Received: March 8, 2022. Revised: April 7, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Graph Attention Network for Movie Recommendation

Thaair Ameen[1]* Ali Adil Ali[2]

[1]*Mosul University Presidency, University of Mosul, Iraq*
[2]*Ministry of Higher Education and Scientific of Research, Iraq*
* Corresponding author's Email: thaearm@gmail.com

Abstract: To provide more accurate, diverse, and interpretable recommendations, one must go beyond modeling user-movie interactions and consider user social correlation graphs and movie-movie correlation graphs. However, most previous research has only considered correlations between user preferences and social correlations, while ignoring the correlation between movies. In this paper, we describe a model for movie recommendations that include a representation of the user's preference in movies based on graph attention networks and multi correlations as in the following graphs. 1) user-movie graphs, which include user-movie interactions, as well as users' opinions on movies, 2) user social correlation graphs, which contain social correlations among users, and 3) movie-movie correlated graphs, which contain the correlation among the movies themselves. Extensive experiments are conducted on two real-world (Netflix and Movie Lens) datasets to verify the efficiency and stability of the proposed model.

Keywords: Attention graphs, Movie recommendation systems, Social correlations.

1. Introduction

Recommendations are essential in most of the current internet businesses, such as e-commerce, location recommendations, movie recommendations, and much more. Collaborative Filtering (CF) [1] is a crucial model for the recommendation task, which analyses the user's prior interactions with movies to forecast the most relevant content. It is another common practice to employ content-based recommendations, which analyse similarities between movies and users to recommend movies to the user based on the feature's similarity [2, 3]. However, matrix factorization [4, 5] is the most prevalent approach for CF, which projects users and movies onto a shared latent space, employing a vector of latent variables to represent a user or a movie. Traditional models cannot generate high-quality recommendations.

Recommendation systems [6, 7], natural language processing [8], and computer vision [9] all benefit from deep learning and neural networks. When used to learn node embedding on a graph, Neural Graph Networks (GNNs) often provide excellent results [10, 11]. Graph Attention Networks (GATs) use the attention mechanism to learn and perform on node embedding's. In addition, several scientists are examining the possibility of merging graph neural networks with collaborative filtering. It is possible to learn complicated user-movie interactions using Neural Graph Collaborative Filtering (NGCF) [12].

The social correlations that is connected between two or more users. A node represents each user in the dataset, and edge represents each social correlation (such as the relationship of two friends) between users. This sort of social correlations arises naturally from a homogenous social graph. It is difficult to design an algorithmic recommendation system around GATs. Users' social networks and user-movie graphs convey information about them in different ways. Using two graphs to develop better user representation requires a lot of data collection. Therefore, combining these two graphs is the first hurdle that we need to overcome. Furthermore, the graph of the user-movie interactions and the user's opinions on movies is also included.

Another important source of information for describing movies and improving movie recommendations is the correlation between movies. This is because movies do not exist independently and are more likely to be linked or are likely to be similar [4, 13]. Those who have seen an action movie are more inclined to see another action movie since both these movies have the same characteristics. It is advisable to consider the correlation between movies to improve their representation learning, which can enhance movies recommendation.

We propose an Aggregation Graph Attention Network (AGAT) for creating stronger user and movie representations. In addition, our model incorporates user opinions about movies while modelling the user-movie interaction to enhance representation learning. Furthermore, an attention mechanism is used to capture social correlations among users in user modeling. Lastly, we also exploit movie-movie correlation to better represent movies. Significant contributions in this paper can be summarized as follows:

- We propose a novel AGAT that can improve movie recommendations by learning better representations of users and movies from graph data.
- We provide a principled approach for jointly capturing interactions and opinions in the user-movie graph.
- Two large datasets have been used to show the proposed framework's efficacy.

2. Related work

2.1 General recommendation

Recommender systems are classified into three types: content-based models, CF, and hybrid models [14]. Among all the models for building recommender systems, latent variable-based models in the CF category are the most popular. This is because of their relatively high performance in practice [15, 16]. These latent variable-based models decompose users and movies in a low latent variable space, and the user's preference for movies can be viewed as fitting the inner product between the user and movie latent vectors. In real-world applications, users often express their opinions implicitly, through action or inactivity, and not through explicit ratings. Bayesian Personalized Ranking (BPR) is a common latent variable-based model that can handle implicit feedback [17]. Specifically, BPR optimizes a pairwise-based ranking loss in such a manner that the implicit feedback that is seen tends to rank higher than the unobserved one. Because users can express their thoughts quickly through many feedback channels (e.g., past interactions) at the same time. SVD++ is conceived as a way to incorporate different types of feedback from users. It is believed that everyone has a base latent variable and an auxiliary latent variable that can be derived from other types of feedback [18].

2.2 Social recommendation

A user's previous preference in social correlations is influenced by other users related to him, as shown by some classic research findings [19]. This theory is then applied by social recommendation to enhance the performance of recommendation systems. Social regularisation [20] present matrix factorization model to address the problem of trust aware recommendation for social recommendation. Trust based recommendation model [21] extends the model SVD++ by joining the implicit and explicit influence of trusted users on the movie estimates of active users [22]. Trust SVD [23] uses implicit and explicit effects of user-movie ratings to make predictions.

2.3 Neural attention mechanism

Attention mechanisms in neural networks have been found to be beneficial in many machine learnings [24] and neural machine translations [25]. A big advantage of attention is that it provides supervision to the neural network, and parts with more weight contain more informative features and these should be paid more attention. It is also applied in recommendation to find important and useful parts in textual reviews, such as specific features, words, sentences. Wu [26] introduced an attention mechanism to better capture the hidden information in implicit feedback, supported the learning ability of neural networks, and designed a balance module to improve the overfitting problem generated by highly sparse interactive information. Xi [27] used an attention mechanism with a graph neural network to discover the value of recommendations. Fan [10] introduced a model that includes user interactions with movies, as well as users' opinions of movies, and also captures social correlations among users to enhance the user representation model.

In this paper, we not only consider social correlations for user modelling with capturing interaction and opinions in the user-movie graph, but also use movie-movie correlation to understand movie correlations, which can enhance movie recommendations.

Received: March 8, 2022. Revised: April 7, 2022.

Table 1. Notation

Symbols	Definition and Descriptions
r_{ij}	the rating value of movie v_j by user u_i
$\mathbf{p}_\mathrm{i}$	the embedding of user u_i
$\mathbf{q}_j$	the embedding of movie v_j
e_r	the opinion embedding for the rating level r, such as 4-star rating, $r \in \{1,2,3,4\}$
d	the length of embedding vector
$C(i)$	the set of movies which user u_i interacted with
$N(i)$	the set of social friends who user u_i directly connected with
$B(j)$	the set of users who have interacted the movie v_j
$M(j)$	the set of movies which are similar/related to the movie v_j
$\mathbf{h}_i^S$	the social space user latent variable from the social friends $N(i)$ of user u_i
$\mathbf{h}_i^H$	the movie space user latent variable from movie set $C(i)$ of user u_i
$\mathbf{h}_i$	the user latent variable of user u_i, combining from movie space $\mathbf{h}_i^I$ and social space $\mathbf{h}_i^S$
$\mathbf{z}_j$	The movie latent variable of movie v_j
$\mathbf{h}_j^U$	the user space movie latent variable from user set $\mathrm{B}(j)$ of movie v_j
$\mathbf{h}_j^V$	the movie-movie space movie latent variable from the similar/related movies set M(j) of movie v_j
x_{ij}	the opinion-aware interaction representation of movie v_j for user u_i
α_{ij}	he movie attention of movie v_j in contributing to $\mathbf{h}_i^V$
$\beta_i o$	the social attention of neighboring user u_o in contributing to $\mathbf{h}_i^S$
μ_{it}	the user attention of user u_t in contributing to $\mathbf{h}_j^U$
k_{jk}	the movie attention of movie v_t in contributing to $\mathbf{h}_j^V$
r'_{ij}	the predicted rating value of movie v_j by user u_i
$\oplus$	the concatenation operator of two vectors
T_{ik}	the user social correlation graph
$\mathcal{O}_{ja}$	the movie-movie correlation graph
$\mathbf{R}$	the user-movie rating matrix (user-movie graph)
W, b	the weight and bias in neural network

3. Preliminaries

Let $u_i(1,2,\cdots,n)$ and $v_j(1,2,\cdots,m)$, be the set of n users and m movies, respectively. According to $\mathbf{R} = \{r_{ij}\} \in \mathbb{R}^{n\times m}$, is the user rating matrix, or user-movie graph. The rating score will be one if a user u_i watch (interaction) with movie v_j, and zero if there is no interaction between user u_i and movie v_j. According to user u_i interaction on the movie v_j, the rating score can be seen as user opinion on the movie. Let $\mathrm{T}_{ik}(i,k = 1,2,\cdots,n; i \neq k)$ in which T_{ik} is in binary and has two potential values, one and zero. $\mathrm{T}_{ik} = 1$, indicates social correlation graph between two users, while no such correlation is indicated when $\mathrm{T}_{ik} = 0$. Similarly, the movie correlation graph is represented by $\mathcal{O}_{ja}(j,a = 1,2,\cdots,m; j \neq a)$, a binary representation of the two movies, will be one if there are correlations and will be zero if there are no correlations. Let $N(i)$ be the set of users whom u_i is directly connected with in the user social graph T, $C(i)$ be the set of movies with user u_i has interacted with in **R**.

Our aim is to predict an unknown rating value in **R** from the above mathematical notations, user-movie graph **R**, movie-movie correlation graph $\mathcal{O}$ and social correlation graph T,. Next [4, 28], we use an embedding vector $\mathbf{p}_\mathrm{i} \in \mathbb{R}^d$ to represent a user u_i, and an embedding vector $\mathbf{q}_\mathrm{i} \in \mathbb{R}^d$ to denote a movie v_j, where d is the length of the embedding vector. The mathematical notation used in this paper is summarized in Table 1.

4. Methodology

The proposed framework has three main parts. 1) User modelling identifies users' latent variables. Both social graphs and user-movie graphs can be used to learn user representations. Therefore, two aggregates are added to accommodate these two graphs. Interacting with movies in a user's movie map can provide information about the user. Social graphs can be used to discover user correlations that can be further used for user modelling. 2) Movie modelling aims to learn movie latent variables. Movie representations can be learned from user interactions in a user-movie graph and from movie perspectives in a movie-movie graph. These two aggregations gather important information from two graphical representations. As shown in Fig. 1, our model combines user and movie modelling to make the rating prediction.

4.1 User modelling

The goal of user modeling is to find out the latent variables of users, which are expressed as $\mathbf{h}_i \in \mathbb{R}^d$of user u_i. The challenge is the combination of the two representations, which are the user-movie graph and the social correlation graph. To overcome this challenge, we learn variables from two aggregating models, as illustrated in the left part of Fig. 1. We combine the two aggregations to learn a user latent

Received: March 8, 2022. Revised: April 7, 2022.

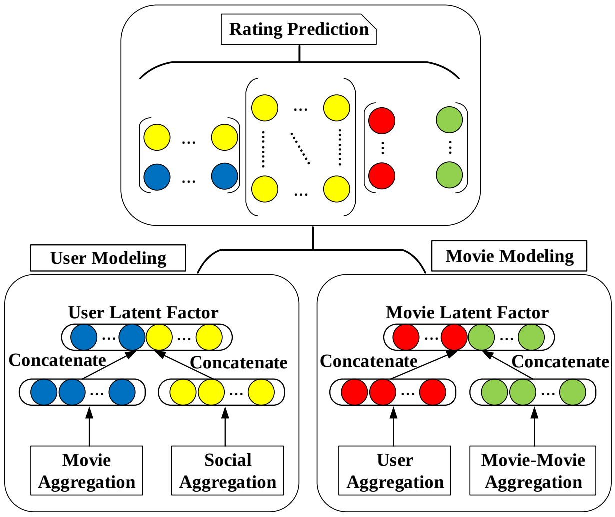

Figure. 1 The proposed framework including user modelling, movie modelling, and rating prediction

variable. The first aggregation, referred to a movie aggregation, is used when learning the movie space for the user-movie graph. It is based on the movie space user latent variable $\mathbf{h}_i^H \in \mathbb{R}^d$ for the user-movie graph. Social aggregation, is used to learn the social space $\mathbf{h}_i^S \in \mathbb{R}^d$. The final user latent variable $\mathbf{h}_i$ is obtained by combining the previously mentioned spaces.

4.1.1. Movie aggregation

Since the user-movie graph includes not only the interaction between the user and the movie, but also the user's opinion (or rating) of the movie, we propose a model for learning movie space based on the user latent variables $\mathbf{h}_i^H$ for a user u_i, which is used to model user latent variables through interactions in the user-movie graph. The goal of movie aggregation is to learn the movie space based on the user latent variables $\mathbf{h}_i^H$ of the user u_i by evaluating the movies that the user has interacted with and the user's opinion about those movies. The following functions are used to mathematically represent this aggregate:

$$\mathbf{h}_i^H = \sigma\left(\mathbf{W}.AGG_{Movies}\left(\{\mathrm{x}_{ij}\forall a \in C(i)\}\right) + \mathbf{b}\right) \quad (1)$$

where, $C(i)$ refers to a set of movies that the user u_i has interacted with (or u_i's neighbors in the user-movie graph), x_{ij} denotes vector representation between u_i and a movie v_j, and AGG_{Movies} is the note of aggregation function of the movie. A non-linear activation function (i.e., a rectified linear unit) is also represented by the symbol, where $\mathbf{W}$ and $\mathbf{b}$ are the weight and bias of a neural network, respectively. Following that, we'll go through how to define the opinion aware interaction representation x_{ij} and the aggregate function AGG_{Movies}.during the user's interaction with the movie, the note r refers to user's opinion of the movie. These movie opinions can be used to capture the user's preference in the movie. For each opinion r, we introduce an opinion embedding vector $e_r \in \mathbb{R}^d$ to represent each opinion r as a vector representation. For example, in a 4-star rating, for each $r \in \{1,2,3,4\}$, we introduce an embedding vector e_r. For the interaction between user u_i and the movie v_j with opinion r, we model the opinion aware interaction representation x_{ij} as a combination of movie embedding q_j and opinion embedding e_r via a Multi Layer Perceptron (MLP). It can be denoted as g_v to fuse the interaction information with the opinion information, as shown in Fig. 1. The MLP takes the concatenation of movie embedding q_j and its opinion embedding e_r as input. The output of MLP is the opinion aware representation of the interaction between u_i and v_j, x_{ij}, and is represented as follows:

$$\mathrm{x}_{ij} = g_v\left(\{\mathrm{q}_j \oplus \mathrm{e}_r\}\right) \quad (2)$$

Where $\oplus$ represents the concatenation operation between two vectors. As a result of the mean operator, we get the element-wise average of the vectors in $\{x_{ij}, \forall a \in \mathrm{C}(i)\}$. As indicated by the following functions, this mean based aggregator is a linear approximation of a localized spectral convolution [29], which may be expressed as follows:

$$\mathrm{h}_i^H = \sigma\left(\mathbf{W}.\left\{\textstyle\sum_{j\in C(i)} \alpha_i\, \mathrm{x}_{ij}\right\}\right) \quad (3)$$

Where α_i is fixed $\frac{1}{|C(i)|}$ for all movies in the aggregator based on the assumption that all interactions contribute equally to the comprehension of the user u_i. To overcome the drawbacks of the mean-based aggregator, which are triggered by attention mechanisms [30], an axiomatic explanation is to tweak α_i to be aware of the target user u_i, i.e., assigning an individualized weight for each $(v_j; u_i)$ pair,

$$\mathrm{h}_i^H = \sigma\left(\mathbf{W}.\left\{\textstyle\sum_{j\in C(i)} \alpha_{ij}\, \mathrm{x}_{ij}\right\}\right) \quad (4)$$

Where α_{ij} denotes the attention weight of movie v_j interaction with the user u_i's movie space latent variable characterizing user u_i's preference from the previous interaction $C(i)$. In particular, we parameterize the movie attention α_{ij} with a two-layer neural network, which we term as the attention network. The input to the attention network is the opinion aware x_{ij} of the interaction and the target user u_i's embedding $\mathbf{p}_i$. Formally,

$$\alpha_{ij}^* = w_2^T . \sigma\left(\mathbf{W}_1 . \left[x_{ij} \oplus \mathbf{p}_i\right] + b_1\right) + b_2 \quad (5)$$

Softmax function is used to normalize, which may be seen as a movie space contribution from user latent variable, which can be expressed in terms of attention weights.

4.1.2. Social aggregation

In fact, user preferences are impacted by their social correlation [31]. In order to include social data in our model to learn hidden variables of users, the social correlation might alter users' behaviors by drawing on the strengths of other users. Therefore, we developed an attention mechanism model to pick representative social correlations to characterize users' social information and then aggregate them. We propose a social space based on user latent variables. User correlation of social neighbours $N(i)$ utilizes the social space based on the user latent variable of u_i, in the following way:

$$\mathbf{h}_i^S = \sigma(\mathbf{W}. AGG_{social}(\{\mathbf{h}_o^H, \forall o \in N(i)\}) + b) \quad (6)$$

where the social neighbours aggregation function is represented as AGG_{social}. An aggregation function of AGG_{social} is the mean operator, which takes the element-wise mean of the vectors in $\{\mathbf{h}_o^H, \forall o \in N(i)\}$ which is provided by the following function. AGG_{social} also has several natural aggregation functions.

$$\mathbf{h}_i^S = \sigma\left(\mathbf{W}. \left\{\sum_{o \in N(i)} \beta_i \, \mathbf{h}_o^H\right\} + \mathbf{b}\right) \quad (7)$$

where β_i is fixed to $\frac{1}{|N(i)|}$. It is predicated on the assumption that all neighbours contribute equally to the representation of the user u_i. However, as mentioned, strong and weak draws are mixed together. Therefore, we implement an attention mechanism using a two-layer neural network to extract these users who are important to influence u_i by correlating social attention and modeling their correlation strength. Thus, we achieve an attention mechanism with a two-layer neural network to extract these users who are important to influence u_i, and model their correlation strengths by relating social attention β_{io} with $\mathbf{h}_o^H$ and the target user embedding $\mathbf{p}_i$, in a social network as follows:

$$\mathbf{h}_i^S = \sigma\left(\mathbf{W}. \sum_{o \in N(i)} \beta_{io} \, \mathbf{h}_o^H + \mathbf{b}\right) \quad (8)$$

$$\beta_{io}^* = w_2^T . \sigma(\mathbf{W}_1 . [\mathbf{h}_o^H \oplus \mathbf{p}_i] + \mathbf{b}_1) + b_2 \quad (9)$$

$$\beta_{io} = \frac{\exp(\beta_{io}^*)}{\sum_{o \in N(i)} \exp(\beta_{io}^*)} \quad (10)$$

Where β_{io} may be thought of as the difference between the users' abilities.

4.1.3. Learning user latent variable

It is important to examine both the movie space and social space depending on user latent variables. To properly explain latent variables, the movie and social aggregate graphs give information about users from various angles. For the final user latent variable representation, before being sent to MLP, we aggregate embedding of two spaces into two latent variables, where the movie space is based on the user latent variable $\mathbf{h}_i^H$ and the social space $\mathbf{h}_i^S$ latent component depends on the user's input. Officially, $\mathbf{h}_i$ is user latent variable and is defined as:

$$c_1^u = \left[\mathbf{h}_i^H + \mathbf{h}_i^S\right] \quad (11)$$

$$\cdots$$

$$\mathbf{h}_i = \sigma(\mathbf{W}_l . c_{l-1}^u + b_l) \quad (12)$$

where hidden layer is denoted by l.

4.2 Movie modeling

Movie modelling is utilized to learn movie latent variable $\mathbf{z}_j$, as illustrated in the right part of Fig. 1. Two type of aggregations are used in movie modeling. 1) User aggregation is used to learn user space based on the movie latent variable $\mathbf{h}_j^U$ and 2) movie-movie aggregation is used to learn movie space based on the movie latent variable $\mathbf{h}_j^V$. Thereafter the two variables are combined to produce the final movie latent variable.

4.2.1. User aggregation

We use a model similar to that used for learning movie space, which is based on user latent variables. We must gather information from the users who have engaged with v_j for each movie v, designated as $B(j)$. Even for the same movie, users may have differing views on the quality of the experience. It's possible to model movie latent components by analyzing the views of other users who have seen the same movie and may give different perspectives on it. Using the main embedding p_t and an MLP, we derive an opinion aware interaction user representation, from the interaction between u_t and v_j. In addition to the facts, we've also included our thoughts:

$$f_{jt} = g_u([p_t \oplus e_r]) \tag{13}$$

For the movie v_j, we suggest a $B(j)$ interaction representation that aggregates viewers' opinions. $\{F_{jt}, \forall t \in B(j)\}$ represents the interaction representation of user knowledge and opinion aggregated as AGG_{users}.

$$\mathbf{h}_j^U = \sigma\left(\mathbf{W}.AGG_{users}\left(\{f_{jt}, \forall t \in B(j)\}\right) + b\right) \tag{14}$$

Furthermore, we employ a two-layer neural attention network using f_{jt} and $\mathbf{q}_j$ as inputs to create an attention mechanism to identify the important weight μ_{it} of users.

$$\mathbf{h}_j^U = \sigma\left(\mathbf{W}.\left\{\sum_{f \in B(j)} \mu_{jt}\, \mathbf{f}_{it}\right\} + b\right) \tag{15}$$

$$\mu_{jt}^* = w_2^T.\sigma\left(\mathbf{W}_1.\left[\mathbf{f}_{it} \oplus \mathbf{q}_j\right] + b_1\right) + b_2 \tag{16}$$

$$\mu_{it} = \frac{\exp\left(\mu_{jt}^*\right)}{\sum_{t \in B_{(j)}} \exp\left(\mu_{jt}^*\right)} \tag{17}$$

We employ the diverging effects of user-movie interactions to capture the user's attention μ_{it} while learning user space, based on the movie latent variables.

4.2.2. Movie-movie aggregation

Since movies are not self-contained and can be similar and related, it is important to extend the movie latent variable for similar movies. Thus, we introduce the movie-movie aggregation operation to learn movie space, based on movie latent variables. More specifically, the movie-movie space based on the movie latent variables of movie v_j, denoted as $\mathbf{h}_j^V$, is to aggregate the user space based on the movie latent variables of movies in v_j's similar or related movies M(j), as follows:

$$\mathbf{h}_j^V = \sigma\left(\mathbf{W}.AGG_{movie-movie}\left(\{\mathbf{h}_k^U, \forall k \in M(j)\}\right) + b\right) \tag{18}$$

Where $AGG_{movie-movie}$ is the aggregation function on the Movie's neighbours' $M(.)$ in the movie-movie graph G_V. Moreover, we also introduce an attention mechanism to distinguish the important weights k_{jk} of related movies through a two-layer neural attention network,

$$\mathbf{h}_j^V = \sigma\left(\mathbf{W}.\left\{\sum_{f \in M(j)} k_{jt}\, \mathbf{h}_k^U\right\} + b\right) \tag{19}$$

$$k_{jt}^* = w_2^T.\sigma\left(\mathbf{W}_1.\left[\mathbf{h}_k^U \oplus \mathbf{q}_j\right] + b_1\right) + b_2 \tag{20}$$

$$k_{jk} = \frac{\exp\left(k_{jk}^*\right)}{\sum_{k \in M_{(j)}} \exp\left(K_{jk}^*\right)} \tag{21}$$

4.2.3. Learning movie latent variable

Given the two separate movie representations (i.e., $\mathbf{h}_j^U$ and $\mathbf{h}_j^V$), we eventually integrate them into a single final movie latent variable using a typical MLP with hidden layers, as shown in the following.

$$c_1^v = \left[\mathbf{h}_j^U + \mathbf{h}_j^V\right] \tag{22}$$

$$\cdots$$

$$\mathbf{z}_j = \sigma(\mathbf{W}_l.c_{l-1}^v + b_l) \tag{23}$$

4.3 Rating prediction

In this section, we propose an AGAT model for the recommendation task by capturing user and movie latent variables. Thereafter, we concatenate them $\left[\mathbf{h}_i \oplus \mathbf{z}_j\right]$ and insert the final representation into the MLP with l layers as follows:

$$\mathbf{g}_1 = \left[\mathbf{h}_i + \mathbf{z}_j\right] \tag{24}$$

$$\cdots$$

$$\mathbf{g}_{l-1} = \sigma(\mathbf{W}_l.\mathbf{g}_{l-1} + b_l) \tag{25}$$

$$r'_{ij} = w^T.\mathbf{g}_{l-1} \tag{26}$$

Where l is the index of a hidden layer and r'_{ij} is the predicated rating. u_i to v_j.

4.4 Model learning

We need an objective function to optimize before we can estimate the parameters of the AGAT model. We'll use the objective function that is often used in this case, since we're focusing on rating predictions:

$$Loss = \frac{1}{2|\mathcal{D}|}\sum_{i,j \in O}(r'_{ij} - r_{ij})^2 \tag{27}$$

where the above, $|\mathcal{D}|$ is the number of observed ratings, and r_{ij} is the ground truth rating assigned by user u_i to Movie v_j. In our implementation, we use the RM-Sprop [32] optimizer instead of the vanilla

SGD to find the best way to solve the problem. In each run, it randomly picks a training instance and applies a negative gradient to all the parameters in the model. The embedding's in our model includes movie embedding $\mathbf{q}_j$, user embedding $\mathbf{p}_i$, and opinion embedding e_r. They have no predetermined starting point, and they are taught together through training. We do not employ one-hot vectors to represent users and movies since their fundamental characteristics are vast and sparse. Light characteristics of high dimensionality may be included in the latent space to train the model. The system's rating scale affects the opinion embedding matrix. The belief embedding matrix consists of four alternative embedding vectors to signify scores in the range of 1, 2, 3, and 4. For example, deep neural network model optimization is plagued by the issue of overfitting. Because of this, we implemented the dropout approach [33]. In the dropout strategy, neurons are removed randomly from the system during training. Only some of the parameters will be updated when you change them. Furthermore, since dropouts are blocked during testing, the whole network is used to make predictions.

5. Experiments

In this section, we evaluate the following: (1) How does the proposed model compare with other state-of-the-art recommendation models? (2) What impact does social aggregation and movie-movie aggregation have on the proposed model?

5.1 Dataset

We analyzed two large datasets of user-movie consumption. The Netflix Prize and MovieLens-20M (ML-20M) are based on movies' aggregated views and opinions. Data with a score of 4 and above is interpreted as implicit feedback and binarized. User records are only kept for those users who have seen four movies or more. With this approach, we intend to replicate a real-world situation to forecast future behavior based on user data. Because these are cold-start users and movies, we did not include them in these two datasets since their inclusion would go beyond the scope of this study. These two dataset' statistics found in the Table 2.

5.2 Parameter setting

The Python module PyTorch5 has been used to create our model. We experimented with d = 8, 16, 32, and 64 to see which embedding size would work best. For 32, 64, 128, and 512 batches, we looked for rates of 0.0005, 0.001, 0.05, and 0.10 k and evaluated them in values ranging from 15 to 100 to use in constructing the movie-movie implicit network. The hidden layer's size was set to the same value as its activation function, and the embedding size was set to ReLU without mentioning it. For fairness, we used three hidden layers for all the neural components. The early stopping model was used, in which we stopped the training if the validation set grew for five consecutive epochs. A Gaussian distribution with a mean and standard deviation equal to zero was used to set the model parameters for all neural network models. The parameters were set up following the relevant articles and then fine-tuned for the baseline algorithms to get the best results.

As part of our experiment, we chose the first 80% of each user's interaction based on their timestamp to build a training set and utilized the remaining 20% for testing. On this scale, 1 denotes a movie that should be recommended, while 0 signifies one that should not be recommended. Additionally, the NDCG@ k (Normalized Discounted Cumulative Gain) measured Assign greater weight to higher grades positive movie. A rule says that the regular class is 0 when all the rank positions surpass k. The HR (Hit Ratio) metric evaluates the ranking of suggestions, and a higher HR shows that the right recommendations are at the top of the list. HR@k and NDCG@k at the $k = 5, 10$, were used to assess all models that were being compared.

5.3 Performance comparison

We compared our AGAT model with other models, including conventional recommendation systems, traditional social recommendation systems, and deep neural network-based recommendation systems. To answer the first question of our research work. We've selected sample starting points for each group, which we will see below in more detail.

- Probabilistic matrix factorization (PMF): [34] introduced PMF, which utilizes user-movie rating matrix based on Gaussian distributions to modeling latent variables of users and movies.
- Social recommendation (SoRec): [5] PMF-based factor analysis method is proposed to solve the problems of data sparseness and poor prediction accuracy by simultaneously utilizing users' social correlation information and rating records.
- It has been proposed that a social regularization (SoReg) model, which designing two social regularization terms to constrain the matrix factorization objective function [20].
- NGCF is a state-of-the-art graph neural network enhanced C.F. This model is suitable for high-

Received: March 8, 2022. Revised: April 7, 2022.

order connectivity modeling in user-movie graphs model [12].

- Deep social relations (DeepSoR): [35]. proposed aa deep neural network based model to learn nonlinear features of each user from social correlations and integrated into PMF for rating prediction.
- Graph convolutional matrix completion (GC-MC): [36] introduced a model that encoder contains a graph convolutional layer that constructs user and movie embeddings by passing messages on a bipartite user-movie interaction graph.
- Graph neural network for social recommendations (GraphRec): [37] introduced GraphRec model that

  Collectively capture interactions and opinions in the user-movie graph. Furthermore, by considering the heterogeneous strength of social correlations to enhance movie recommendation.

NGCF and PMF are the only two CF models that do not consider social information for predicting ratings. All comparison models are shown in table 3 in terms of NDCG and HR on two datasets. If we look at any benchmark tests, we can see that AGAT consistently beats them. Collaboration modelling of correlations and dependencies between movie users and other user-movie interactions is responsible for this improvement. Graphtec, which models the user-movie connection using a graph neural network, follows the performance. This proves the value of using a graph-structured learning architecture to propagate knowledge between users and movies. When it comes to movie representation, graph does not include a movie-movie correlation, but the AGAT model incorporates this connection, which improves movie modelling and thus enhances movie recommendation.

5.4 The impact of aggregation on the AGAT model

In this part, we examine the aggregation impact on the AGAT to answer the second question of our research work, as detailed below:

- Social Aggregation (SA): Eliminate social aggregation's impact to perform of AGAT model (i.e., user social correlation) by setting $\mathbf{h}_i^S = 0$ in Eq. (11).
- Movie-Movie aggregation (MA): Eliminate the movie-movie aggregation's impact (i.e., movie-movie related) by setting $\mathbf{h}_j^V = 0$ in Eq. (22).
- Social/Movie-Movie aggregation (SMA): Eliminating the impact of both social and movie-movie aggregation’s impact by setting $\mathbf{h}_i^s = 0$ in Eq. (11) and $\mathbf{h}_j^V = 0$ in Eq. (22).

Fig. 2 shows the comparative findings for HR@N and NDCG@N. Note that it indicates matching characteristics and also indicates non-matching features. The proposed framework's performance diminishes when the influence of social aggregation (SA) (i.e., user social correlations) is removed. We have a similar observation for AGAT\ MA when the impact of movie-movie aggregation is eliminated. AGAT obtains worse performance than AGAT\SA and AGAT\MA when the impacts of social aggregation and movie-movie (SMA) are eliminated.

Table 2. Statistics of the movie datasets

Dataset	**ML-20M**	**Netflix**
User	138,000	480,189
Movie	27,000	17,770
Rating	20,000,000	100,000,000
Social correlation	2,097,600	7,346,891
The density of social correlation	0.387%	0.2196%

Table 3. Recommendation comparisons in term of NDCG and HR

Dataset	Metric	PMF	SoRec	SoReg	GNCF	DeepSoR	GC-MC	GraphRec	AGAT	Improv.
Netflix	HR@5	0.1865	0.2255	0.2469	0.2449	0.2480	0.2214	<u>0.2477</u>	**0.2518**	1.63%
	HR@10	0.2686	0.3422	0.3494	<u>0.3663</u>	0.3613	0.3431	0.3599	**0.3688**	0.68%
	NDCG@5	0.1638	0.1638	0.1689	0.1736	0.1776	0.1733	<u>0.1779</u>	**0.1891**	5.92%
	NDCG@10	0.1604	0.2051	0.1985	0.2113	<u>0.2128</u>	0.2060	0.2115	**0.2167**	1.80%
Movie Lens	HR@5	0.4489	0.6077	0.6035	0.6085	0.6154	0.6220	<u>0.6448</u>	**0.6816**	5.39%
	HR@10	0.5933	0.7084	0.7189	0.7224	0.7371	0.7520	<u>0.7617</u>	**0.8068**	5.59%
	NDCG@5	0.3256	0.4731	0.4866	0.5005	<u>0.5199</u>	0.4779	0.4814	**0.5339**	2.62%
	NDCG@10	0.3723	0.5089	0.5139	0.5174	0.5199	0.5233	<u>0.5476</u>	**0.5745**	4.68%

Received: March 8, 2022. Revised: April 7, 2022.

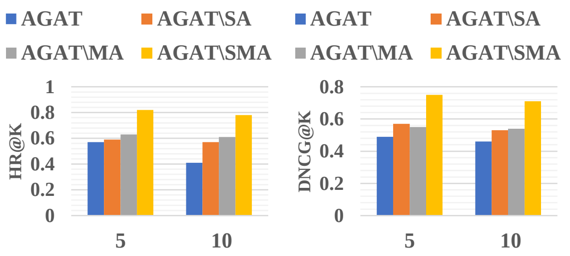

Figure. 2 The Impacts of social aggregation and movie-movie aggregation on AGAT model

6. Conclusions and future work

In this paper, we proposed AGAT for movie recommendation. In particular, different aggregation operations are proposed to model the graph data to learn better user and movie representations. Moreover, our model encodes the users' opinions towards movies to enhance representation learning when modeling the user-movie graph. In addition, we propose to utilize an attention mechanism to capture heterogeneous strengths of social relations when modeling social graphs. Finally, we propose to explicitly incorporate the correlations between movies into our model in order to profile movies. Our experiments show that the correlation of movies helps to improve the performance of our proposed model by nearly 5.92% compared to existing GraphRec models. Comprehensive experiments on two real-world datasets show the effectiveness of our model.

In future research, it will be interesting to see how to use additional auxiliary data to create a movie-movie graph for further enhancing the recommendation model.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Contributions include code formulation, implementation, and programming, in addition to analysis and writing done 1st and reply to reviewers done by 2nd.

Reference

[1] J. Herlocker, J. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations", In: *Proc. of International Conf. on Computer Supported Cooperative Work*, pp. 241-250, 2000.

[2] A. Majid, L. Chen, G. Chen, H. T. Mirza, I. Hussain, and J. Woodward, "A context-aware personalized travel recommendation system based on geotagged social media data mining", *International Journal of Geographical Information Science*, Vol. 27, No. 4, pp. 662-684, 2013.

[3] Z. Xu, L. Chen, and G. Chen, "Topic based context-aware travel recommendation method exploiting geotagged photos", *Neurocomputing* Vol. 155, pp. 99-107, 2015.

[4] T. Ameen, L. Chen, Z. Xu, D. Lyu, and H. Shi, "A convolutional neural network and matrix factorization-based travel location recommendation method using community-contributed geotagged photos", *ISPRS International Journal of Geo-Information*, Vol. 9, No. 8, pp. 464, 2020.

[5] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization", In: *Proc. of International Conf. on Information and Knowledge Management*, pp. 931-940, 2008.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering", In: *Proc. of International Conf. on World Wide Web*, pp. 173-182, 2017.

[7] T. Shanker and A. Barman, "Collaborative recommendation system using dynamic content based filtering, association rule mining and opinion mining", *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 5, pp. 57-66, 2017, doi: 10.22266/ijies2017.1031.07.

[8] Y. Goldberg, "Neural network methods for natural language processing", *Synthesis Lectures on Human Language Technologies*, Vol. 10, No. 1, pp. 1-309, 2017.

[9] S. Khan, H. Rahmani, S. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision", *Synthesis Lectures on Computer Vision*, Vol. 8, No. 1, pp. 1-207, 2018.

[10] Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation", In: *Proc. of International Conf. on World Wide Web*, pp. 417-426, 2019.

[11] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. Yu, "Heterogeneous graph attention network", In: *Proc. of International Conf. on World Wide Web*, pp. 2022-2032, 2019.

[12] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural graph collaborative filtering", In: *Proc. of International Conf. on Research and Development in Information Retrieval*, pp. 165-174, 2019.

[13] R. Bell, Y. Koren, and C. Volinsky, "The bellkor solution to the netflix prize", *KorBell Team's Report to Netflix*, 2007.
[14] M. Soares and P. Viana, "The semantics of movie metadata: enhancing user profiling for hybrid recommendation", In: *Proc. of International Conf. on Information Systems and Technologies*, pp. 328-338, 2017.
[15] B. Konstantinos and T. Giannakopoulos, "Enhanced movie content similarity based on textual, auditory and visual information", *Expert Systems with Applications*, No. 96, pp. 86-102, 2018.
[16] T. Ameen, M. Khalid, A. Abdulqahar, and A. Tariq, "Exploiting Visual Content for Travel Location Recommendation", In: *Proc. of International Conf. on Electrical, Computer and Energy Technologies*, pp. 1-6, 2021.
[17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. S. Thieme, "BPR: Bayesian personalized ranking from implicit feedback", In: *Proc. of Uncertainty in Artificial Intelligence*, p. 2618, 2012.
[18] J. Jiao, X. Zhang, F. Li, and Y. Wang, "A novel learning rate function and its application on the SVD++ recommendation algorithm", *IEEE Access*, No. 8, pp. 14112-14122, 2019.
[19] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, and S. O. Koifman, "Personalized recommendation of social software items based on social relations", In: *Proc. of International Conf. on Recommender Systems*, pp. 53-60, 2009.
[20] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization", In: *Proc. of International Conf. on Web Search and Data Mining*, pp. 287-296, 2011.
[21] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 8, pp. 1633-1647, 2016.
[22] G. Guo, J. Zhang, and N. Y. Smith, "Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings", In: *Proc. of International Conf. on the Conference on Artificial Intelligence*, Vol. 29, No. 1, 2015.
[23] J. Wang, P. Han, Y. Miao, and F. Zhang, "A collaborative filtering algorithm based on svd and trust factor", In: *Proc. of International Conf. on Computer, Network, Communication and Information Systems*, 2019.
[24] M. Zhang, S. Wu, M. Gao, X. Jiang, K. Xu, and L. Wang, "Personalized graph neural networks with attention mechanism for session-aware recommendation", *IEEE Transactions on Knowledge and Data Engineering*, 2020.
[25] Y. Zhao, Y. Wang, J. Zhang, and C. Zong, "Phrase table as recommendation memory for neural machine translation", In: *Proc. of International Joint Conference on Artificial Intelligence*, pp. 1805.09960, 2018.
[26] J. Wu, R. Cai, and H. Wang, "Déjà vu: A contextualized temporal attention mechanism for sequential recommendation", In: *Proc. of International Conf. on Web Conference*, pp. 2199-2209, 2020.
[27] W. Xi, L. Huang, C. Wang, Y. Zheng, and J. Lai, "BPAM: Recommendation Based on BP Neural Network with Attention Mechanism", In: *Proc. of IJCAI*, pp. 3905-3911, 2019.
[28] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, "Irgan: A minimax game for unifying generative and discriminative information retrieval models", In: *Proc. of International Conf. on Research and Development in Information Retrieval*, pp. 515-524, 2017.
[29] Z. Meng, L. Jiao, M. Liang, and F. Zhao, "A lightweight spectral-spatial convolution module for hyperspectral image classification", *IEEE Geoscience and Remote Sensing Letters*, No. 19, pp. 1-5, 2021.
[30] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations", In: *Proc. of International Conf. on World Wide Web*, pp. 1583-1592, 2018.
[31] M. McPherson, L. S. Lovin, and J. Cook, "Birds of a feather: Homophily in social networks", *Annual Review of Sociology*, Vol. 27, No. 1, pp. 415-444, 2001.
[32] T. Tijmen and G. Hinton, "Lecture 6.5-rmsprop, coursera: Neural networks for machine", *Toronto, Technical Report*, No. 6, 2012.
[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929-1958, 2014.
[34] M. Andriy and R. Salakhutdinov, "Probabilistic matrix factorization", In: *Proc. of International Conf. on Advances in Neural Information Processing Systems*, pp. 1-8, 2007.
[35] W. Fan, Q. Li, and M. Cheng, "Deep modeling of social relations for recommendation", In: *Proc. of International Conf. on Artificial Intelligence*, Vol. 32, No. 1, 2018.

[36] R. Berg, T. Kipf, and M. Welling, "Graph convolutional matrix completion", In: *Proc. of International Conference on Software and Computer Applications*, 2017.

[37] W. Fan, Y. Ma, Q. Li, J. Wang, G. Cai, J. Tang, and D. Yin, "A graph neural network framework for social recommendations", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 34, pp. 2033-2047, 2020.