World Wide Web (2025) 28:33
https://doi.org/10.1007/s11280-025-01345-z

Multi-role social behavior modeling in content-free news recommendation

Yihong Zhang[1] · **Takahiro Hara**[1]

Received: 15 October 2024 / Revised: 15 April 2025 / Accepted: 16 April 2025 /
Published online: 5 May 2025

Abstract
Recommendation systems nowadays are deployed everywhere to solve the information overload problem. Thousands of news items that appear on online news platforms also need recommendations. However, news recommendation differs from product recommendation because of its focus on public interest. In this paper, we propose a multi-role social behavior model that includes a chaser model, which describes behavior patterns of chasing popular news, and a sider model, which describes behavior patterns of seeking news from a similar ideology standpoint. We propose a framework that integrates this model into a graph-based recommendation system. Unlike other intent disentangling techniques, our model explicit models social behavior patterns originated from public interest. We test our framework with two real-world news recommendation datasets. Compared to state-of-the-art baseline news recommendation models, our method achieves significantly higher recommendation accuracy. By analyzing the model outputs, we also gain a better understanding of news-seeking social behavior.

Keywords News recommendation · Social recommendation · User behavior modeling

1 Introduction

Recommendation systems are designed to solve the information overload problem by providing personalized item recommendations. In recent years, recommendation systems have been used to recommend e-commerce products [1], online videos [2], location-based services [3], and social network friends [4]. Typically, there are thousands or even millions of candidate items, and the task is to find what could be interesting for millions of users. News is one major type of information that nowadays faces the information overload problem, as thousands of news items appear in online news platforms each day. Therefore, there is a need to design recommendation systems for news [5].

The mainstream recommendation systems nowadays are mostly based on collaborative filtering, in which a user's preference is learned by comparing this user with similar users [6]. Matrix factorization is a neat solution in collaborative filtering. Given a partially

✉ Yihong Zhang
zhang.yihong@ist.osaka-u.ac.jp

1 Graduate School of Information Science and Technology, The University of Osaka, 567-0074 Suita, Japan

filled user-item interaction matrix, matrix factorization can fill the empty cells using the dot-product of user and item embedding, revealing the latent preferences [7]. This line of techniques focuses on learning the distributed user and item representations, or so-called embeddings. Recently, graph neural network techniques have gained wide attention because of their effectiveness and efficiency in providing informative embeddings [8]. Graph-based recommendation systems use users and items as nodes, and the interaction as edges. Then a common technique called graph convolution [9] can be deployed to aggregate information of multi-hop neighbors to each node. It is shown that by using the graph convolution technique, the learned embedding can outperform simple matrix factorization in recommendation accuracy [10, 11].

The existing recommendation techniques can be applied directly to news recommendations, as long as we can model it as a standard recommendation problem. There are three types of recommendation problems, collaborative filtering, content-based recommendation, and hybrid recommendation that has features from the last two problems [12]. Collaborative filtering learns user preference purely based on user-item interaction, while content-based recommendation takes into account content information that is associated with users or items, such as user age, gender, or item textual descriptions. Because news is inherently associated with text content, content-based news recommendation has been the mainstream approach [13]. However, with the emergence of graph neural networks, collaborative filtering methods that consider user-item interaction graphs as the main information source have quickly matured in [14]. In this work, we follow the collaborative filtering problem setting, or content-free recommendation, because it has a wider applicability. Especially in news recommendation, the content-free method is more resilient to noises and biases in the news contents [5].

On the other hand, news recommendation is different from e-commerce product recommendation in user behavior patterns. In e-commerce platforms, a major motivation for a user to purchase a product is their personal needs. The choice of which brand or style of clothes to purchase, for example, is a very personal choice. However, this is not so in news-seeking behavior. News-seeking behavior is mostly out of public interest. A user clicks a news mostly because they are interested in what is going on in the society. Therefore public interest has a much stronger influence on news-seeking behavior. As such, a recommendation system that takes into account the public interest can be more suitable for news recommendation.

In this work, we propose a multi-role social behavior model for news recommendation. Our intuition is that there should be two types of information-seeking patterns led by public interest. First, the *Chaser* pattern, which describes the behavior of a user given popular news. They may be attracted strongly to popular news or ignore them. Second, the *Sider* pattern, which describes the behavior of a user given a neighborhood of users that have similar ideology standpoint as theirs. They may side with their ideology neighbors closely or ignore them. The sider pattern has a particularly strong influence on news viewing behavior and is the cause of the so-called *echo chamber* phenomenon [15].

Figure 1 illustrates how these two patterns appear in the interaction graph when they have a positive impact on the user. The positive chaser pattern is characterized by the high degrees of its neighbor item nodes. The positive sider pattern is characterized by a high item interaction similarity of the user and some other users. We note that these two patterns are not mutually exclusive, as one can chase popular news at the same time as siding with a certain ideology neighborhood. Therefore a multi-role behavior model that combines these two patterns is more appropriate.

As a result, we propose a method called Chaser-Sider Multi-role Embedding Learning (CS-MREL) for graph-based news recommendation. We find that the joint multi-grained

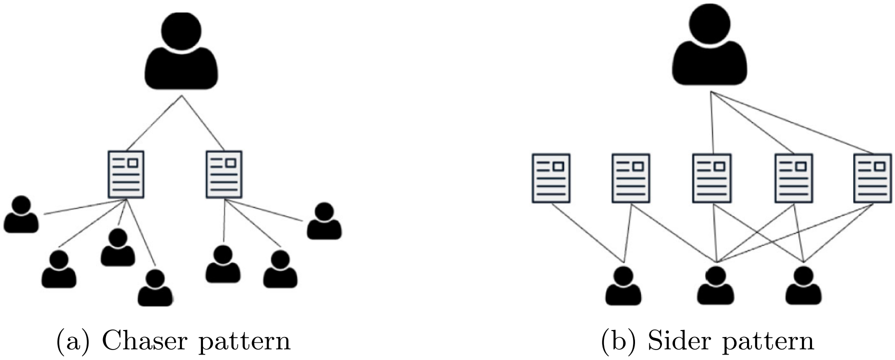


Figure 1 Two social behavior patterns

popularity-aware graph convolution collaborative filtering (JMP-GCF) [16] can be used to model the chaser behavior by enabling the popularity bias. On the other hand, a user-user ideology neighborhood graph constructed from interaction data can be used to model the sider behavior. The required information, including item popularity and user ideology neighborhood, can be extracted from the interaction graph, so our model requires no additional data. Our main contributions are summarized as follows:

- We identify two social behavior patterns, which are called chaser and sider, respectively. We find graph patterns that are associated with these behaviors, which allows us to integrate the model into graph-based recommendation systems.
- We propose a Chaser-Sider Multi-role Embedding Learning (CS-MREL) model that integrates neatly with existing graph-based recommendation systems. The model is content-free and requires no external information except the user-item interaction data.
- We test our method in two real-world datasets, comparing it with several state-of-the-art baseline news recommendation methods. The experimental results clearly show the advantage of our method. We also test our method on non-news datasets for gaining more insights.

The remainder of this paper is organized as follows. In Section 2, we will discuss related work. In Section 3, we will formulate the problem of graph-based news recommendation. Then in Section 4, we will present our method. We conduct a preliminary analysis on news and non-news datasets to verify our assumption in Section 5. Section 6 will present our experimental evaluation on recommendation performance. Finally, Section 7 will conclude this paper.

2 Related work

We follow closely the recent research trend of interaction graph-based news recommendation, which supports content-free recommendation. Qian et al. proposed an early method called IGNN that used graph convolution on user-item interactions to obtain user embeddings [17]. For item embeddings, they relied on graph embedding techniques such as TransE and these embedding are not learned in graph convolution. They method required item content information which cannot be separated from the graph convolution. Hu et al. proposed a framework that models short-term and long-term interest for news recommendation [18]. The short-term interest is extracted from recent data, while the long-term interest is based on the whole interaction data. Apart from adding news-tag relationship into the graph, they have not improved the graph convolution mechanism itself. Qiu et al. proposed a method called GREP that deals with the existing and potential interest in news recommendation [19]. They combined the interaction graph and a knowledge graph constructed from entities extracted from news contents. The entities are critical to the method and cannot be separated from the interaction graph. Ge et al. proposed a method called GERL for graph news recommendation [14]. They improved the recommendation architecture by adding a transformer and an attention block for selecting more important information. Their method learns user embedding based on news content but can be adapted for content-free recommendation. Hu et al. conducted another study that is very similar to ours, which proposed a graph news recommendation method with preference disentanglement called GNUD [20]. The key idea is to use a latent variable model to select subsets of news items for a user, from all news items they clicked. Although their method was shown to achieve better recommendation performance, we find that the computational overhead of the added component is large. Although these works have already begun to explore interest disentanglement in graph news recommendation, they have not model different user behavior patterns explicitly and thus are not optimal solutions.

Although not directly designed for news recommendation, several graph-based recommendation systems with interest disentanglement are very relevant to our task. We distinguish works that deal with explicit interest patterns from that deal with latent interest patterns [21], which is less relevant to our work. The most studied explicit interest patterns are long and short-term interests, and conformity, and personal interests. Long and short-term interests are mostly studied for sequential prediction, in which a next item is recommended to users based on their past interaction [22, 23]. The conformity and personal interest disentanglement are more relevant to our tasks because news recommendation involves public interest. For example, Zheng et al. proposed a method that distinguished two parts of user embedding, one part for conformity, and the other for personal interest [24]. Their method relies on training-time item popularity inference, which compares the popularity of the positive and negative training instance pairs, and thus has smaller applicability. More recently, Liu et al. proposed a method called JMP-GCF that also models user conformity [16]. Their method deploys graph convolution, and thus tends to learn high-order structural information and has a wider applicability. However, their model only describes different levels of user conformity and does not consider other possible patterns. Zhang et al. proposed a social recommendation method called GL-HGNN, that is based on user social relationships and graph convolution [25]. By considering social relationships, the method can distinguish item interaction as private or social behavior. Their method requires explicit social relationship data but can be adopted in our task by constructing user ideology neighborhoods from interaction data. We argue that these methods have not considered the combined behavior of different social interests, such as chasing popular items and siding with users with similar tastes. Our method can potentially achieve better performance by adding the behavior patterns together. For the purpose of verification, we will compare our method with GERL, GNUD, JMP-GCF and GL-HGNN in our experimental evaluation.

3 Problem definition

We follow the standard recommendation problem setting, in which we have sets of users $\mathcal{U}$ and news items $\mathcal{V}$. The goal of a recommendation model is to produce a preference score given a pair of user and item:

$$\hat{y}_{uv} = f(u, v) \tag{1}$$

where $u \in \mathcal{U}$, $v \in \mathcal{V}$ and $f(.)$ is the recommendation model. The key to learn the recommendation model is through historical interaction data, which can be user clicking the news item, or user retweeting the news item.

Graph-based recommendation systems [26] model the interaction data neatly. Given a set of interaction data, we have $\mathcal{G} = \{\mathcal{U} \cup \mathcal{V}, \mathcal{E}\}$, where an edge $e_{ij} \in \mathcal{E}$ connects user i and item j. When working with graph-based models, the set of edges $\mathcal{E}$ is often transformed into an adjacency matrix $\mathbf{A}$, where $\mathbf{A}_{ij} \in \{1, 0\}$ indicates whether or not there is an interaction between user i and item j.

In many previous works on news recommendation, the news item v is often associated with some external information, such as titles or entities [27]. In this work, we follow the content-free approach, which does not assume such external information. Like these works, we associate user u and item v with feature vectors $\mathbf{u}$ and $\mathbf{v}$. However, unlike previous works, our feature vectors are not initialized with external information. They are randomly initialized and can be learned only through the interaction data. By considering only the interaction data, we increase the applicability of our method while also avoiding potential biases that could be brought in by external information [28].

4 Methodology

The point of our method is to model chaser and sider behavior separately, and then combine them together to obtain a multi-role embedding, which is then utilized in user-item preference prediction. Figure 2 shows an overview of our model. In this section, we will discuss the details of our modeling of chaser and sider behavior, and how we combine them together into a unified model.

4.1 Combined Behavior Model

Let us first consider the basic recommendation model set as (1). Suppose the model makes recommendation on vector representations of u and v, denoted as $\mathbf{u}$ and $\mathbf{v}$. The most common recommendation model is the matrix factorization model, which fills the interaction matrix using the dot product:

$$f(u, v) = \mathbf{u} \cdot \mathbf{v}. \tag{2}$$

The vectors $\mathbf{u}$ and $\mathbf{v}$ contain information regarding the user and item behavior. Now suppose we have two sets of behavior models, recorded in separate vectors ($\mathbf{u}^1$, $\mathbf{v}^1$) and ($\mathbf{u}^2$ and $\mathbf{v}^2$). In casual disentanglement research, the combination of two causal factors is often modeled as the addition between them [24]. We adopt this approach and combine the two behavior models using the addition operator:

$$f(u, v) = (\mathbf{u}^1 + \mathbf{u}^2) \cdot (\mathbf{v}^1 + \mathbf{v}^2) \tag{3}$$

which will preserve the features of different behavior models in the unified representation. By such an additional model, we do not exclude the possibility that a user can exhibit both the Chaser and the Sider behavior at the same time. Next, we will devise a graph neural network framework to model Chaser-Sider user behavior and produce the vector representation of

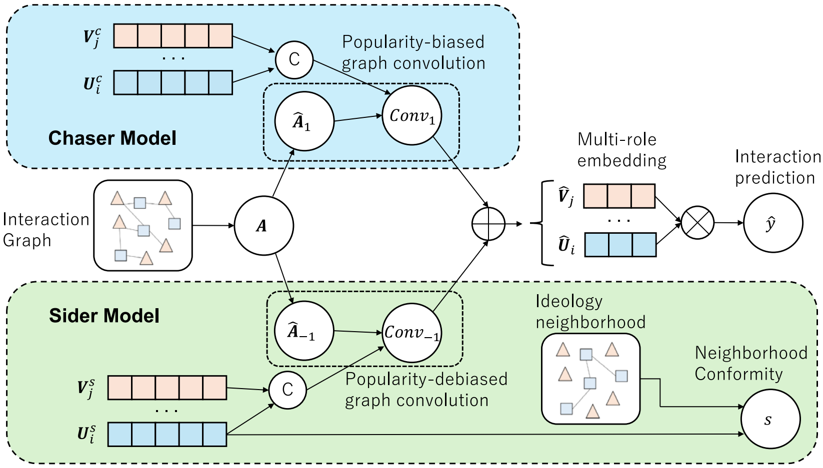


Figure 2 Overview of the Chaser-Sider Multi-role Embedding Learning (CS-MREL) model. It contains two main modules. On the upper part, the Chaser model is enhanced by popularity-biased graph convolution $Conv_1$. User and item Chaser embeddings U_i^c and V_j^c are concatenated for convolution with the biased adjacency matrix $\hat{A}_1$. On the lower part, the Sider model is similarly enhanced by popularity-debiased graph convolution $Conv_{-1}$. It furthermore strengthens the conformity behavior according to the ideology neighborhood, which is adjusted through a second learning objective, the neighborhood conformity score s. The main objective is to improve interaction prediction $\hat{y}$, which is calculated as the inner product of combined user and item embeddings $\hat{U}_i$ and $\hat{V}_j$

the two behavior models. Particularly, we will borrow the recent development of popularity-biased graph convolutional networks to learn the popularity-related information in the model.

4.2 Popularity-biased graph convolutional network

Graph convolutional networks (GCN) have been used for generating graph node embeddings that contain structural information, which is effective in various computational tasks, especially recommendations. The core technique of GCN is to aggregate neighbor information to a node using the convolution operation. Here, we briefly introduce this technique. Given a graph of N nodes, we should have an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ that indicates whether there is an edge between every pair of nodes. GCN first produces a normalized adjacency matrix

$$\hat{\mathbf{A}} = (\mathbf{D} + \mathbf{I})^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})(\mathbf{D} + \mathbf{I})^{-\frac{1}{2}} \tag{4}$$

where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$ and $\mathbf{I}$ is the identity matrix of $\mathbf{A}$. The normalized adjacency matrix is then put to the convolution operation, which go through several convolution layers. The first layer of convolution is run on the node features. In our case:

$$\mathbf{H}^{(1)} = \hat{\mathbf{A}} \times [\mathbf{U}||\mathbf{V}] \tag{5}$$

where $[\mathbf{U}||\mathbf{V}]$ is the concatenation of user and item features. Then in each subsequent convolution, the input is the output of the previous convolution:

$$\mathbf{H}^{(l)} = \hat{\mathbf{A}} \times \mathbf{H}^{(l-1)}. \tag{6}$$

More conveniently, we write a GCN as:

$$\mathrm{Conv}(\mathbf{U}||\mathbf{V}) = \{\hat{\mathbf{A}} \times \{...\{\hat{\mathbf{A}} \times [\mathbf{U}||\mathbf{V}\}...\}\} \tag{7}$$

The first convolution layer aggregates information of one-hop neighbors, while subsequent convolution aggregates neighbors of several hops away. The output of the last layer can be seen as the graph convolution node embedding.

Recently, a study proposed that the degree matrix $\mathbf{D}$ can be manipulated to integrate popularity bias into the model [16]. In what is called k-degree-biased normalization, the normalized adjacency matrix is calculated as:

$$\hat{\mathbf{A}}_k = (\mathbf{D}+\mathbf{I})^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})(\mathbf{D}+\mathbf{I})^{-\frac{1}{2}+kc}. \tag{8}$$

The difference between $\hat{\mathbf{A}}_k$ and $\hat{\mathbf{A}}$ calculated using (4) is that it adds a weight kc to adjust the effect of the last factor $(\mathbf{D}+\mathbf{I})^{-\frac{1}{2}}$ of the equation, which contains popularity information. Here, k is a user parameter, and c is a constant, which is usually set to 0.1. The convolution layers follow the same formula except that $\hat{\mathbf{A}}$ is replaced by $\hat{\mathbf{A}}_k$:

$$\mathrm{Conv}_k(\mathbf{U}||\mathbf{V}) = \{\hat{\mathbf{A}}_k \times \{...\{\hat{\mathbf{A}}_k \times [\mathbf{U}||\mathbf{V}\}...\}\}. \tag{9}$$

The above formulas construct what is called a popularity-biased graph convolutional network. In the original paper [16], k is set to positive values, e.g., 1, 2, and 3, to enhance the effect of popularity. However, we propose that while setting k to a positive value makes the model sensitive to popular items, setting k to a negative value can make the model sensitive to unpopular items. In other words, a negative k leads to popularity-debiased graph convolution. In our Chaser-Sider model, the Chaser model is influenced more by popular items and thus can be enhanced by a positive k. Since siding behavior looks at neighbor preferences rather than item popularity, the Sider model is influenced less by popular items and thus can be enhanced by a negative k.

4.3 The chaser model

The chaser model describes how users would behave given the popularity of the items they interacted with. Compared to a standard user model, higher degrees of popularity are emphasized to have more impact. To do this, we only need the popularity-biased graph convolution we introduced in the last section.

First, we set up chaser model user and item embeddings $\mathbf{U}^c$ and $\mathbf{V}^c$. Next, we select a positive k value, e.g., 1, and perform biased convolution using the biased normalized adjacency matrix $\hat{\mathbf{A}}_1$ calculated from (8). The convolution will give us convolution user and item embeddings:

$$\hat{\mathbf{U}}^c, \hat{\mathbf{V}}^c = \mathrm{Conv}_1(\mathbf{U}^c||\mathbf{V}^c) \tag{10}$$

As the result, embeddings $\hat{\mathbf{U}}_1^c$ and $\hat{\mathbf{V}}_1^c$ would contain the chaser behavior model. This model will be learned during the training phases with a particular focus on popular news, which is achieved through the biased normalization. It does not mean the model will always recommend popular news. This Chaser model describes how users react to popular news, and it is possible that some users like popular news while others do not.

4.4 The sider model

The sider model describes how users would behave given their similarity with other users. We can assume that siders are influenced less by popularity. As such, we can utilize the popularity-debiased graph convolution. The mechanism is similar to the biased convolution used in the chaser model, except that we use a negative value for k, e.g., -1. We set up sider model user and item embeddings $\mathbf{U}^s$ and $\mathbf{V}^s$. The convolution user and item embeddings are generated from:

$$\hat{\mathbf{U}}^s, \hat{\mathbf{V}}^s = \mathrm{Conv}_{-1}(\mathbf{U}^s||\mathbf{V}^s) \tag{11}$$

However, just reducing the effect of item popularity does not describe the sider behavior completely. We need to model conformity when learning the model. The conformity can be calculated by comparing the user features with the actual user ideology neighborhood. In this paper we assume no external information, so the user ideology neighborhood needs to be calculated directly from the interaction data.

Since user i has interactions with a set of items $\mathcal{I}_i \subseteq \mathcal{V}$, we calculate the mutual information between user i and j as

$$\mathrm{mi}(i, j) = |\mathcal{I}_i \cap \mathcal{I}_j|/N \log\left(\frac{|\mathcal{I}_i \cap \mathcal{I}_j|/N}{|\mathcal{I}_i|/N \times |\mathcal{I}_j|/N}\right) \tag{12}$$

where N is the total number of interactions in the data. We then use a threshold δ to make connections among users. Specifically, we construct an ideology neighborhood graph $\mathbf{B}$, in which $\mathbf{B}_{i,j} = 1$ if $\mathrm{mi}(i, j) > \delta$, otherwise 0. In our experiments, we choose a value for δ that allows the number of user-user edges to be similar to the number of user-item interaction edges, which tends to provide better results. In this way, users who have connections in the ideology neighborhood should have similar interests on some sets of news articles.

Next, we calculate a conformity score s. The score is based on user feature similarity and neighborhood connections:

$$s = \frac{1}{N}\sum_i\sum_j \mathbf{B}_{i,j} \times CosSim(\mathbf{U}_i^s, \mathbf{U}_j^s) \tag{13}$$

where $CosSim(a, b)$ is the cosine similarity of two embeddings a and b.

Following the multi-task learning approach, in order to make our model learn user conformity, we make a secondary loss function to enlarge the conformity score, calculated as:

$$\mathcal{L}_{conform} = 1 - s. \tag{14}$$

By enlarging the conformity score, users in the same ideology neighborhood will have even more similar embeddings. This will lead to recommendations of news articles present in the neighborhood but not otherwise in the user's interest.

4.5 Model combination and learning

As we explained in Section 3.1, we use the addition operator to combine the two models into a unified model. From the last two sections, we obtain chaser and sider model embeddings from the convolution outputs, denoted as $(\hat{\mathbf{U}}^c, \hat{\mathbf{V}}^c)$, and $(\hat{\mathbf{U}}^s, \hat{\mathbf{V}}^s)$, respectively. The combined multi-role embeddings are thus

$$\hat{\mathbf{U}}_i = \hat{\mathbf{U}}_i^c + \hat{\mathbf{U}}_i^s, \tag{15}$$

$$\hat{\mathbf{V}}_j = \hat{\mathbf{V}}_j^c + \hat{\mathbf{V}}_j^s. \tag{16}$$

We use the multi-role embeddings to make the final prediction of user-item preference:

$$\hat{y}_{uv} = f(u, v) = \hat{\mathbf{u}} \cdot \hat{\mathbf{v}}, \text{ for } \hat{\mathbf{u}} \in \hat{\mathbf{U}}, \hat{\mathbf{v}} \in \hat{\mathbf{V}}. \tag{17}$$

When learning the model, we follow the standard recommendation system practice, which randomly selects some negative items, i.e., items that have no interaction with the user, to be used in training along with the positive items. In this way, we can calculate a binary cross-entropy loss from the predicted and true preference:

$$\mathcal{L}_{pref} = - \sum_{(u,v)\in\mathcal{I}^+\cup\mathcal{I}^-} y_{uv} \log(\hat{y}_{uv}) + (1 - y_{uv}) \log(1 - \hat{y}_{uv}). \tag{18}$$

where $\mathcal{I}^+$ and $\mathcal{I}^-$ are the positive interaction set and randomly sampled negative interaction set, separately, $y_{uv} = \{0, 1\}$ is the true label, and $\hat{y}_{uv}$ is the prediction.

Finally, we combine the two loss functions, which then can be used to update the model parameters through back-propagation:

$$\mathcal{L} = \mathcal{L}_{pref} + \mathcal{L}_{conform}. \tag{19}$$

Algorithm 1 CS-MREL model learning.

Input: user-item interaction adjacency matrix $\mathbf{A}$, ideology neighborhood matrix $\mathbf{B}$, $nEpoch$
Output: Multi-role Embedding $\hat{\mathbf{U}}$, $\hat{\mathbf{V}}$

1: randomly initialize $\mathbf{U}^c, \mathbf{V}^c, \mathbf{U}^s, \mathbf{V}^s$
2: **for** $epoch = 1$ **to** $nEpoch$ **do**
3: $\hat{\mathbf{U}}^c, \hat{\mathbf{V}}^c = \text{Conv}_1(\mathbf{U}^c||\mathbf{V}^c)$
4: $\hat{\mathbf{U}}^s, \hat{\mathbf{V}}^s = \text{Conv}_{-1}(\mathbf{U}^s||\mathbf{V}^s)$
5: $\hat{\mathbf{U}}_i = \hat{\mathbf{U}}^c_i + \hat{\mathbf{U}}^s_i \quad \forall i \in \mathcal{U}$
6: $\hat{\mathbf{V}}_j = \hat{\mathbf{V}}^c_j + \hat{\mathbf{V}}^s_j \quad \forall j \in \mathcal{V}$
7: $\hat{y}_{uv} = \hat{\mathbf{u}} \cdot \hat{\mathbf{v}}, \text{ for } \hat{\mathbf{u}} \in \hat{\mathbf{U}}_{train}, \hat{\mathbf{v}} \in \hat{\mathbf{V}}_{train}.$
8: $\mathcal{L}_{pref} = -\sum_{(u,v)\in\mathcal{I}^+\cup\mathcal{I}^-} y_{uv} \log(\hat{y}_{uv}) + (1-y_{uv}) \log(1-\hat{y}_{uv})$
9: $s = \frac{1}{N} \sum_i \sum_j \mathbf{B}_{i,j} \times CosSim(\mathbf{U}^s_i, \mathbf{U}^s_j)$
10: $\mathcal{L}_{conform} = 1 - s.$
11: $\mathcal{L} = \mathcal{L}_{pref} + \mathcal{L}_{conform}$
12: update model parameters using $\mathcal{L}$
13: **end for**
14: **return** $\hat{\mathbf{U}}$, $\hat{\mathbf{V}}$

Algorithm 1 summarizes our procedure of learning the multi-role embedding. We can see that except for the user-item interaction adjacency matrix $\mathbf{A}$ and the ideology neighborhood matrix $\mathbf{B}$, we do not need additional data input. The ideology neighborhood matrix $\mathbf{B}$ can be constructed using only the interaction data as we described in Section 4.4. The number of epochs parameter $nEpoch$ can be eliminated if we use convergence detection and early stopping.

4.6 Inference

We obtain from the training phase the multi-role embeddings $(\hat{\mathbf{U}}, \hat{\mathbf{V}})$. In the test phase, where we are given a list of candidate items for a user, we use these embeddings to make predictions on all user-item pairs. The calculation is the same as calculating the prediction in the training phase, defined in (17). Then the preference scores $\hat{y}_{uv}$ are used to rank the items and make top-K recommendations.

5 Preliminary study of news and non-news datasets

Our assumption in this work is that the interaction pattern in news-viewing behavior is different from that in product shopping behavior. Specifically, they are different in the aspects of chasing popular items and siding with users of similar views. Before conducting recommendation experiments, we conduct a preliminary study to verify our hypothesis. We select a news-viewing dataset, called ADRESSA [29], and compare it with a movie review dataset, called MovieLens, and an online shopping dataset, which is the Grocery sub-category of the well-known Amazon dataset [30].

First, we look at the popularity-chasing behavior. Figure 3 (a), (b) and (c) show the distribution of items over different popularity groups for the news, grocery, and movie review datasets, respectively. The popularity of an item is calculated as the number of users who interacted with the item in the dataset. As we can see, while for all three datasets, the majority of the items are unpopular items (in bin 1), the news dataset does have a higher distribution in more popular items (bin 2 onward) than the grocery dataset. On the other hand, the movie review dataset has a higher distribution on unpopular items than the news dataset. This indicates that chasing popular items is most probable for movie watching, then for news reading, and grocery shopping is least likely for popular item chasing.

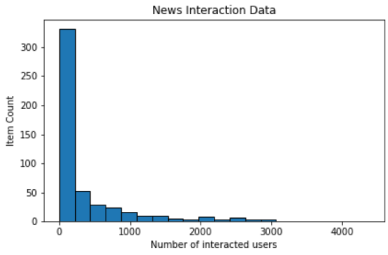
Then, we look at the siding behavior. We measure the mutual information (MU) between all user pairs based on their interacted items. Then, we calculate the average MU for each user, which indicates the strength of user siding behavior. We plot the distribution of average MU as shown in Figure 3 (d), (e), and (f) for the news, grocery, and movie review datasets, respectively. Here, we see a clearer distinction. For the grocery dataset, most users have a weak similarity to other users, causing them to have the lowest average MU. This indicates that there is little siding behavior in grocery shopping. For movie review, a large portion of users have some weak siding behavior, while a small portion of users have strong siding behavior, as indicated by a decreasing distribution. On the other hand, for news viewing behavior, we see a different pattern. The average MU takes a distribution that looks like a normal distribution, which indicates that most users have a typical degree of similarity to other users, and the distribution is similar for weaker and stronger siding users.

Now we see the evidence that news viewing behavior is different from movie watching and product shopping with regards to popular item chasing and neighborhood siding, we are motivated to test our model that takes into account these peculiar behavior patterns in recommendation experiments.

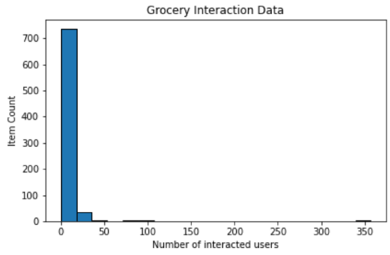
6 Experimental evaluation

We conduct experimental evaluations to test the effectiveness of our proposed method. We aim to answer four research questions:

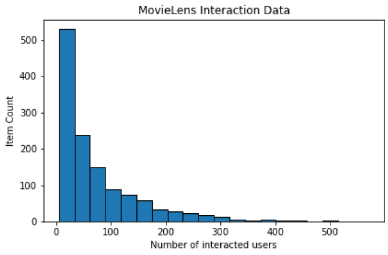
- **RQ1**: How will our method perform in comparison to state-of-the-art methods, in terms of recommendation accuracy?
- **RQ2**: How do the chaser model and the sider model in our combined model contribute to recommendation accuracy separately?



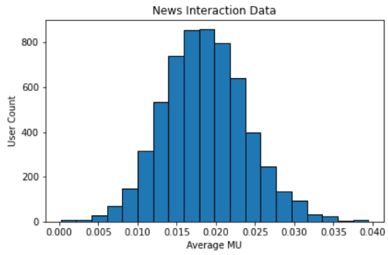
(a) News item popularity



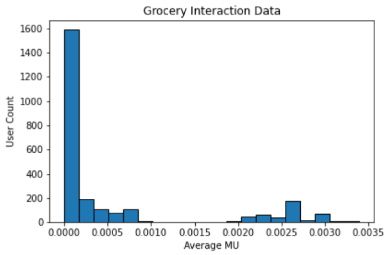
(b) Grocery item popularity



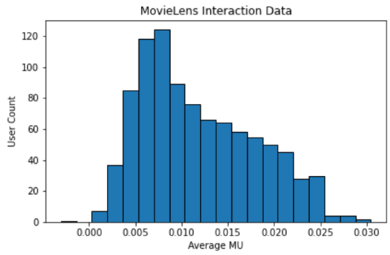
(c) Movie review item popularity



(d) News user siding



(e) Grocery user siding



(f) Movie review user siding

Figure 3 Distribution differences of news, grocery, and movie recommendation data regarding item popularity and user mutual information with other users

- **RQ3**: Is there any behavioral difference for the chaser and sider model?
- **RQ4**: How well can our method perform for non-news interaction data?

In this section, we will discuss our experimental datasets, baseline methods, and experiment settings, before presenting the evaluation results.

6.1 Datasets

We use two real-world datasets in the experiment. The first is called JP-POLI, which we collected from Twitter[1]. We first collect past tweets of over 100,000 Twitter users who are followers of Japanese politicians. Each tweet contains text and a username, and the original tweet if the tweet is a retweet. Then, we identify news tweets by searching "RT @nhk" in the tweet text. "RT" is Twitter's mark of retweeting, and "@nhk" is a famous Japanese news

[1] Renamed to X recently.

channel. In this way, we collected over 10,000 news tweets and the users who retweeted them. The retweeting action is treated as the user-item interaction in this dataset. The second is called ADRESSA, which is made publicly available [29] and has been the benchmark in several existing works [20]. The dataset is based on the online news platform Adressa[2], and contains the news viewing behavior of several thousands of users. The clicking action is treated as the user-item interaction in this dataset (Table 1).

For both datasets, we select a subset of popular news items for experimenting. Specifically, we select users who interact with at least 30 news items, and from there, we select news items that are interacted with by at least 10 users. The numbers of users, items, and interactions of the final datasets are summarized in Table 2. When creating training and test sets, we follow the leave-one-out strategy, which picks one news item from all news items interacted with by each user, and the rest goes to training data.

6.2 Baseline methods

We have discussed several relevant methods in the Related Work section. In our experiment, we pick the state-of-the-art methods from them for comparison. First, we have base methods GCN [10] and LightGCN [31] that other baseline methods are built upon. Then we have graph neural network news recommendation models GNUD [20] and GERL [14]. Third, we have state-of-the-art graph-based recommendation models with an emphasis on interest disentanglement recommendation, which include IMP-GCN [11], JMP-GCF [16], and GL-HGNN [25].

6.3 Experiment settings

We implement all compared models using Python and Tensorflow. For all graph neural network models, the number of convolution layers is set to 3. The dimensions of user and item embeddings for all models are set to 20. For measuring recommendation accuracy, we use hit rate HR@K[3] and NDCG@K, which are common accuracy metrics in existing works. The experiments are conducted on a workstation with an Intel i9-14900K CPU, 64 GB memory, and a GeForce RTX 4080 SUPER GPU with 16 GB dedicated memory. The code and data of this paper will be made available soon.

6.4 Recommendation accuracy results

To answer research question 1, we evaluate the recommendation accuracy of all compared methods. The results are shown in Table 2. We get several insights from the results. First, our proposed method CS-MREL outperforms all baseline methods in both datasets. Second, the second best method is GERL, which outperforms non-news graph-based recommendation models in most measures. This shows that designing a recommendation model specifically for news recommendation has an advantage over general-purpose recommendation models. Third, GCN and LightGCN are outperformed by most of the advanced methods, which shows the advantage of deeper modeling in graph-based recommendation. Considering all results, we conclude that our method can achieve superior performance mostly due to its explicit user modeling that describes accurately user news-seeking behavior.

[2] https://www.adressa.no/

[3] Which is equivalent to Recall@K when there is only one target item in item candidates

Table 1 Dataset statistics of two recommendation datasets

	JP-POLI	ADRESSA
number of users	3,095	5,968
number of news items	7,115	511
number of interactions	100,500	206,994

6.5 Ablation study

To answer research question 2, we conduct an ablation study. Specifically, we compare the recommendation performance of the chaser and sider models by running them separately. The results are shown in Figure 4. As we can see from the results for both datasets, the chaser model reaches a higher accuracy than the sider model. This means that for accurate news recommendations in general, the behavior of seeking popular items is more critical than the behavior of following friends. The combined model CS-MREL reaches better results than either individual model, showing that both models contribute positively when they are combined.

6.6 Model behavior analysis

To answer research question 3, we conduct in-depth analyses of the recommendation models, paying particular attention to the difference between the Chaser and Sider models.

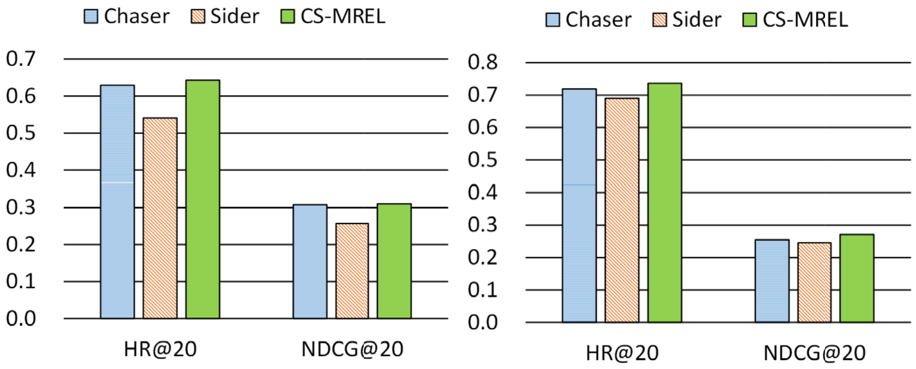
6.6.1 Recommendation rate over different popular groups

First, we study it from the perspective of diversity. For this purpose, we follow an existing approach [32] for measuring diversity. We divide the items into 10 groups according to the order of their interaction number in the training data. Group 1 contains the most popular items, group 10 contains the least popular items, and so on. The total numbers of interactions in training data for each group are kept to the same. This means that more popular item groups will contain fewer items, while less popular item groups will contain more items. Then, we perform recommendation on the test dataset and calculate the popularity group

Table 2 Recommendation accuracy results of compared methods on two tested datasets

	Dataset: JP-POLI				Dataset: ADRESSA			
	H@10	H@20	N@10	N@20	H@10	H@20	N@10	N@20
GCN	0.381	0.531	0.216	0.253	0.365	0.711	0.164	0.252
LightGCN	0.380	0.534	0.216	0.254	0.378	0.704	0.168	0.251
GNUD	0.412	0.554	0.231	0.266	0.404	0.713	0.173	0.251
GERL	<u>0.451</u>	<u>0.625</u>	<u>0.255</u>	<u>0.299</u>	<u>0.406</u>	0.715	<u>0.182</u>	<u>0.260</u>
IMP-GCN	0.386	0.538	0.217	0.255	0.381	0.703	0.168	0.250
JMP-GCF	0.408	0.557	0.229	0.267	0.396	<u>0.725</u>	0.174	0.259
GL-HGNN	0.370	0.510	0.206	0.241	0.394	0.706	0.173	0.253
CS-MREL	**0.489**	**0.657**	**0.275**	**0.318**	**0.440**	**0.785**	**0.197**	**0.285**

The best results are highlighted in bold font, and the second-best results are underlined



(a) JP-POLI dataset (b) ADRESSA dataset

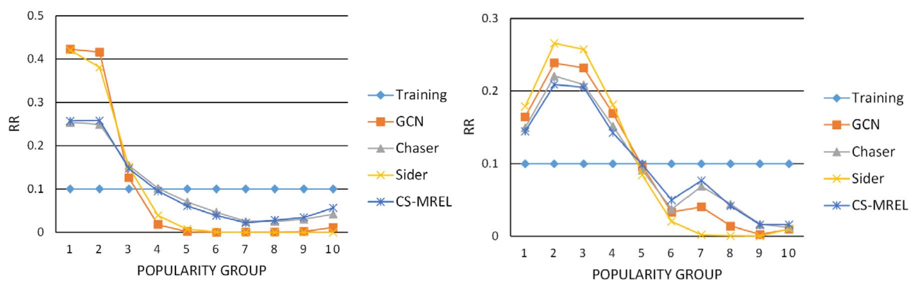
Figure 4 Ablation study that shows the contributions of Chaser and Sider models in overall accuracy

recommendation rate (RR):

$$RR_i = \frac{\text{RecCount}_i}{N \times K}$$

where RecCount_i is the number of items in group i that appear in top-K recommendations in N test cases. If a recommendation model is not biased, the popularity group recommendation rates should have the same distribution over groups as in the training data, which is nearly 0.1 for all groups. If the model is biased towards more popular groups, then the recommendation rate will be higher for more popular groups and lower for less popular groups. Consequently, a smaller difference in the recommendation rate of different popularity groups indicates higher diversity. We follow the existing work [32] and calculate the standard deviation of the recommendation rate over 10 groups as a one-value indicator of diversity.

The result of recommendation rate distribution over popularity groups for GCN, Chaser, Sider, and CS-MREL models is shown in Figure 5. The standard deviation summary is shown in Table 3. The results for both datasets show a similar trend. GCN and sider models are influenced more strongly by the popularity bias, as they tend to recommend more popular items, while chaser and combined models receive less influence. The reason is that by popularity-biased graph convolution, users who seek popular news and users who do not are separated



(a) JP-POLI dataset (b) ADRESSA dataset

Figure 5 Recommendation rate distribution of popularity groups for different recommendation models. Being closer to the training distribution (constant at 0.1) means the model is less influenced by popularity bias

Table 3 Standard deviation of recommendation rates

	GCN	Chaser	Sider	CS-MREL
JP-POLI	0.173	**0.089**	0.165	0.091
ADRESSA	0.093	0.078	0.110	**0.072**

A lower number means recommended items are distributed more evenly across popularity groups. The lowest numbers are highlighted in bold font

more clearly. For the chaser model, users who do not seek popular news are much less likely to receive popular item recommendations. This is not so for the sider or the base GCN model, which shows normal popularity bias. Even though the sider model takes into account user ideology neighborhood, it has little effect on reducing the popularity bias.

6.6.2 Embedding visualization

Second, we visualize the embeddings of the Chaser and Sider models. Using the dimension reduction technique t-SNE, we reduce the user embeddings of two models to two dimensions and plot them in the figures, which are shown in Figure 6. As we can see, there is a clear separation between Chaser and Sider embeddings, which shows their different emphasis on user preferences. Particularly, we see that in the JP-POLI dataset, Chaser embeddings are concentrated in the middle, while Sider embeddings are polarized into two sides, which is a good reflection of the meaning of the two sets of embeddings. By this visualization, we get a hint on why these two models would recommend different sets of news articles.

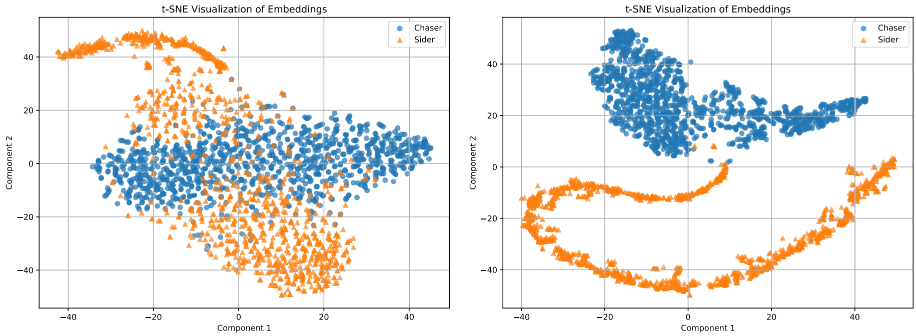
6.6.3 Actual recommended news articles

Last, we look at actual news articles recommended by different models. Table 4 shows the title[4] of top news articles in our JP-POLI dataset that are most frequently recommended by the chaser and the sider model. To show the difference, we omit the articles recommended by both models. Because of the collection period, both sets contain several news articles about Covid-19. However, articles from the chaser model focus more on neutral events, such as power outages and shortage of masks. On the other hand, articles from the sider model tends to be more political, as it contains criticism of politicians and large corporations. This is consistent with our expectation of the chaser and sider models.

6.7 Popularity bias degree test

The popularity-biased graph convolutional network has a parameter k that determines the degree of bias. Generally, when k is set to a positive number, the model will be more sensitive to popular items. When k is set to a negative number, the model will be less sensitive to these items. However, it is uncertain that how a low or high value of k impacts the model learning. Here we conduct a sensitivity test of k. Specifically, we choose three k values for the Chaser convolution, (1, 2, 3), and three k values for the Sider convolution, (-1, -2, -3). Then we learn the CS-MREL model with different combinations of two k values, and obtain the recommendation accuracy. The NDCG@10 results are shown in Figure 7. As we can see,

[4] translated from Japanese by ChatGPT



(a) JP-POLI dataset (b) ADRESSA dataset

Figure 6 Embedding visualization of Chaser and Sider user embeddings. The technique t-SNE is used to reduce embedding dimensions to two

the best result is achieved by two k values of 3 and -3. Therefore, in our main experiments, we use these k values by default when testing our method.

6.8 Performance on non-news datasets

As stated in research question 4, we would also like to investigate the effectiveness of our method on non-news data. We use the datasets we discussed in the preliminary study in Section 5, which include a grocery shopping dataset collected from Amazon and a movie review dataset from MovieLens. The grocery dataset has 2,507 users, 793 items, and 10,768 interactions between them. The MovieLens dataset has 943 users, 1,283 items, and 97,171 interactions. As we discussed, these datasets are different from the news-viewing dataset in terms of the ratio of popular items and the average user similarity to other users. The recommendation accuracy results on these two datasets for all compared methods are shown in Table 5. The results also include individual performances of the chaser and sider models.

First, we look at the grocery dataset. We see that our method does not perform as well as it does for the news dataset. Although our method outperforms GCN by a large margin, it is worse than LightGCN and IMP-GCN, which are designed as general recommendation models. Models designed specifically for news recommendation, GNUD, and GERL, also do not perform very well, even though GERL was the second-best method for our news recommendation evaluation. LightGCN and IMP-GCN based on it perform quite well because the user preference for grocery shopping is easier to capture, which suits simplistic models. The accuracy of our model comes mostly from the Chaser model, because the Sider model provides very inaccuracy predictions. This is consistent with our preliminary study, which shows that Grocery purchasing has a much lower portion of siding behavior than news reading.

Then, we look at the MovieLens dataset. We see that for this dataset, our model again generally achieves the best performance. Particularly, unlike the Grocery dataset, the Chaser model and the Sider model both contribute significantly to the accuracy achieved by our model. Sider model even achieves a higher accuracy than the Chaser model in terms of hit-rate@20. This is reasonable because we see from the preliminary study that the movie review activity has a moderate level of siding behavior. Other methods that deal specifically with news recommendation also achieve better accuracy. Particularly, GERL again achieves

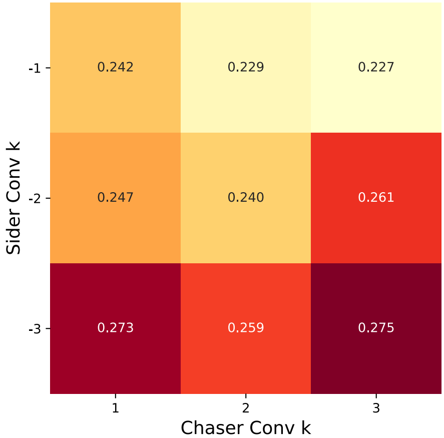
Table 4 Top recommended news by Chaser and Sider models

Chaser model
There is a power outage in Yamagata and Niigata prefectures.
It has been revealed that a scam phone call claiming, "You need to register a bank account to receive subsidies for coronavirus measures," has been made in Tokyo. The Metropolitan Police Department is warning the public to be cautious of this new type of scam.
If the number of patients continues to increase, there is a risk that lives that could normally be saved might not be saved.
In the United States, a pregnant woman who contracted the novel coronavirus and gave birth while in a coma regained consciousness about two weeks later.
In response to the spread of the novel coronavirus, Prime Minister Abe called on people across the country to refrain from using restaurants that provide entertainment in nightlife districts.
Chief Cabinet Secretary Suga stated, "We will do our utmost to resolve the shortage of masks and other items."
A new case of coronavirus infection has been confirmed in Chiba Prefecture in a woman in her 70s.
Due to the spread of the novel coronavirus, Sony has decided to cancel its student event scheduled for the 15th and 16th in Tokyo. The event was supposed to introduce corporate activities to students on a large scale, but Sony decided to prioritize the health and safety of attendees.
A New Year's fire drill ceremony, featuring the sharp dance moves of the team members, was held by the Tokyo Fire Department.
The labor union of private school teachers has requested compensation for the wages of part-time lecturers.
Sider model
A woman in her 80s from Kanagawa Prefecture, who was infected with the novel coronavirus, has passed away. This is the first case in Japan where a coronavirus-infected person has died.
A new case of coronavirus infection has been confirmed in Osaka.
Film director Nobuhiko Obayashi has passed away.
An ANA aircraft made an emergency landing at Fukuoka Airport due to engine trouble.
[A video] for those whose family members went missing in the earthquake: A short version has been uploaded. Kazuko-san, who appeared in the video, said, "I was happy that the dream of my husband, which I no longer see, was carefully recreated." You can watch the footage of that dream.
Two new cases of coronavirus infection have been confirmed in Osaka, one of whom attended a live house concert.
It has been confirmed that a mother and her infant daughter, who is less than a year old, have been infected with the novel coronavirus in Fukuoka.
Prime Minister Abe appeared in a Yoshimoto Shinkigeki performance.
What is a "Special Warning"? In a press conference on the morning of the 6th, the Japan Meteorological Agency mentioned the possibility of issuing a special warning for heavy rain. How is a special warning different from a regular warning? Take this opportunity to learn more.
Employees of Unizo have made an unusual self-buyout in response to a takeover proposal by a third party.

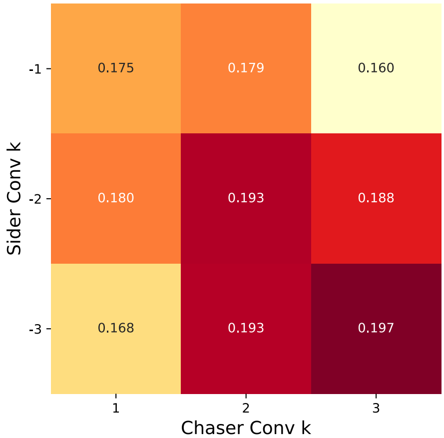
the second-best results. These results show that, even though the dataset is not from a news service, our news recommendation model can achieve superior accuracy as long as there is a clear siding behavior.

7 Conclusion

In this paper, we proposed a method called Chaser-Sider Multi-role Embedding Learning (CS-MREL) for news recommendation. Our method models two types of news-seeking behavior



(a) JP-POLI dataset



(b) ADRESSA dataset

Figure 7 NDCG@10 results for different combination of two k values in Chaser and Sider convolution operators

that come from public interest. The chaser model describes the behavior of seeking popular news. The sider model describes the behavior of seeking news of similar ideological stances. We integrate the behavior models neatly into a graph neural network news recommendation framework. In our experimental evaluation, we compare our method with several state-of-the-art graph-based recommendation models and show that our model achieves higher accuracy than all compared models. This confirms the effectiveness of modeling explicitly social behavior in news recommendation. In the future, we plan to extend our study to other recommendation scenarios, such as POI recommendation or social friend recommendation.

Table 5 Recommendation accuracy on non-news dataset

	Dataset: Grocery				Dataset: MovieLens			
	H@10	H@20	N@10	N@20	H@10	H@20	N@10	N@20
GCN	0.216	0.256	0.132	0.142	0.197	0.237	0.122	0.131
LightGCN	0.397	0.483	**0.261**	**0.282**	0.184	0.229	0.109	0.120
IMP-GCN	**0.450**	**0.572**	0.246	0.277	0.197	0.232	0.121	0.130
GNUD	0.269	0.331	0.194	0.210	0.196	0.260	0.122	0.137
GERL	0.259	0.313	0.175	0.189	0.200	**0.262**	0.141	0.157
JPM-GCF	0.272	0.329	0.177	0.191	0.201	0.249	0.134	0.146
GL-HGNN	0.202	0.239	0.163	0.172	0.076	0.090	0.046	0.050
CS-MREL	0.272	0.359	0.183	0.205	**0.211**	0.259	**0.150**	**0.162**
- Chaser	0.278	0.340	0.183	0.199	0.207	0.249	0.145	0.156
- Sider	0.193	0.266	0.121	0.140	0.196	0.256	0.120	0.135

The best results are highlighted in bold font

Author Contributions Yihong Zhang conceived the idea and performed the majority part of the work. Takahiro Hara supervised the project and contributed several ideas. Both Yihong Zhang and Takahiro Hara involved in writing and revising the manuscript.

Funding Open Access funding provided by The University of Osaka. No funding is received for this work.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Zhao, W.X., Li, S., He, Y., Chang, E.Y., Wen, J.-R., Li, X.: Connecting social media to e-commerce: cold-start product recommendation using microblogging information. IEEE Trans Knowl Data Eng **28**(5), 1147–1159 (2015)
2. Davidson J, Liebald B, Liu J, Nandy P, Van Vleet T, Gargi U, Gupta S, He Y, Lambert M, Livingston B, et al (2010) The youtube video recommendation system. In: Proceedings of the fourth ACM conference on recommender systems, pp 293–296
3. Gao H, Tang J, Hu X, Liu H (2013) Exploring temporal effects for location recommendation on location-based social networks. In: Proceedings of the 7th ACM Conference on Recommender Systems. ACM, pp 93–100
4. Wan S, Lan Y, Guo J, Fan C, Cheng X (2013) Informational friend recommendation in social media. In: Proceedings of the 36th International ACM SIGIR conference on research and development in information retrieval, pp 1045–1048
5. Wu, C., Wu, F., Huang, Y., Xie, X.: Personalized news recommendation: methods and challenges. ACM Trans Inf Syst **41**(1), 1–50 (2023)
6. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on world wide web. ACM, pp 285–295
7. Shan H, Banerjee A (2010) Generalized probabilistic matrix factorizations for collaborative filtering. In: 2010 IEEE international conference on data mining. IEEE, pp 1025–1030
8. Zhang, S., Tong, H., Xu, J., Maciejewski, R.: Graph convolutional networks: a comprehensive review. Comput Soc Netw **6**(1), 1–23 (2019)
9. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: Proceedings of the fifth international conference on learning representations ICLR
10. Wang X, He X, Wang M, Feng F, Chua T-S (2019) Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 165–174
11. Liu F, Cheng Z, Zhu L, Gao Z, Nie L (2021) Interest-aware message-passing GCN for recommendation. In: Proceedings of the web conference 2021, pp 1296–1305
12. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv (CSUR) **52**(1), 1–38 (2019)
13. Zheng G, Zhang F, Zheng Z, Xiang Y, Yuan NJ, Xie X, Li Z (2018) Drn: a deep reinforcement learning framework for news recommendation. In: Proceedings of the 2018 world wide web conference, pp 167–176

14. Ge S, Wu C, Wu F, Qi T, Huang Y (2020) Graph enhanced representation learning for news recommendation. In: Proceedings of the web conference 2020, pp 2863–2869
15. Auxier, B.E., Vitak, J.: Factors motivating customization and echo chamber creation within digital news environments. Soc Med Soc **5**(2), 2056305119847506 (2019)
16. Liu, K., Xue, F., He, X., Guo, D., Hong, R.: Joint multi-grained popularity-aware graph convolution collaborative filtering for recommendation. IEEE Trans Comput Soc Syst **10**(1), 72–83 (2022)
17. Qian Y, Zhao P, Li Z, Fang J, Zhao L, Sheng VS, Cui Z (2019) Interaction graph neural network for news recommendation. In: Web information systems engineering–WISE 2019: 20th international conference, Hong Kong, China, November 26–30, 2019, Proceedings 20. Springer, pp 599–614
18. Hu, L., Li, C., Shi, C., Yang, C., Shao, C.: Graph neural news recommendation with long-term and short-term interest modeling. Inf Process Manag **57**(2), 102142 (2020)
19. Qiu, Z., Hu, Y., Wu, X.: Graph neural news recommendation with user existing and potential interest modeling. ACM Trans Knowl Disc Data (TKDD) **16**(5), 1–17 (2022)
20. Hu L, Xu S, Li C, Yang C, Shi C, Duan N, Xie X, Zhou M (2020) Graph neural news recommendation with unsupervised preference disentanglement. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 4255–4264
21. Wang S, Chen X, Sheng QZ, Zhang Y, Yao L (2023) Causal disentangled variational auto-encoder for preference understanding in recommendation. In: Proceedings of the 46th international conference on research and development in information retrieval, pp 1874–1878
22. Zheng Y, Gao C, Chang J, Niu Y, Song Y, Jin D, Li Y (2022) Disentangling long and short-term interests for recommendation. In: Proceedings of the ACM web conference 2022, pp 2256–2267
23. Zhao Y, Zhang Y, Amagata D, Hara T (2023) Poi recommendation by learning short-, long-and mid-term preferences through GNN. In: 2023 24th IEEE international conference on mobile data management (MDM). IEEE, pp 1–10
24. Zheng Y, Gao C, Li X, He X, Li Y, Jin D (2021) Disentangling user interest and conformity for recommendation with causal embedding. In: Proceedings of the web conference 2021, pp 2980–2991
25. Zhang, Y., Wu, L., Shen, Q., Pang, Y., Wei, Z., Xu, F., Chang, E., Long, B.: Graph learning augmented heterogeneous graph neural network for social recommendation. ACM Trans Recomm Syst **1**(4), 1–22 (2023)
26. Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B.: Graph neural networks in recommender systems: a survey. ACM Comput Surv **55**(5), 1–37 (2022)
27. Joseph K, Jiang H (2019) Content based news recommendation via shortest entity distance over knowledge graphs. In: Companion proceedings of the 2019 world wide web conference, pp 690–699
28. Chen J, Dong H, Wang X, Feng F, Wang M, He X (2020) Bias and debias in recommender system: a survey and future directions. ACM Trans Inf Syst
29. Gulla JA, Zhang L, Liu P, Özgöbek Ö, Su X (2017) The adressa dataset for news recommendation. In: Proceedings of the international conference on web intelligence, pp 1042–1048
30. Ni J, Li J, McAuley J (2019) Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 188–197
31. He X, Deng K, Wang X, Li Y, Zhang Y, Wang M (2020) Lightgcn: simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 639–648
32. Zhang Y, Feng F, He X, Wei T, Song C, Ling G, Zhang Y (2021) Causal intervention for leveraging popularity bias in recommendation. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, pp 11–20