quad (1)$$

where $h_v^{(k)}$ denotes the node embedding of v after the k -layer graph convolution, σ denotes non-linearity activation

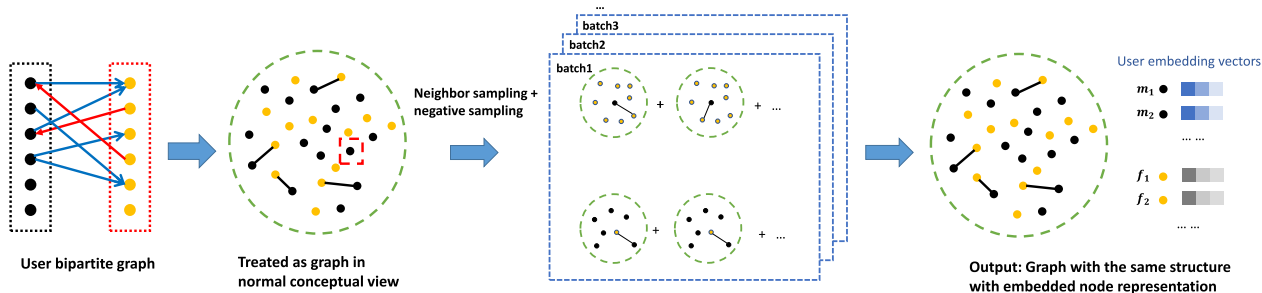


FIGURE 5. The flowchart of graph convolution in our proposed method.

function, W_k denotes the parameter matrix of neural network, the middle part of $\sum_{u \in \mathcal{N}_v} \frac{h_u^{(k-1)}}{|\mathcal{N}_v|}$ denotes the aggregation function to learn the representation of target node v 's neighbor nodes, and the part of $B_k h_v^{(k-1)}$ can be considered as a traditional multilayer perceptron.

D. OPTIMIZATION

Training a link prediction model involves comparing the scores between nodes connected by an edge against the scores between an arbitrary pair of nodes, which is known as negative sampling in embedding methods. For example, given an edge connecting u and v , we encourage the score between the nodes u and v to be higher than the score between node u and a sampled node v' from an arbitrary noise distribution $P_n(v)$. Such negative sampling is used in our work. When training the graph convolutional based models, we employ the noise-contrastive estimation loss which can be expressed as:

$$L = -\log \left(\sigma \left(z_u^T z_v \right) \right) - Q E_{v_n \sim P_n(v)} \left(\log \left(\sigma \left(-z_u^T z_{v'} \right) \right) \right), \tag{2}$$

where the first term in Equation (2) is to maximize the probability of target node u and sampled positive node v , while the second term tries to iterate over negative sampled nodes v' and minimize their probability with target node u from an arbitrary noise distribution. The Q here is a hyper-parameter to decide the sampling size, i.e., how many negative sampled nodes in one subgraph sampling. The discussion of influence of this parameter is in Section IV-D.

IV. EXPERIMENTS

A. EVALUATION PROTOCOLS

The dataset we use is introduced in Section III-A. We selected the interaction behavior data including like sending and the reply towards them occurred in 2020. As our dataset have the problem of label imbalance, the mean matching rate (label = 1) is lower than 10%. We employed two commonly used evaluation metrics in prediction task: area under ROC (AUC) and average precision (AP). The higher the scores are, the better they are.

B. HYPER-PARAMETER SETTINGS

Similar with many graph-based recommendation models such as NGCF [44] and LightGCN [50], the the embedding size is fixed to 64 for our GFRR model. The number of hidden layer is test in the range of 1 to 3 and the satisfactory performance can be achieved when the number equals to 1. We optimize the model with Adam [51] and use the default learning rate of 0.001. Typically, 300 epochs are sufficient for the models to converge from our experimental results.

C. COMPARISON TO PRIOR METHODS

We compared our proposed GFRR with the following previous works:

- **FM** [10]: The method is a traditional feature interaction based method that apply latent vector for every feature to compute the interaction weights easily.
- **NFM** [22]: The method is one of the FM based neural network method that can learn both low and high-order feature interactions via a wide part (logistic regression) and a deep part (FM embedding based MLP).
- **DeepFM** [21]: The method is another FM based neural network method. It can also learn low and high-order feature interactions. The difference is that the wide part of DeepFM results and stack layer in the deep part.
- **NIFM** [9]: Our previous work. The variant of NFM with a novel design of stack method before feeding embedding into MLP.
- **LFRR** [7]: The method is a latent factor based method for reciprocal recommender systems.

Table 2 shows the best performance results in terms of AUC and AP. We further plot the training curves of testing AUC and AP in Figure 6 and Figure 7 respectively to reveal the advantages of GFRR and to be clear of the training process. It can be observed that our proposed method, GFRR, outperforms the other models by over 3.20% point absolute improvement in term of AUC and over 2.79% point absolute improvement in term of AP on send prediction; by over 1.74% point absolute improvement in term of AUC and over 0.70% point absolute improvement in term of AP on reply prediction; by over 4.35% point absolute improvement in term of AUC and by over 0.30% point absolute improvement

TABLE 2. Comparison of proposed GFRR with various methods based on feature interaction and latent factor.

	Send Prediction		Reply Prediction		Reciprocal Prediction	
	AUC \uparrow	AP \uparrow	AUC \uparrow	AP \uparrow	AUC \uparrow	AP \uparrow
FM [10]	0.6722	0.2094	0.6548	0.1904	0.6608	0.1965
NFM [22]	0.6834	0.2305	0.6701	0.2181	0.6691	0.2221
DeepFM [21]	0.6602	0.2194	0.6553	0.2165	0.6601	0.2159
NIFM [9]	0.6995	0.2322	0.6721	0.2232	0.6698	0.2365
LFRR [7]	0.6058	0.1356	0.5437	0.0989	0.5569	0.1068
GFRR	0.7315	0.2601	0.6895	0.2302	0.7126	0.2425

TABLE 3. Comparison of models' performance on different negative sample mining guidelines. Only easy negative means randomly sample negative training data from all unlinked node pairs (no interaction user pairs). Easy + Hard means training data contain hard negative. As in our formulation of reciprocal recommendation, hard negative samples do not exist in reply prediction. Therefore there is no comparison experiment results on it.

	Send Prediction				Reply Prediction		Reciprocal Prediction			
	AUC \uparrow		AP \uparrow		AUC \uparrow	AP \uparrow	AUC \uparrow		AP \uparrow	
	Only Easy	Easy + Hard	Only Easy	Easy + Hard	Only Easy	Only Easy	Easy + Hard	Only easy	Easy + Hard	
FM [10]	0.6689	0.6722	0.2021	0.2094	0.6548	0.1904	0.6591	0.6608	0.1928	0.1965
NFM [22]	0.6791	0.6834	0.2269	0.2305	0.6701	0.2181	0.6614	0.6691	0.2213	0.2221
DeepFM [21]	0.6671	0.6498	0.2194	0.2182	0.6553	0.2165	0.6601	0.6593	0.2199	0.2175
NIFM [9]	0.6889	0.6995	0.2301	0.2322	0.6721	0.2232	0.6689	0.6698	0.2291	0.2365
GFRR	0.7103	0.7315	0.2568	0.2601	0.6895	0.2302	0.7094	0.7126	0.2329	0.2425

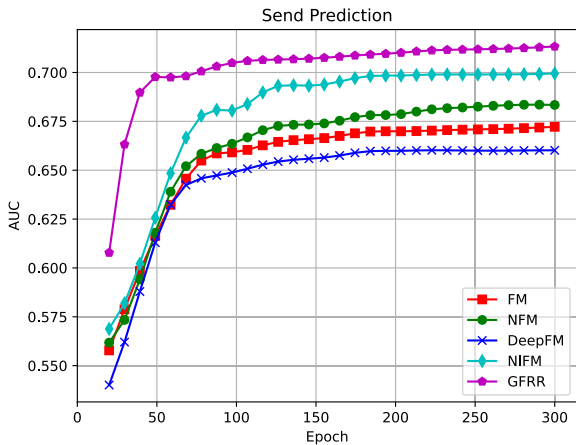


FIGURE 6. Training curves of GFRR and comparison models on send prediction, which are evaluated by testing AUC per 10 epochs.

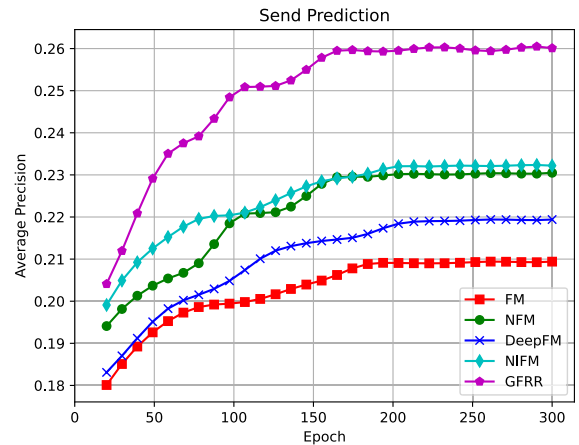


FIGURE 7. Training curves of GFRR and comparison models on send prediction, which are evaluated by testing AP per 10 epochs.

in term of AP on fusion, i.e., reciprocal prediction. Therefore, the learning of structural information of user historical interactions contains rich implicit feedback on their preference towards others. The graph convolution based method is not only able to learn users' profile feature but also the implicit feedback from their historical interactions. LFRR [7] does not perform well in this task. A possible reason is that in their work they made several limitations on the data they selected for their experiments: they only selected active users to avoid a cold start problem. However, the data we use are randomly sampled from corporation's database and consistent with the data distribution of the actual business. Therefore, LFRR is

not involved in the next discussion about the negative sample mining.

The computation complexity of mentioned FM-based models depends on the number of dimensions of original input per node (user). They need to train second-order interaction vectors; therefore larger the user's feature dimensions is, higher computation cost is. In our used dataset, the number of user's personal feature dimensions (after one-hot encoding) is over 500. Empirically, under the same experimental setting of negative sample mining (as explained in Section IV-D), GFRR and NFM (the most efficient FM-based DNN method)

cost around 15s and 25s per epoch on the used data for training.

D. NEGATIVE SAMPLE MINING

Negative sample mining shows huge potential on information and search retrieval [16], [17], [18]. We are interested in how such negative sample mining works in our reciprocal recommendation task. The easy/hard negative grading in a typical item recommendation task is that: easy negative samples refer to those unseen items/products for a single user and hard negative samples refer to those items/products which had been recommended by service (seen by the user) but the user did not click/purchase. They are all treated as negative samples but have different semantic information. Therefore, it is meaningful to let the model learn how to distinguish them from each other.

In our reciprocal recommendation formulation, easy negative samples do not show significant difference and mean two users have not interacted yet. Hard negative samples mean one user (source side) have seen the other (destination side) user's profile but do not send "like." Definitely, we do have our reciprocal signals, i.e. those user pairs that src side user send "like" but dst user do not reply. They are strong negative signals but do not contain semantic information for send prediction. On the other hand, hard negative samples do not exist in reply prediction. Therefore, there is no comparison experiment results on it here.¹

Table 3 shows that the introducing hard negative mining enhances the send prediction results. The final reciprocal prediction performance is also improved with it.

E. NEGATIVE SAMPLING SIZE

We believe that negative sample diversity is important to our task. We are also interested in how much the negative sampling size has an impact to the prediction results. It is not realistic that we train a graph model directly on the industrial level data. As we introduced in Section III-D, we construct a subgraph for one node pair, i.e. one target node with one positive node and several negative nodes as a subgraph. It should be mentioned that FM-based methods do not train as a graph but pairwise feature interaction based ones. Therefore, the negative sampling comparison experiments here are conducted by adjusting the proportion of positive and negative in their whole training samples.

Table 4 and Figure 8 shows the results of changing negative sample size Q . We initialize this parameter as 10 because the origin proportion of the positive pair samples and the negative ones is about 1:10. It is shown that $Q = 4$ is slightly better than the other conditions. From the results, the sample size is set to 4 in our other tuning.

¹We actually conducted such experiments. Although there is no meaning of hard negative samples in reply task, it can be established in practice because send graph and reply graph share the same graph structure, i.e., same user nodes but different edges. However there is no obvious enhancement.

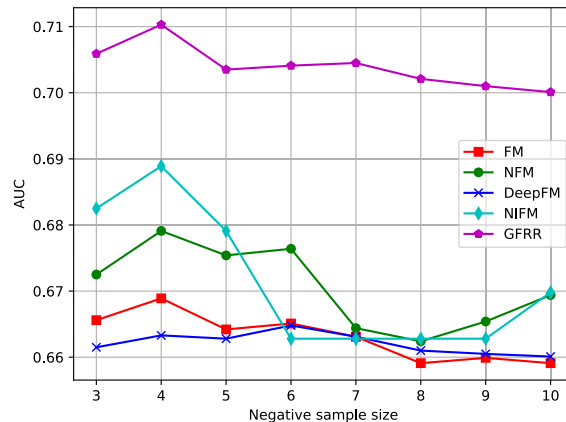


FIGURE 8. Results of GFRR and comparison models at different negative sample size Q . Most of the models share the same trend that satisfactory performance can be achieved when Q equals to 4.

TABLE 4. Influence of negative sampling size Q . The experiment is for send prediction and operated with only easy negative samples.

	$Q = 3$	$Q = 4$	$Q = 5$	$Q = 10$
FM [10]	0.6656	0.6689	0.6642	0.6591
NFM [22]	0.6725	0.6791	0.6754	0.6694
DeepFM [21]	0.6615	0.6633	0.6628	0.6601
NIFM [9]	0.6825	0.6889	0.6791	0.6698
GFRR	0.7059	0.7103	0.7035	0.7001

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced a graph convolutional network based method to achieve reciprocal prediction. We reformulated the prediction the task as predicting send and reply signals simultaneously rather than reciprocal signals only. We enhanced the sample diversity and the realism and reliability of the model through negative sample mining to keep the offline training data and actual online distributions consistent. Our proposed method outperformed earlier models because it captured both side information from user profiles and structural information from the user historical network. The negative sample mining also showed its effectiveness for reciprocal recommendation.

One possible direction of future development is exploiting knowledge graphs to enrich the user embedding results with more information: rather than just the user's profile and historical behavior, additional data about interest features and social networks can be included. Work on echo chamber effects [52], [53], [54] should also be conducted to improve the users' experience by avoiding tedious similar recommendation.

REFERENCES

- [1] L. Pizzato, T. Rej, J. Akehurst, I. Koprinska, K. Yacef, and J. Kay, "Recommending people to people: The nature of reciprocal recommenders with a case study in online dating," *User Model. User-Adapted Interact.*, vol. 23, no. 5, pp. 447–488, 2013.
- [2] A. Kleinerman, A. Rosenfeld, F. Ricci, and S. Kraus, "Optimally balancing receiver and recommended users' importance in reciprocal recommender systems," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 131–139.

- [3] Z. Siting, H. Wenxing, Z. Ning, and Y. Fan, "Job recommender systems: A survey," in *Proc. 7th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Jul. 2012, pp. 920–924.
- [4] W. Hong, S. Zheng, H. Wang, and J. Shi, "A job recommender system based on user clustering," *J. Comput.*, vol. 8, no. 8, pp. 1960–1967, Aug. 2013.
- [5] J. He and W. W. Chu, "A social network-based recommender system (SNRS)," in *Data Mining for Social Network Data*. Springer, pp. 47–74, 2010.
- [6] P. Xia, B. Liu, Y. Sun, and C. Chen, "Reciprocal recommendation system for online dating," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2015, pp. 234–241.
- [7] J. Neve and I. Palomares, "Latent factor models and aggregation operators for collaborative filtering in reciprocal recommender systems," in *Proc. 13th ACM Conf. Recommender Syst.*, Sep. 2019, pp. 219–227.
- [8] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay, "RECON: A reciprocal recommender for online dating," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 207–214.
- [9] L. Zhang, X. Wang, and T. Yamasaki, "Deep feature interaction embedding for pair matching prediction," in *Proc. ACM Multimedia Asia*, Dec. 2019, pp. 1–6.
- [10] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 995–1000.
- [11] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
- [12] B. Mitra and N. Craswell, *An Introduction to Neural Information Retrieval*. Now Foundations and Trends, 2018.
- [13] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 950–958.
- [14] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2017, [arXiv:1706.02216](https://arxiv.org/abs/1706.02216).
- [15] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-I. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5453–5462.
- [16] J.-T. Huang, A. Sharma, S. Sun, L. Xia, D. Zhang, P. Pronin, J. Padmanabhan, G. Ottaviano, and L. Yang, "Embedding-based retrieval in Facebook search," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2553–2561.
- [17] M. Grbovic and H. Cheng, "Real-time personalization using embeddings for search ranking at airbnb," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 311–320.
- [18] M. Fan, J. Guo, S. Zhu, S. Miao, M. Sun, and P. Li, "MOBIUS: Towards the next generation of query-ad matching in Baidu's sponsored search," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2509–2517.
- [19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, Sep. 2015.
- [20] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, and H. Shah, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.
- [21] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," 2017, [arXiv:1703.04247](https://arxiv.org/abs/1703.04247).
- [22] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 355–364.
- [23] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," 2017, [arXiv:1708.04617](https://arxiv.org/abs/1708.04617).
- [24] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data," in *Proc. Eur. Conf. Inf. Retr.* Springer, 2016, pp. 45–57.
- [25] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1149–1154.
- [26] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, Feb. 2010, pp. 81–90.
- [27] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and testing low-degree polynomial data mappings via linear SVM," *J. Mach. Learn. Res.*, vol. 11, pp. 1471–1490, Apr. 2010.
- [28] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich, "Web-scale Bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Jun. 2010, pp. 13–20.
- [29] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 111–112.
- [30] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1235–1244.
- [31] J. Chen, B. Sun, H. Li, H. Lu, and X.-S. Hua, "Deep CTR prediction in display advertising," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 811–820.
- [32] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for Youtube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 191–198.
- [33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, [arXiv:1609.02907](https://arxiv.org/abs/1609.02907).
- [34] Y. Ding, X. Zhao, Z. Zhang, W. Cai, N. Yang, and Y. Zhan, "Semi-supervised locality preserving dense graph neural network with ARMA filters and context-aware learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2021.
- [35] Y. Ding, X. Zhao, Z. Zhang, W. Cai, and N. Yang, "Graph sample and aggregate-attention network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2021, doi: [10.1109/LGRS.2021.3062944](https://doi.org/10.1109/LGRS.2021.3062944).
- [36] Y. Ding, X. Zhao, Z. Zhang, W. Cai, and N. Yang, "Multiscale graph sample and aggregate network with context-aware learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4561–4572, 2021.
- [37] D. Yao, Z. Zhi-li, Z. Xiao-feng, C. Wei, H. Fang, C. Yao-ming, and W.-W. Cai, "Deep hybrid: Multi-graph neural network collaboration for hyperspectral image classification," *Defence Technol.*, Feb. 2022.
- [38] Y. Ding, Z. Zhang, X. Zhao, D. Hong, W. Li, W. Cai, and W. Cai, "AF2GNN: Graph convolution with adaptive filters and aggregator fusion for hyperspectral image classification," *Inf. Sci.*, vol. 602, pp. 201–219, Jul. 2022.
- [39] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
- [40] J. Hu, Q. Fang, S. Qian, and C. Xu, "Multi-modal attentive graph pooling model for community question answer matching," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 3505–3513.
- [41] M. Gori, A. Pucci, V. Roma, and I. Siena, "ItemRank: A random-walk based scoring algorithm for recommender engines," in *Proc. IJCAI*, vol. 7, 2007, pp. 2766–2771.
- [42] F. Christoffel, B. Paudel, C. Newell, and A. Bernstein, "Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks," in *Proc. 9th ACM Conf. Recommender Syst.*, Sep. 2015, pp. 163–170.
- [43] J.-H. Yang, C.-M. Chen, C.-J. Wang, and M.-F. Tsai, "HOP-rec: High-order proximity for implicit recommendation," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 140–144.
- [44] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 165–174.
- [45] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, [arXiv:1706.02263](https://arxiv.org/abs/1706.02263).
- [46] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.
- [47] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 311–319.
- [48] K. W. Church, "Word2Vec," *Nat. Lang. Eng.*, vol. 23, no. 1, pp. 155–162, 2017.
- [49] Y. Goldberg and O. Levy, "Word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," 2014, [arXiv:1402.3722](https://arxiv.org/abs/1402.3722).
- [50] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 639–648.

- [51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [52] D. Sacharidis, "Diversity and novelty in social-based collaborative filtering," in *Proc. 27th ACM Conf. User Model., Adaptation Personalization*, Jun. 2019, pp. 139–143.
- [53] A. Tommasel, J. M. Rodriguez, and D. Godoy, "I want to break free! Recommending friends from outside the echo chamber," in *Proc. 15th ACM Conf. Recommender Syst.*, Sep. 2021, pp. 23–33.
- [54] T. Donkers and J. Ziegler, "The dual echo chamber: Modeling social media polarization for interventional recommending," in *Proc. 15th ACM Conf. Recommender Syst.*, Sep. 2021, pp. 12–22.



XUETING WANG (Member, IEEE) received the M.S. and Ph.D. degrees in information science from Nagoya University, Japan, in 2015 and 2018, respectively. She was a Project Assistant Professor at the Department of Information and Communication Engineering, The University of Tokyo, Japan. She is currently working with the AI Laboratory, CyberAgent, Inc. Her research interests include multimedia processing, social media analysis, and recommendation. She has served as the Primary Organizer of MMAsia2019 Tutorial on "Multimedia Attractiveness Computing." She received the Best Paper Award from ISM2014. She was the Local Chair of MIPR2021 and MMAsia2022.



LUWEI ZHANG (Graduate Student Member, IEEE) received the M.S. degree from The University of Tokyo, Japan, where he is currently pursuing the Ph.D. degree. His interests include recommender systems and graph convolution networks. He is a Student Member of ACM.



TOSHIHIKO YAMASAKI (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from The University of Tokyo. He is currently an Associate Professor with the Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo. He was a JSPS Fellow for Research Abroad and a Visiting Scientist at Cornell University, from February 2011 to February 2013. His current research interests include attractiveness computing based on multimedia big data analysis and fundamental problems in pattern recognition and machine learning. He is a member of ACM, AAAI, IEICE, IPSJ, and ITE.

• • •