A New Method for Recommendation Based on Embedding Spectral Clustering in Heterogeneous Networks (RESCHet)

Saman Forouzandeh *1, Kamal Berahmand †2, Razieh Sheikhpour ‡3, and Yuefeng Li §2

[1]School of Mathematics and Statistics, University of New South Wales, Sydney, NSW, Australia
[2]Department of Science and Engineering, Queensland University of Technology, Brisbane, Australia
[3]Department of Computer Engineering, Faculty of Engineering, Ardakan University, P.O. Box 184, Ardakan, Iran

February 17, 2023

*s.forouzandeh@unsw.edu.au
†kamal.berahmand@hdr.qut.edu.au
‡rsheikhpour@ardakan.ac.ir
§Yuefeng.Li@hdr.qut.edu.au

Abstract

The advancement in internet technology has enabled the use of increasingly sophisticated data by recommendation systems to enhance their effectiveness. This data is comprised of Heterogeneous Information Networks (HINs) which are composed of multiple nodes and link types. A significant challenge is effectively extracting and incorporating valuable information from HINs. Clustering has been proposed as one of the main methods in recommender systems, but in Heterogeneous Information Networks for recommender systems has received less attention. In this paper, we intend to present a new method for Recommendation Based on Embedding Spectral Clustering in Heterogeneous Networks (RESCHet), which uses the embedding spectral clustering method, whose similarity matrix is generated by a heterogeneous embedding approach. Subsequently, we employed the concepts of submeta-paths and atomic meta-paths to uncover the relationships between users and items that are pertinent to each cluster. Finally, we generated recommendations for users by computing the Hadamard product between the relevant vectors. Experiments carried out on three open benchmark datasets have demonstrated that RESCHet outperforms current leading methods in a significant manner.

Keywords: Recommender Systems, Heterogeneous Information Networks, Network Embedding, Spectral Clustering, Hadamard product

1 Introduction

In recent years, due to the explosion of information, the classification and personalization of information have become one of the most critical challenges for users. Therefore, filtering information based on the user's personal needs and in line with their interests is particularly important. In this regard, recommender systems are prevalent, filtering information, finding users' interests and preferences, and then generating suggestions based on this. Most traditional recommendation systems analyze and provide recommendations based on users' historical behavior, such as through matrix factorization (e.g., user-item rating matrices), concentrating on a single user-item interaction record, and not considering all of the auxiliary data that helps produce recommendations. Regardless, due to its variability and complexity, it is typically challenging to model this supplemental information in recommender systems.

Researchers have become more interested in graph-based recommender systems as the popularity of social networks has grown. These recommender systems pull out helpful information, which is used on homogeneous graphs based on the relationship between a user and an item or between a user and another user. Accordingly, they can generate various recommendations. Nevertheless, in the real world, we usually have to deal with heterogeneous graphs, where millions of interactions between different entities may exist, and traditional methods cannot be used to find similarities between different types of nodes. Currently, there are several recommendation algorithms that have been specifically designed for modeling heterogeneous information networks, which can be broadly grouped into three main categories: similarity measurement, matrix factorization, and graph representation learning [Liu et al., 2022a]. Similarity measurement methods focus on identifying the similarity between different items or users based on the network structure and semantic information. Matrix factorization methods aim to factorize the information in the network into a low-dimensional latent space. Graph representation learning methods, on the other hand, focus on learning the representations of nodes in the network to capture the structural and semantic information of the network.

Classical recommendation techniques, such as SVD and MF, often cannot indicate the complexity of user-item preferences due to the simplicity and sparsity of the data. These methods often fail to fully capture the underlying latent feature representations of users and items in real-world datasets [Pham et al., 2023]. Extracting and utilizing relevant information from HIN is a significant challenge. In this regard, one popular approach for recommendations is network embedding, which can effectively extract useful information from HINs to make recommendations. Compared to homogeneous information network embedding, HIN embedding can more thoroughly and accurately capture the semantic and structural information of the network as reported in studies such as [Hou et al., 2019, Shi et al., 2016, Shi et al., 2019, Shi et al., 2018b]. As a result, many recommendation models have been developed in recent years using HIN embedding. Examples include [Wang et al., 2019] which combines HIN embedding and recommendation models for personalized recommendations, and [Guo et al., 2018] which applies HIN embedding to social network recommendations.

Clustering in heterogeneous information networks (HINs) is a technique used to group similar nodes or items based on the network structure and semantic information, making it easier for users to find relevant information and for the system to make accurate recommendations [Wang et al., 2022a, Guo and Dai, 2022]. Therefore, clustering is a crucial topic in the field of graph analysis. Various clustering techniques have been employed in homogeneous graphs by researchers to partition the vertices into multiple groups or communities. Previous studies have primarily focused on utilizing deep learning approaches to learn latent representations of nodes, which are then utilized by simple clustering methods such as k-means. In recent years, researchers have turned towards deep learning techniques to learn compact representations or embeddings to extract rich information from graph data [Shen et al., 2018, Gao et al., 2018]. These embeddings are then used with simple clustering algorithms such as k-means to obtain the final clustering result [Wang et al., 2022a]. The autoencoder, which consists of an encoder that encodes input data into a low-dimensional space and a decoder that reconstructs the input data from the encoded embedding, is a popular solution for this type of embedding-based approach [Gao et al., 2018, Tian et al., 2014]. Additionally, researchers have considered the similarity of an autoencoder and spectral clustering, and have learned a latent representation for clustering through sparse autoencoder [Tian et al., 2014]. Other recent developments, such as GAE and VGAE, [Kipf and Welling, 2016] based on graph convolutional networks (GCN), can also be adapted for node clustering analysis. Clustering in homogeneous networks has been widely studied; however, the task becomes more challenging in heterogeneous networks due to the presence of different types of entities and hidden semantic relationships between nodes. In this context, the identification of different types of nodes is crucial in order to group similar nodes in a cluster. To the best of our knowledge; this research represents the first investigation of embedding spectral in heterogeneous graphs based on recommendation systems.

The results of various studies have shown that combining HIN embedding with recommendation models can significantly improve performance. Recent research on heterogeneous networks has shown that they can make recommendations work better because they combine different pieces of information well [Liu et al., 2023]. This makes it easier for users to get different recommendations. Most existing methods for making recommendations based on heterogeneous networks work on the idea that meta-paths can be used to measure the semantic relatedness between users and items. However, the use of recommender systems is facing many problems. In a HIN, finding and calculating the similarity between nodes is obtained by meta-path. The meta-path may find hundreds of communication links among the nodes based on their similarity. But not all connections are important to us, and many of them are hidden, so the recommender system can't make a good suggestion. This research endeavors to address the challenges associated with clustering in heterogeneous networks by utilizing heterogeneous information network embedding. The ultimate goal of this research is to cluster users in a heterogeneous graph and subsequently provide personalized recommendations to members of each cluster of users.

To achieve this goal, this study is divided into two main sections. A novel framework for Recommendation based on Embedding Spectral Clustering in Heterogeneous Networks (RESCHet) is proposed, which is divided into two main parts. In the first part, clusters from the HIN are obtained. In order to do this, Metapath2vec has been utilized to embed HIN, in which the reduced vector for each user is created by removing nodes that are not of the same type and integrating the sequence using a non-linear method. A high-accuracy similarity matrix is produced from users based on the cosine similarity between these vectors. In the following, with a simple spectral clustering method, the clusters that contain the most similarity between users are obtained. In the second part of the study, recommendations are generated for users based on the cluster members and the Hadamard product, and the atomic meta-paths are defined based on the cluster members. The accuracy of these recommendations was compared to that of similar methods from previous research, and the findings indicated an increase in accuracy, highlighting the superiority of the proposed method over comparable approaches. The major contributions can be summarized as follows:

1. This study presents the RESCHet method, which, to the best of our knowledge, is the first to incorporate embedding spectral clustering in the context of heterogeneous information network-based recommender systems.

2. A novel approach for spectral clustering in Heterogeneous Information Networks (HINs) is proposed, utilizing cosine similarity between two integrations of embedding vectors to generate the similarity matrix.

3. The recommendations given to users are based on the clustering of users and the relationships between users and items embedding vectors within those clusters.

4. To determine the efficacy of the RESCHet method, experiments were carried out using three real-world datasets. The evaluation of these experiments revealed that the RESCHet approach is not only efficient but also surpasses the performance of other state-of-the-art baseline methods.

The organization of the rest of the paper is as follows: Section 2 presents a review of related works, encompassing both recommender systems and Recommendation Systems Based on Heterogeneous Information Networks (HIN). In Section 3, we provide an overview of the preliminary concepts and notations utilized in this study. This is followed by a comprehensive description of our methodology in Section 4, which encompasses Heterogeneous Network Embedding-based Recommendation, Embedding Spectral Clustering, and the Recommendation Model. The experimental evaluation of the proposed RESCHet method, along with its comparison to baseline methods, is discussed in Section 5. Finally, the paper concludes with a summary of the key findings and future directions for research in Section 6.

2 Related Works

Researchers use recommender systems to filter information and find hidden knowledge in it. There is a lot of data and details in different sciences, and using recommender systems helps solve the problem of too much data. So, in the first part, we looked at the ideas behind recommender systems and how they work. In the second section, we looked at related work on recommendation systems based on heterogeneous information networks (HIN).

2.1 Recommendation Systems

Information overload happens when there is too much information about too many products. This bothers customers and buyers. In this way, recommender systems, whose job is to find products based on user's interests and preferences, have come to the aid of buyers. Because their purchase offer is based on users' interests, they can choose products that are good for them from the many items available in online networks. A sound recommender system gives customers valuable information based on their wants and needs. On the one hand, the following strategies could effectively alleviate consumers' information overload: [Koren et al., 2022, Polatidis and Georgiadis, 2016, Bobadilla et al., 2011, Forouzandeh et al., 2015]. Earlier works on recommender systems relied heavily on collaborative filtering (CF) techniques to generate various recommendations based on the recommendation's historical interactions [Wu et al., 2022]. Some problems with these strategies are cold starts and a lack of data, and many studies have been done to improve the performance of recommendations by using more data.

Matrix factorization is used by a lot of traditional recommendation methods to solve the "cold start" problem and predict missing ratings. The prediction techniques used in Collaborative Filtering (CF) rely on a low-dimensional user matrix and an item matrix to interpret the rating matrix. However, these techniques often suffer from sparsity and scalability issues, as noted in [Zhao et al., 2020]. To address these issues, researchers have explored various approaches to improve the performance of CF. One such approach is the "Neighborhood-Enriched Contrastive Learning (NCL)" method, which is proposed in [Lin et al., 2022]. This approach leverages a contrastive learning framework and explicitly incorporates the potential neighbors of users or items from both the graph structure and semantic space. The authors demonstrate the effectiveness of NCL in improving CF performance.

The paper "Resolving Cold Start and Data Sparsity Problems in Recommender Systems using Linked Open Data" [Natarajan et al., 2020] proposes a solution to two common challenges in recommender systems: the cold start problem and the data sparsity problem. To tackle the cold start problem, the authors develop a Recommender System with Linked Open Data (RS-LOD) model. To address the data sparsity issue, they introduce a Matrix Factorization with Linked Open Data (MF-LOD) model. According to a subsequent study by the authors [Duan et al., 2022], the sparsity of the rating matrix, caused by limited rating data, has a significant impact on the performance of collaborative filtering for recommendation systems. Two important research strategies that could help solve the problem are to improve the recommendation algorithm and add side information. This paper's researchers, who integrated these two methodologies, proposed the review-based matrix factorization method. The use of graph neural networks (GNU) in recommender systems has recently become prevalent, and much research has been done to improve it [Wu et al., 2022, Gao et al., 2022a, Gao et al., 2022b, Liu et al., 2022b, Da'u et al., 2021].

2.2 Recommendation Systems Based on Heterogeneous Information Networks (HIN)

In a general classification for graphs, there are two types: homogeneous and heterogeneous graphs. Homogeneous graphs, objects, and the links (nodes and edges) between them are all the same. For example, social network graphs show different types of social relationships and user friendships. Considerable researchers are interested in social recommender systems because they can predict friendship links between users based on homogeneous graphs, calculate users' trust, or form a user-user matrix and make various recommendations based on that [Yang et al., 2014, Forouzandeh et al., 2021, Palomares et al., 2021, Walker et al., 2022]. A heterogeneous information network (HIN) is a heterogeneous graph in which nodes and edges are not of the same type and there are no special rules for communication between nodes [Xu et al., 2019, Lu et al., 2020, Wang et al., 2020].

In other words, the real world comprises many objects and components and interactions between them that form a HIN. It contains all kinds of social networks, online shopping, etc., and you will get rich semantic and structural information by analyzing and studying the interactions between its components. So, the HIN analysis is fundamental and is used extensively in searches for similarity, classification, recommendation, etc [He et al., 2021]. The older technique for analyzing homogeneous graphs, such as cosine similarity, Jaccard coefficient, or Euclidean distance with nodes and edges, cannot be used directly on HINs. However, having multiple meta-paths in HIN contains numerous semantic information that will result in various similarities. As a result, the HINs' extensive semantic and structural information can improve the accuracy of the measure of similarity used to compare items [Lu et al., 2020, He et al., 2021]. HIN learning aims to obtain the latent vertex representations by mapping vertices into a low-dimensional space, which can then be leveraged for recommendations[Jiang et al., 2018, Shi et al., 2018a, Huang et al., 2022, Wang et al., 2022b]. Meta-paths, an essential component of a HIN, are widely utilized to calculate the similarity of entities in the HIN. Various studies are currently focusing on enhancing recommender system performance using HIN.

HetNERec, a recommendation approach based on heterogeneous network embedding, was proposed by the paper's authors [Zhao et al., 2020]. They created a heterogeneous co-occurrence network to discover the hidden representations of users and items. Then, to provide recommendations, the several representations of each node are combined into a single representation and integrated utilizing matrix factorization. For explainable recommendations, researchers in work [Ai et al., 2018] suggest explaining knowledge-based embeddings. They propose a knowledge-based representation learning framework to integrate heterogeneous entities for the recommendation. The paper [Shi et al., 2018a] shows a method to use extra information in HINs to make recommendations based on HIN embedding. To get more meaningful node sequences for network embedding, they came up with a random walk strategy based on meta-paths. Since the meanings of embeddings based on different meta-paths are different, the learned embeddings were added to an extended matrix factorization.

This paper's authors propose a recommendation system based on unified embeddings of behavior and knowledge features. They find out what users like by looking at their past actions and knowledge graphs. This lets them make more accurate and varied recommendations for users. In the second part of the proposed method, spectral clustering is used. Since clustering is done based on the HIN embeddings, some research from the past is reviewed. For instance, research [Fang et al., 2020], the community detection method is based on the meta-path concept, which is the order of edges between different types of vertices, and the community's cohesiveness is calculated. Another two techniques [Zhou et al., 2020, Li et al., 2020b] for clustering objects in attributed heterogeneous information networks (AHIN) have been devised. The former, called CMOC-AHIN, use attribute data and multi-type node clustering to cluster objects of different types. At the same time, the latter provided a semi-supervised method (SCHAIN-IRAM) based on the similarity of objects that considers both the objects' qualities and their structural connection. In order to solve personalization and recommendation issues at scale, the research [Satuluri et al., 2020] describes a framework called SimClusters based on identifying bipartite communities from the user-user graph and using them as a representation space. This paper [Chen et al., 2018] uses evolutionary clustering and collaborative filtering to rate prediction items. An evolutionary clustering algorithm is generated for heterogeneous network nodes. Individuals with similar stable states are put into the same cluster, and user-based collaborative filtering methods are proposed for each cluster.

3 PRELIMINARIES

In this section, we will formally introduce the basic concepts of HIN and illustrate the symbols used in this paper. Then we will elaborate on the unique challenges HIN brought by heterogeneity compared with homogeneous graphs.

3.1 Definition 1. Heterogeneous Information Network (HIN)

A Heterogeneous Information Network (HIN) is a type of graph represented as $G = V, E$, where V represents the set of nodes and E represents the set of links. The mapping functions $\phi(v) : V \to A$ and $\psi : E \to R$ associate each node $v \in V$ and each link $e \in E$ with their respective sets of node types, denoted as

A, and link types, denoted as R. The number of nodes and link types must be greater than 2, meaning $A + R > 2$. HINs contain nodes and edges of multiple types, whereas networks with only one type for both nodes and edges are known as Homogeneous Information Networks. The objective of heterogeneous graph embedding is to map the nodes in the HIN into a lower-dimensional space using a function $\Phi : V \rightarrow R^d$ such that $d << |V|$. In a heterogeneous graph, nodes and edges have different relationships, and each association has a unique meaning and interpretation.

3.2 Definition 2. Heterogeneous Network Embedding

In the context of a heterogeneous network represented as $G = (V, E, T)$, the objective of network graph embedding is to map each node in V to a low-dimensional vector representation in a Euclidean space, where d is much smaller than $|V|$ ($d \ll |V|$), and preserve the rich semantic information present in the heterogeneous network.

3.3 Definition 3. Network schema

A network schema, represented as $S = \{A, R\}$, acts as a meta template for a heterogeneous information network (HIN) $G = \{V, E\}$, with mapping functions ϕ and ψ assigned to nodes and links respectively. In a HIN, there exist multiple entities connected through diverse semantic connections, known as meta-paths, as described in works such as [Li et al., 2020a, Shi et al., 2018a].

3.4 Definition 4. Meta-Path

A meta-path $p = A_0 \xrightarrow{R_1} A_1 \xrightarrow{R_2} ... \xrightarrow{R_k} A_k$ is a path that is defined on the graph of the network schema $G_T = (A, R)$. This definition creates a new composite relation $R_1R_2...R_k$ between type A_0 and A_k, where A_i belongs to set A and R_i belongs to set R for $i = 0, 1, ..., k$. A meta-path is a sequence of relations that connect object pairs in a HIN, and it can be utilized to extract features of connectivity within the graph (as discussed in references [Wang et al., 2022b, Cai et al., 2018]).

3.5 Definition 5. Metapath2vec

Metapath2vec is a node embedding method for a heterogeneous information network (HIN) proposed by Yuxiao Dong in 2017. It is a modification of node2vec for heterogeneous networks. Metapath2vec uses extra information about node types to give an alternative method, known as "meta-path-based random walk," instead of utilizing a random walk scheme with explicit p and q parameters. It includes two main parts; first, construct a node sequence for each vertex based on random walks of meta-paths, Using Equ. 1:

$$p(v^{i+1}|v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}|v_t^i) \in E, \phi(v^{i+1}) = t+1 \\ 0 & (v^{i+1}|v_t^i) \in E, \phi(v^{i+1}) \neq t+1 \\ 0 & (v^{i+1}|v_t^i) \notin E \end{cases} \tag{1}$$

Whereas in (1) and in the second step, skip-gram produces an embedded vector from the sequence of generated nodes, which uses Equ. 3:

$$\arg\max_{\theta} \sum_{v\in V} \sum_{t\in T_v} \sum_{c_t\in N_t(v)} \log \surd(c_t|v\ ;\theta) \tag{2}$$

Whereas (2), in which T_V denotes the set of node types, $N_t(v)$ represents the neighborhood of node v, and $(c_t|v\ ;\theta)$ is typically defined as a softmax-like function.

3.6 Definition 6. Cosine Similarity

Cosine similarity measures the similarity between two nodes by calculating the number of common neighbors between them. Specifically, it is expressed by dividing the number of common neighbors of node i and node j by the geometric mean of their degrees [24], which is calculated as follows:

$$S_{ij} = \frac{V_i^T V_j}{||V_i||.||V_j||} \tag{3}$$

where in (3), v_i, v_J are the vectors corresponding to nodes V_i and V_j in the adjacency matrix A.

3.7 Definition 7. Spectral clustering

Spectral clustering, which uses a non-linear approach, is one of the most efficient clustering algorithms. There are three phases in spectral clustering:

1. Compute a similarity matrix: This stage creates a similarity matrix that indicates how similar two nodes are to each other.

2. Project the data onto a low-dimensional space: in this stage, we compute the matrix Laplacian, which is just another matrix representation of data and can be useful in finding interesting data properties. This can be computed as:

$$L = D - A \tag{4}$$

   Where (4) A represents the adjacency matrix and D represents the degree matrix, as well as in the (5) as follows:

$$d_i = \sum_{\{j|(i,j)\in E\}} w_{ij} \tag{5}$$

   Thus, (6) includes,

$$L_{ij} = \begin{cases} d_i & if \quad i = j \\ -w_{ij} & if (i,j) \in E \\ 0 & if (i,j) \notin E \end{cases} \tag{6}$$

3. Create clusters: in the last stage, we compute the k largest eigenvectors of the Laplacian matrix to define a feature vector for each data point, then cluster the data into k clusters via traditional clustering methods such as k-means.

3.8 Definition 8. HIN-based recommendation

A Heterogeneous Information Network (HIN) $G = \{V, E\}$ can model various types of information in a recommender system. In recommendation-based HINs, there are two types of entities, i.e. users and items, and their connections. Let $U \subset V$ and $I \subset V$ represent the sets of users and items respectively, a triplet $< u, i, r_{u,i} >$ signifies that a user u has assigned a rating of $r_{u,i}$ to an item i, and $R = < u, i, r_{u,i} >$ indicates the set of all rating records. U, I, and R are subsets of V and E respectively. The goal of the HIN $G = \{V, E\}$ is to predict the rating score $r_{u,i'}$ of a user $u \in U$ for an unrated item $i' \in I$.

4 Methodology

In this section, we describe our proposed method's design in detail, which is basically composed of three parts. The first is a heterogeneous network embedding-based recommendation method that uses meta-paths to learn the embedding representation of users, and it is shown in Fig. 1. Subsequently, the user cluster is created by embedding spectral clustering on heterogeneous network users. We show an overall schematic of embedding spectral clustering in Fig. 2. Finally, a framework system recommender is proposed based on the discovered clusters. The general schematic of our method is shown in Fig. 1.

4.1 Heterogeneous Network Embedding-based Recommendation

We utilize the Metapath2vec method for embedding nodes, and when we use the random walk to generate the sequence, there will be several types of nodes, and they each have unique attribute features. The nodes that are different from the starting node type must be removed to generate a walk sequence. As illustrated in Fig. 3, we only consider meta-paths if the start type is a user type to learn efficient representations for users. By removing nodes that are not of the same type as U_1, we can derive some meta-paths $U_1, U_2, U_3, U_4, ...$, such as the random walk sequence $U_1\, M_1\, U_2\, M_2\, U_3\, M_4\, U_4...$. The remaining sequences can now be considered a homogeneous network, and the embedding representation of the nodes can be learned using the homogeneous network embedding approach. Therefore, we could achieve the embedding of nodes on various meta-paths with e_u^l and $|P_u| = 1$, in which P denotes the set of meta-paths, $|P_u|$ is the number of the meta-path sets, and e_u^l and represents the nodes embedding on the lth meta-path.

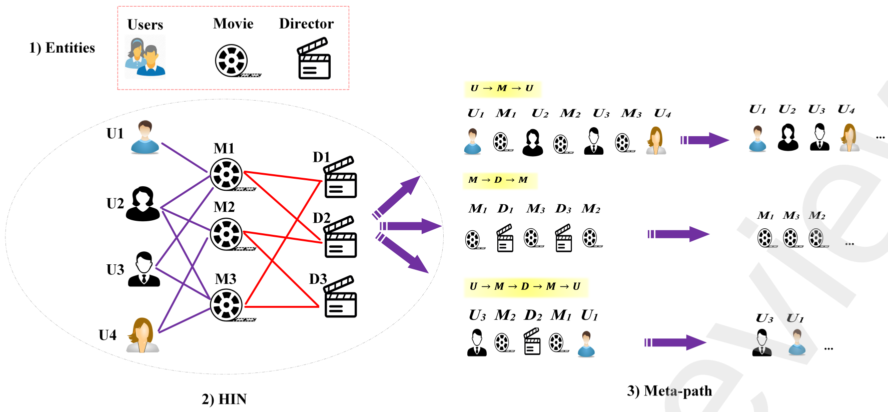

Figure 1: An illustration of constructing a Meta-path based random walk from a heterogeneous information network.

A node can be represented in different vector spaces since it has various meta-paths. These vectors are processed using the fusion function to enable later integration with the spectral clustering model. There are different fusion functions, including linear and nonlinear functions. Linear functions that cannot model the complex structure have low performance, so we utilize nonlinear functions to enhance the effect of the function, using the following objective function:

$$g(\{e_u^{(l)}\}) = \sigma(\sum_{l=1}^{|P_u|} w_u^l \, \sigma(M^l e_u^{(l)} + b^l) \tag{7}$$

where (7), in which σ is a nonlinear function that generally uses a sigmoid function, $|P|$ is the total number of meta-paths, $M^l \in \mathbb{R}^{D\times d}$ is the transformation matrix, and $b^l \in \mathbb{R}^D$ is the bias vector of the lth meta-path.

4.2 Embedding Spectral Clustering

Due to its superior performance in finding non-convex clusters among the several algorithms proposed to perform graph clustering, spectral clustering is becoming more popular. One of the most important challenges is the definition of the similarity matrix. The similarity matrix directly affects spectral clustering results because it preserves the relationship between objects. We create a similarity matrix based on the reduction vectors produced by the embedding approach to cluster users in the heterogeneous network. Fig. 2 details it in the embedding spectrum clustering process. The first step is to obtain a reduced vector for each user; the embedded vector is generated using the Metapath2vec method in Equ. (4), and the second is to construct a matrix of similarity based on the similarity between the two reduced vectors in Equ. (4), the third is to create a Laplacian matrix from the similarity matrix using Equ. (4). The Eigengap technique is then used to select a set from k to the smallest eigenvalues in the fifth phase. Eigngap is a specific method for estimating the quantity of k or (connected components). In this case, k is chosen so that all of the eigenvalues k_1,..., k_m—are small while $k_(m+1)$ is relatively large. Actually, Eigngap determined the variation between two successive eigenvalues. The choice of k that maximizes the differential expression generally results in the most stable clustering. The Laplacian matrix is translated into a space with lower dimensions and more information by choosing k eigenvalues. The users are better described in the newly transferred space. The final stage involves applying K-means clustering to the newly created space of data that contains more relevant information. This process's output includes the clusters with the most similarities among users.

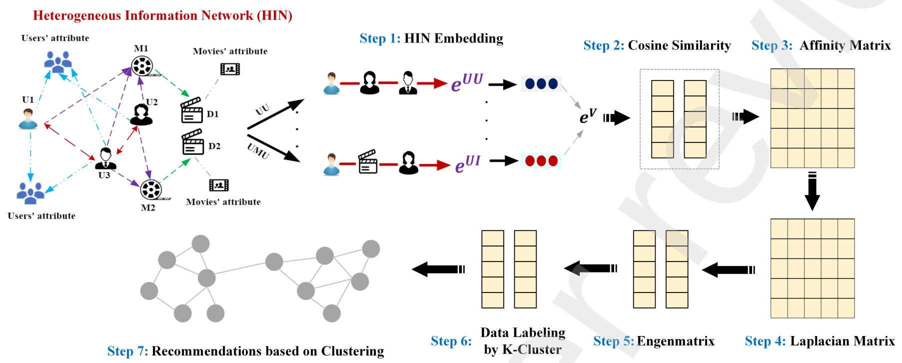

Figure 2: Model framework of Embedding Spectral Clustering on HIN

4.3 Recommendation Model

In this section, we explore the application of spectral clustering to generate personalized recommendations for users in a heterogeneous network. Unlike in homogeneous networks, where the similarity between users can be calculated using functions based on distance, such as Euclidean or Cosine, obtaining the similarity between users in a heterogeneous graph presents a challenge. The complexity of connections between entities in the Heterogeneous Information Network (HIN) further exacerbates the situation, as there are multiple relationships between nodes and it is not possible to discover all of them. To address this challenge, we resort to the use of meta-paths. However, using meta-paths alone to provide recommendations based on the distance between users in each cluster presents another challenge, as the meta-paths are developed based on all entities in the heterogeneous network and do not consider the specific clusters or users in question. In the following, we present potential solutions to these problems and describe the recommendation model in detail, as depicted in Fig. 3.

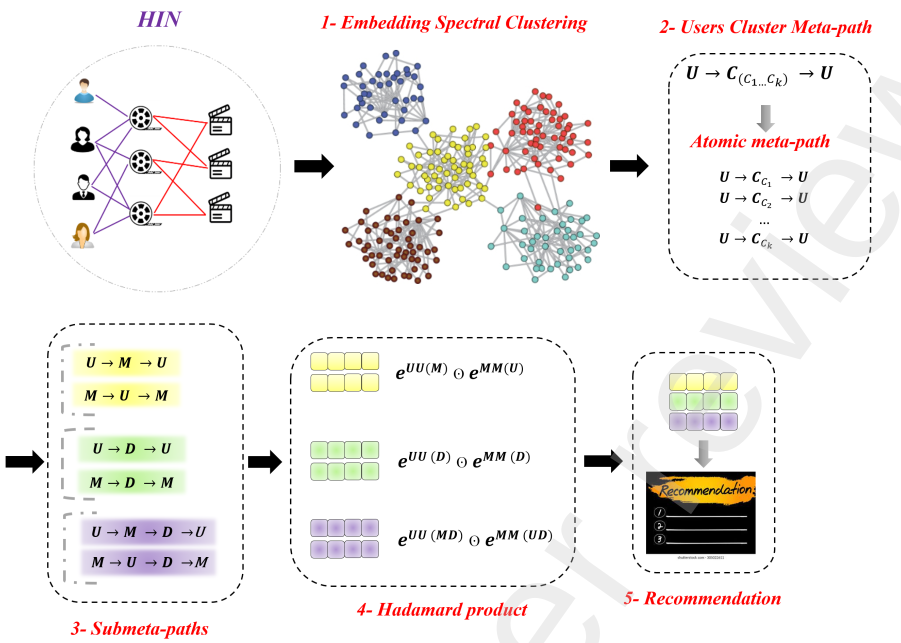

Figure 3: Final Recommendation Framework on HIN using User Clusters and Atomic Meta-path method.

As you can see, there is a HIN at the beginning of the model, and embedding spectral clustering has been done based on the users in the first stage; we have several clusters, and in each cluster, there are several users based on behavioral similarity. In the second step, in order to be able to limit the meta-paths based on the clusters' members, two types of restricted meta-paths are defined and used. According to these studies [Shi et al., 2015, Liu et al., 2022a, Liu et al., 2023], they used group meta-paths ($User - Group(UG)$) that indicate the users belong to the same groups in the social networks. As a result, we defined the cluster meta-path ($User - Cluster - User(UCU)$), which indicates that the users belong to one or more clusters. But, existing path-based similarity measurements cannot address this type of path. The study [Shi et al., 2015] proposed the atomic meta-path instead of presenting an ad hoc similarity measure and proposed a general approach to make current path-based similarity measures useful to meet the variable. Accordingly, the weighted meta-path can be decomposed into a group of atomic meta-paths with specific attribute value constraints. These atomic meta-paths can then be directly utilized with current path-based similarity measures; a weighted meta-path is a collection of atomic meta-paths that satisfy constraint clusters in which $C_1, ..., C_k$ are embedding spectral clusters created in the HIN. Therefore, $U \rightarrow C_{C_1,...,C_k} \rightarrow U$ indicate the clustering of users and $U \rightarrow C_{C_1} \rightarrow U$, ..., $U \rightarrow C_{C_k} \rightarrow U$ displays the atomic meta-paths in the HIN based on the number of the embedding spectral clusters.

In the third step, we created submeta-paths based on each cluster, which we want to use to find hidden dependencies and relations between users and items. We have six submeta-paths (three pairs), and they are used in pairs with each other and include the following:

1. $[U \rightarrow M \rightarrow U$, and $M \rightarrow U \rightarrow M]$.
2. $[U \rightarrow D \rightarrow U$, and $M \rightarrow D \rightarrow M]$.
3. $[U \rightarrow M \rightarrow D \rightarrow U$, and $M \rightarrow D \rightarrow U \rightarrow M]$.

As you can see, the start and end points of each pair of submeta-paths (Movies or Users) are the same, and we can use these submeta-paths to discover different modes of communication and hidden dependencies between nodes in the HIN. For instance, the submeta-paths of the first pair return two types of relationships between nodes, which are as follows:

- $U \rightarrow M \rightarrow U$: This submeta-path returns relationships between users who have viewed and rated the same movies.
- $M \rightarrow U \rightarrow M$: This submeta-path shows the relationships between movies that have been viewed and rated by the same users.

In the fourth phase, the results from each submeta-path are transformed into vectors for embedding, and to use the found relations in each pair of submeta-paths, a Hadamard product (an element-wise product) is used between embedded vectors, and the strategy is formulated as follows:

$$E_{UM} = e^{UU(M)} \odot e^{MM(U)} \tag{8}$$

The result of (8) is a vector containing relationships between nodes in each cluster based on UMU and MUM.

$$E_{D} = e^{UU(D)} \odot e^{MM(D)} \tag{9}$$

The result of (9) is a vector containing relationships between nodes in each cluster based on UDU and MDM, which shows relations between movies based on the director nodes and users based on the director nodes.

$$E_{UMD} = e^{UU(MD)} \odot e^{MM(UD)} \tag{10}$$

The result of (10) is a vector containing relationships between nodes in each cluster based on $UMDU$ and $MDUM$, which shows relations between movies based on the director nodes and users based on the director nodes. Ultimately, in the fifth step, we have three vectors resulting from the fourth step. These vectors show the relationships between users and movies based on three states. Based on these three vectors, we can generate and provide users with various recommendations. The result of (10) is a vector containing relationships between nodes in each cluster based on $UMDU$ and $MDUM$, which shows relations between movies based on the director nodes and users based on the director nodes. Ultimately, in the fifth step, we have three vectors resulting from the fourth step. These vectors show the relationships between users and movies based on three states. Based on these three vectors, we can generate and provide users with various recommendations. Accordingly, we used the Equ. (3) to calculate the similarity between two users based on the obtained embedding vectors. The general function of RESCHet is described in Algorithm 1.

Algorithm 1 The overall process of RESCHet

Input: The heterogeneous Network $G^U = \{(V_{I,U}, E_{I,U})\}$ and $G^I = \{(V_{k,I}, E_{k,I})\}$
Output: The set of representation embedding vectors for users and items $emb(e^{UU})$ and $emb(e^{UI})$, Clusters $C_1, C_2, ..., C_k$.
for (**do**$l = 1$) to $|G^U|$
  for (**do**U_1, U_2) in $emb(e^{UU})$ do Learning users' embeddings according to the Metapath2vec
    for (**do**U_1, I_2) in $emb(e^{UI})$ do Learning items' embeddings according to the Metapath2vec
    end for
  end for
end for
The integration of the learned network embeddings into a unified form through the non-linear function.
Compute the similarity matrix $A \in R^{n \times n}$ based on Cosine similarity between vectors
Embedding Spectral Clustering (D, k)
Generate the submeta-path and atomic meta-path based on each cluster
Hadamard product according to the Equ. (8), (9), and (10)
Providing recommendations based on the users' similarity according to the Equ.(3)

5 Experiments

This section discusses the results of the $RESCHet$ experiment. To completely assess the performance of the proposed approach, a number of experiments were conducted. The organization's structure is listed as follows: The datasets used in the following investigations are reviewed in Section 5.1. Evaluation metrics are discussed in Section 5.2. Nine well-known and cutting-edge comparison techniques are examined in Section 5.3, including PMF, FM_{HIN}, $SoMF$, $HeteMF$, $SemRec$, DSR, $HERec$, $HetNERec$, and $HopRec$. The efficiency of the method is assessed on three types of datasets in Section 5.4. The Python programming language is used to implement the comparison algorithms, which are then put into action on a computer with an Intel Core i7 processor and 16 GB of RAM.

5.1 Datasets and preprocessing

We adopted three commonly used data sets according to the papers [He et al., 2021, Zhao et al., 2020, Shi et al., 2018a] that have been widely used to make recommendations as follows:

1. The Douban Movie dataset comprises 13,367 users and 12,677 films, with a total of 1,068,278 movie ratings that range from 1 to 5. In addition to this, the dataset encompasses information about the social relationships and attributes of both users and films.

2. The Douban Book dataset is comprised of 13,024 users and 22,347 books, with a total of 792,026 ratings that fall within a range of 1 to 5.

3. The Yelp dataset encompasses the ratings of 16,239 users and 14,282 local businesses, with a total of 198,397 ratings ranging from 1 to 5. Furthermore, the dataset incorporates information about the social relationships and attributes of the businesses.

5.2 Evaluation Parameters

In this section, the metrics that are used to evaluate the research's method and compare it to other methods are explained, and the following metrics are used.

$$\text{MAE} = \frac{1}{|R_{test}|} \sum_{i,j \in R_{test}} |r_{i,j} - \acute{r}_{i,j}| \tag{11}$$

$$\text{RMSE} = \sqrt{\frac{1}{|R_{test}|} \sum_{i,j \in R_{test}} |r_{i,j} - \acute{r}_{i,j}|^2} \tag{12}$$

where in (11) and (12) the MAE and RMSE, N represents the whole number of ratings, $P_{u,i}$ denotes the predicted rating of user u given on item i, and $r_{u,i}$ denotes the actual rating of user u gave on item i.

5.3 Compared baseline algorithms

The following models are related to baseline algorithms, and we compare our method with the following models:

1. FM_{HIN}: The Context-Aware Factorization Machine decomposes the user-item rating matrix into two low-rank feature matrices representing users and items. The method operates under the assumption that the feature matrices for users and items have a Gaussian distribution [Rendle, 2012].
2. PMF: The traditional probabilistic matrix factorization technique is employed to model the user-item rating matrix. This is achieved by decomposing the matrix into two lower-rank matrices, one representing user-specific information and the other representing item-specific information [Mnih and Salakhutdinov, 2007].
3. $SoMF$: This technique presents a general strategy for enhancing recommender systems by integrating data from social networks [Ma et al., 2011].
4. $HeteMF$: This approach uses similarities between various entities on the heterogeneous networks by utilizing the meta-path based on the related algorithms [Yu et al., 2013, Shi et al., 2015].
5. $SemRec$: A weighted heterogeneous information network (HIN), which may adaptably represent heterogeneous information for personalized recommendations, is the foundation of this approach [Shi et al., 2015].
6. DSR: It is a dual similarity regularization MF-based recommendation approach that simultaneously imposes constraints on users and items with high and low similarity [Zheng et al., 2017].
7. $HERec$: This approach uses the homogeneous network embedding technique Deep Walk and is based on heterogeneous information network embeddings [Shi et al., 2018a].
8. $HetNERec$: To discover the latent representations of users and items, this study suggests HetNERec, which builds a heterogeneous co-occurrence network. In order to provide recommendations, the multiple representations of each node are then combined into a single representation and integrated utilizing matrix factorization [Zhao et al., 2020].

In this section, to evaluate the performance of the proposed model, we separate each dataset into a training set and a test set. According to the sparsity of the dataset, we divided the Douban Movie and Douban Book datasets into $\{80\%, 60\%, 40\%, 20\%\}$ training ratios, and we used from $\{90\%, 80\%, 70\%, 60\%\}$ training ratios for the YELP dataset, in accordance with the references [Shi et al., 2018a, Zhao et al., 2020, He et al., 2021] and [Zheng et al., 2017]. It is noteworthy that, with any data set, it is clear that as the training rate increases, all methods improve at making recommendations. In this section, we present the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) of each method for various training ratios of the three datasets. Additionally, we display the improvement rate of other methods compared to Probabilistic Matrix Factorization (PMF). A low MAE and RMSE or a high improvement rate indicate exceptional performance. Tables 1, 2, and 3 depict the experimental data, and the following succinct summary of our findings follows:

Table 1: Comparison methods with together in Douban Movie dataset based on MAE and RMSE

Algorithm	Training	MAE	Improve	RMSE	Improve
PMF	20%	0.7247	-	0.9440	-
	40%	0.6078	-	0.8321	-
	60%	0.5867	-	0.7891	-
	80%	0.5741	-	0.7641	-
FM_{HIN}	20%	0.6080	+16.10%	0.7878	+16.55%
	40%	0.5871	+3.40%	0.7563	+9.10%
	60%	0.5769	+1.67%	0.7842	+0.62%
	80%	0.5696	+0.78%	0.7248	+5.55%
$SoMF$	20%	0.6979	+3.69%	0.9852	-4.36
	40%	0.6328	-4.11%	0.8479	-1.89%
	60%	0.5991	-2.11%	0.7950	-0.75%
	80%	0.5817	-1.32%	0.7680	-0.07%
$HeteMF$	20%	0.6896	+4.84%	0.9357	+0.88 %
	40%	0.6165	-1.43 %	0.8221	+1.20%
	60%	0.5894	-0.46%	0.7785	+1.34%
	80%	0.5750	-0.16%	0.7556	+1.53%
$SemRec$	20%	0.6392	+11.79%	0.8599	+8.91 %
	40%	0.5945	2.18%	0.7836	+5.82%
	60%	0.5738	+2.19%	0.7551	+4.30%
	80%	0.5695	+0.80%	0.7399	+3.58%
DSR	20%	0.6584	+9.14%	0.8345	+11.60%
	40%	0.6170	-1.51%	0.7850	+5.66%
	60%	0.5831	+0.61%	0.7408	+6.12%
	80%	0.5681	+1.04%	0.7225	+5.85%
$HERec$	20%	0.5900	+18.59%	0.7660	+18.86 %
	40%	0.5699	+6.23%	0.7315	+12.09%
	60%	0.5587	+4.77%	0.7148	+9.41%
	80%	0.5519	+3.86%	0.7053	+8.09%
$HetNERec$	20%	0.5623	-	0.7146	-
	40%	0.5555	-	0.7087	-
	60%	0.5464	-	0.6950	-
	80%	0.5412	-	0.6906	-
$HopRec$	20%	0.6212	+68.66%	0.7913	+65.15%
	40%	0.5876	+30.74%	0.7492	+33.78 %
	60%	0.5747	+12.22%	0.7311	+15.02 %
	80%	0.5677	+7.01 %	0.7242	+8.45 %
$RESCHet$	20%	**0.5856**	**+70.34** %	**0.7352**	**+67.25** %
	40%	**0.5681**	**+33.44** %	**0.7144**	**+35.82** %
	60%	**0.5437**	**+14.36** %	**0.6922**	**+18.70** %
	80%	**0.5388**	**+9.12** %	**0.6890**	**+10.45** %

Table 2: Comparison methods with together in Douban Book dataset based on MAE and RMSE

Algorithm	Training	MAE	Improve	RMSE	Improve
PMF	20%	1.0344	-	1.4414	-
	40%	0.6800	-	0.9203	-
	60%	0.6065	-	0.7908	-
	80%	0.5774	-	0.7414	-
FM_{HIN}	20%	0.6396	+38.17%	0.8188	+43.19%
	40%	0.6028	+11.35%	0.7617	+17.23%
	60%	0.5812	+4.17%	0.7319	+7.45%
	80%	0.5716	+1.00%	0.7199	+2.94%
$SoMF$	20%	0.6327	+38.83%	0.8236	+42.86%
	40%	0.6161	+9.40%	0.7936	+13.77%
	60%	0.5903	+2.67%	0.7518	+4.93%
	80%	0.5756	+0.31%	0.7302	+1.55%
$HeteMF$	20%	0.6311	+38.99%	0.8304	+42.39%
	40%	0.5982	+12.03%	0.7779	+15.47%
	60%	0.5823	+3.99%	0.7466	+5.59%
	80%	0.5740	+0.59%	0.7360	+0.77%
$SemRec$	20%	0.6481	+37.35%	0.8350	+42.07%
	40%	0.6025	+11.40%	0.7751	+15.78%
	60%	0.5833	+3.83%	0.7505	+5.10%
	80%	0.5675	+1.71%	0.7283	+1.81%
DSR	20%	0.6300	+39.10%	0.8200	+43.11%
	40%	0.6271	+7.78%	0.7730	+16.01%
	60%	0.6020	+0.74%	0.7552	+4.50%
	80%	0.5740	+0.59%	0.7206	+2.84%
$HERec$	20%	0.6450	+37.65%	0.8581	+40.47%
	40%	0.5774	+15.09%	0.7400	+19.59%
	60%	0.5600	+7.67%	0.7123	+9.93%
	80%	0.5502	+4.71%	0.6811	+8.17%
$HetNERec$	20%	0.5806	-	0.7341	-
	40%	0.5652	-	0.7102	-
	60%	0.5582	-	0.7029	-
	80%	0.5489	-	0.6942	-
$HopRec$	20%	0.5937	+61.80%	0.7612	+50.09%
	40%	0.5678	+22.25%	0.7231	+26.33%
	60%	0.5567	+10.34%	0.7090	+12.96%
	80%	0.5533	+4.77%	0.7004	+6.74%
$RESCHet$	20%	**0.5866**	**+63.18** %	**0.7288**	**+56.80** %
	40%	**0.5601**	**+24.38**%	**0.7483**	**+29.05**%
	60%	**0.5511**	**+13.02** %	**0.7423**	**+15.77** %
	80%	**0.5467**	**+8.30**%	**0.6801**	**+10.25**%

Table 3: Comparison methods with together in Yelp dataset based on MAE and RMSE

Algorithm	Training	MAE	Improve	RMSE	Improve
PMF	60%	1.1778	-	1.6167	-
	70%	1.1170	-	1.5387	-
	80%	1.0791	-	1.4816	-
	90%	1.0412	-	1.4268	-
FM_{HIN}	60%	0.9435	+19.89%	1.2039	+25.53%
	70%	0.9108	+18.46%	1.1651	+24.28%
	80%	0.9038	+16.25%	1.1497	+22.40%
	90%	0.9013	+13.44%	1.1417	+19.98%
$SoMF$	60%	1.1135	+5.46%	1.4748	+8.78%
	70%	1.0694	+4.26%	1.4201	+7.71%
	80%	1.0373	+3.87%	1.3782	+6.98%
	90%	1.0095	+3.04%	1.3392	+6.14%
$HeteMF$	60%	1.0368	+11.97%	1.3713	+15.18%
	70%	0.9975	+10.70%	1.3229	+14.02%
	80%	0.9654	+10.54%	1.2799	+13.61%
	90%	0.9487	+8.88%	1.2549	+12.05%
$SemRec$	60%	0.9637	+18.18%	1.2380	+23.42%
	70%	0.9407	+15.78%	1.2108	+21.31%
	80%	0.9176	+14.97%	1.1771	+20.55%
	90%	0.9043	+13.15%	1.1637	+18.44%
DSR	60%	1.0043	+14.73%	1.2257	+24.19%
	70%	0.9429	+15.59%	1.1582	+24.73%
	80%	0.9098	+15.69%	1.1208	+24.35%
	90%	0.9054	+13.04%	1.1186	+21.60%
$HERec$	60%	0.8759	+25.63%	1.1488	+28.94%
	70%	0.8580	+23.19%	1.1256	+26.85%
	80%	0.8475	+21.46%	1.1117	+24.97%
	90%	0.8395	+19.37%	1.0907	+23.56%
$HetNERec$	60%	0.8072	-	1.0319	-
	70%	0.8002	-	1.0320	-
	80%	0.8075	-	1.0321	-
	90%	0.7838	-	1.0204	-
$HopRec$	60%	0.8133	+32.66%	1.0478	+33.15%
	70%	0.8007	+29.68%	1.0474	+29.76%
	80%	0.7934	+26.24%	1.0388	+26.74%
	90%	0.7926	+24.15%	1.0346	+25.02%
$RESCHet$	60%	**0.8177**	**+29.05%**	**1.0510**	**+31.55%**
	70%	**0.8005**	**+29.77%**	**1.0463**	**+30.55%**
	80%	**0.7968**	**+25.97%**	**1.0315**	**+28.50%**
	90%	**0.7810**	**+26.08%**	**1.0192**	**+28.41%**

5.4 RESCHet Analysis and Experiment Results

- The RESCHet method has been demonstrated to have superior performance compared to other methods in all three datasets. Specifically, in the Douban Movie dataset, RESCHet achieved a Mean Absolute Error (MAE) of 0.5388 and a Root Mean Squared Error (RMSE) of 0.6890. Similarly, in the Douban Book dataset, RESCHet obtained an MAE of 0.5467 and an RMSE of 0.6801. In the Yelp dataset, RESCHet demonstrated the best performance with an MAE of 0.7810 and an RMSE of 1.0192. These results highlight the accuracy and effectiveness of RESCHet in the recommendation task.

- The results of our experiments show that the recommendation performance of all methods improves as the training rate increases across all datasets. The rate of improvement compared to PMF is found to be positively correlated with the density of the data set. Furthermore, methods based on Heterogeneous Information Networks (HIN) demonstrate superior recommendation performance compared to traditional matrix factorization models.

- The proposed method, RESCHet, demonstrates consistent superiority over baseline methods, including PMF and DSR. RESCHet adopts a more systematic approach for leveraging heterogeneous information networks (HINs) to enhance recommender systems, leading to improved information extraction and utilization when compared to other HIN-based methods. The method proposed in this study involves incorporating spectral clustering and limiting meta-paths based on the membership of the clusters. The atomic meta-paths are defined such that users in each cluster are highly similar in terms of interests. As a result of this approach, the recommendations generated by the proposed method are shown to be more accurate when compared to other methods.

- In comparison to other methods, the proposed RESCHet method demonstrates the highest overall performance. RESCHet outperforms the HERec and HopRec models (HIN embedding based on the matrix factorization method) by taking into account not only the embedding features of the user and item but also the interaction information between these features. This is achieved through the use of special sub-meta-paths and the Hadamard product for embedding vectors of users and items within each cluster, leading to a more accurate determination of items that align with the interests of the users.

6 Conclusion

In this paper, we introduce RESCHet, a novel recommendation method based on heterogeneous network embedding. Our approach utilizes spectral clustering within the framework of a Heterogeneous Information Network (HIN) to generate accurate and diverse recommendations. To the best of our knowledge, this is the first investigation of incorporating embedding spectral clustering in the context of HIN for recommender systems. Our approach begins with utilizing meta-paths to learn the embedding representation of users. Then, user clustering is performed through embedding spectral clustering on the heterogeneous network users. The Metapath2vec method is employed for node embedding, with a focus on user nodes to ensure efficient representations as the clustering process is based on these representations. In the subsequent stage, recommendations are generated for each cluster of users by defining two types of restricted meta-paths: sub-meta-paths and atomic meta-paths. These meta-paths are designed to limit their scope based on the cluster memberships of users. Submetapaths are then created for each cluster to uncover hidden dependencies and relationships between users and items. Finally, the results from each sub-metapath are transformed into vectors for embedding, and the relationships between pairs of sub-meta-paths are exploited through the application of a Hadamard product between the embedded vectors. RESCHet aims to effectively discover and leverage relationships between users and items in heterogeneous networks, resulting in improved recommendation performance. The experimental results demonstrate that the proposed method in this research outperforms other comparative methods in terms of the accuracy of recommendations for users, as measured by the established evaluation criteria of MAE and RMSE. In this paper, user-based clusters were generated. As a next step, future research aims to establish item-based clusters and generate meta-paths based on item similarity. This will result in multiple clusters of items with high similarities, enabling recommendations of similar items to users based on their preferred items.

References

[Ai et al., 2018] Ai, Q., Azizi, V., Chen, X., and Zhang, Y. (2018). Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9):137.

[Bobadilla et al., 2011] Bobadilla, J., Ortega, F., Hernando, A., and Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-based systems*, 24(8):1310–1316.

[Cai et al., 2018] Cai, H., Zheng, V. W., and Chang, K. C.-C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.

[Chen et al., 2018] Chen, J., Wang, H., Yan, Z., et al. (2018). Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering. *Swarm and Evolutionary Computation*, 38:35–41.

[Da'u et al., 2021] Da'u, A., Salim, N., and Idris, R. (2021). An adaptive deep learning method for item recommendation system. *Knowledge-Based Systems*, 213:106681.

[Duan et al., 2022] Duan, R., Jiang, C., and Jain, H. K. (2022). Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem. *Decision Support Systems*, 156:113748.

[Fang et al., 2020] Fang, Y., Yang, Y., Zhang, W., Lin, X., and Cao, X. (2020). Effective and efficient community search over large heterogeneous information networks. *Proceedings of the VLDB Endowment*, 13(6):854–867.

[Forouzandeh et al., 2021] Forouzandeh, S., Rostami, M., and Berahmand, K. (2021). Presentation a trust walker for rating prediction in recommender system with biased random walk: Effects of h-index centrality, similarity in items and friends. *Engineering Applications of Artificial Intelligence*, 104:104325.

[Forouzandeh et al., 2015] Forouzandeh, S., Soltanpanah, H., and Sheikhahmadi, A. (2015). Application of data mining in designing a recommender system on social networks. *International Journal of Computer Applications*, 124(1).

[Gao et al., 2022a] Gao, C., Wang, X., He, X., and Li, Y. (2022a). Graph neural networks for recommender system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1623–1625.

[Gao et al., 2022b] Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., Quan, Y., Chang, J., Jin, D., He, X., et al. (2022b). A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems (TORS)*.

[Gao et al., 2018] Gao, L., Yang, H., Zhou, C., Wu, J., Pan, S., and Hu, Y. (2018). Active discriminative network representation learning. In *IJCAI International Joint Conference on Artificial Intelligence*.

[Guo and Dai, 2022] Guo, L. and Dai, Q. (2022). Graph clustering via variational graph embedding. *Pattern Recognition*, 122:108334.

[Guo et al., 2018] Guo, L., Wen, Y.-F., and Wang, X.-H. (2018). Exploiting pre-trained network embeddings for recommendations in social networks. *Journal of Computer Science and Technology*, 33:682–696.

[He et al., 2021] He, Y., Zhang, Y., Qi, L., Yan, D., and He, Q. (2021). Outer product enhanced heterogeneous information network embedding for recommendation. *Expert Systems with Applications*, 169:114359.

[Hou et al., 2019] Hou, Y., Yang, N., Wu, Y., and Yu, P. S. (2019). Explainable recommendation with fusion of aspect information. *World Wide Web*, 22:221–240.

[Huang et al., 2022] Huang, C., Fang, Y., Lin, X., Cao, X., and Zhang, W. (2022). Able: Meta-path prediction in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(4):1–21.

[Jiang et al., 2018] Jiang, Z., Yin, Y., Gao, L., Lu, Y., and Liu, X. (2018). Cross-language citation recommendation via hierarchical representation learning on heterogeneous graph. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 635–644.

[Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

[Koren et al., 2022] Koren, Y., Rendle, S., and Bell, R. (2022). Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142.

[Li et al., 2020a] Li, H., Wang, Y., Lyu, Z., and Shi, J. (2020a). Multi-task learning for recommendation over heterogeneous information network. *IEEE Transactions on Knowledge and Data Engineering*.

[Li et al., 2020b] Li, X., Wu, Y., Ester, M., Kao, B., Wang, X., and Zheng, Y. (2020b). Schain-iram: An efficient and effective semi-supervised clustering algorithm for attributed heterogeneous information networks. *IEEE Transactions on knowledge and data engineering*.

[Lin et al., 2022] Lin, Z., Tian, C., Hou, Y., and Zhao, W. X. (2022). Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022*, pages 2320–2329.

[Liu et al., 2022a] Liu, J., Shi, C., Yang, C., Lu, Z., and Philip, S. Y. (2022a). A survey on heterogeneous information network based recommender systems: Concepts, methods, applications and resources. *AI Open*.

[Liu et al., 2023] Liu, X., Wu, K., Liu, B., and Qian, R. (2023). Hnerec: Scientific collaborator recommendation model based on heterogeneous network embedding. *Information Processing & Management*, 60(2):103253.

[Liu et al., 2022b] Liu, Z., Yang, L., Fan, Z., Peng, H., and Yu, P. S. (2022b). Federated social recommendation with graph neural network. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–24.

[Lu et al., 2020] Lu, Y., Fang, Y., and Shi, C. (2020). Meta-learning on heterogeneous information networks for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1563–1573.

[Ma et al., 2011] Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. (2011). Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296.

[Mnih and Salakhutdinov, 2007] Mnih, A. and Salakhutdinov, R. R. (2007). Probabilistic matrix factorization. *Advances in neural information processing systems*, 20.

[Natarajan et al., 2020] Natarajan, S., Vairavasundaram, S., Natarajan, S., and Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Systems with Applications*, 149:113248.

[Palomares et al., 2021] Palomares, I., Porcel, C., Pizzato, L., Guy, I., and Herrera-Viedma, E. (2021). Reciprocal recommender systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation. *Information Fusion*, 69:103–127.

[Pham et al., 2023] Pham, P., Nguyen, L. T., Nguyen, N.-T., Pedrycz, W., Yun, U., Lin, J. C.-W., and Vo, B. (2023). An approach to semantic-aware heterogeneous network embedding for recommender systems. *IEEE Transactions on Cybernetics*.

[Polatidis and Georgiadis, 2016] Polatidis, N. and Georgiadis, C. K. (2016). A multi-level collaborative filtering method that improves recommendations. *Expert Systems with Applications*, 48:100–110.

[Rendle, 2012] Rendle, S. (2012). Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–22.

[Satuluri et al., 2020] Satuluri, V., Wu, Y., Zheng, X., Qian, Y., Wichers, B., Dai, Q., Tang, G. M., Jiang, J., and Lin, J. (2020). Simclusters: Community-based representations for heterogeneous recommendations at twitter. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3183–3193.

[Shen et al., 2018] Shen, X., Pan, S., Liu, W., Ong, Y.-S., and Sun, Q.-S. (2018). Discrete network embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3549–3555.

[Shi et al., 2018a] Shi, C., Hu, B., Zhao, W. X., and Philip, S. Y. (2018a). Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370.

[Shi et al., 2016] Shi, C., Li, Y., Zhang, J., Sun, Y., and Philip, S. Y. (2016). A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37.

[Shi et al., 2019] Shi, C., Zhang, Z., Ji, Y., Wang, W., Yu, P. S., and Shi, Z. (2019). Semrec: a personalized semantic recommendation method based on weighted heterogeneous information networks. *World Wide Web*, 22:153–184.

[Shi et al., 2015] Shi, C., Zhang, Z., Luo, P., Yu, P. S., Yue, Y., and Wu, B. (2015). Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 453–462.

[Shi et al., 2018b] Shi, Y., Zhu, Q., Guo, F., Zhang, C., and Han, J. (2018b). Easing embedding learning by comprehensive transcription of heterogeneous information networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2190–2199.

[Tian et al., 2014] Tian, F., Gao, B., Cui, Q., Chen, E., and Liu, T.-Y. (2014). Learning deep representations for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

[Walker et al., 2022] Walker, J., Zhang, F., Zhou, F., and Zhong, T. (2022). Social-trust-aware variational recommendation. *International Journal of Intelligent Systems*, 37(4):2774–2802.

[Wang et al., 2022a] Wang, C., Pan, S., Celina, P. Y., Hu, R., Long, G., and Zhang, C. (2022a). Deep neighbor-aware embedding for node clustering in attributed graphs. *Pattern Recognition*, 122:108230.

[Wang et al., 2020] Wang, W., Zhang, W., Rao, J., Qiu, Z., Zhang, B., Lin, L., and Zha, H. (2020). Group-aware long-and short-term graph representation learning for sequential group recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1449–1458.

[Wang et al., 2022b] Wang, X., Bo, D., Shi, C., Fan, S., Ye, Y., and Philip, S. Y. (2022b). A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Transactions on Big Data*.

[Wang et al., 2019] Wang, Z., Liu, H., Du, Y., Wu, Z., and Zhang, X. (2019). Unified embedding model over heterogeneous information network for personalized recommendation. In *Proceedings of the 28th international joint conference on artificial intelligence*, pages 3813–3819.

[Wu et al., 2022] Wu, S., Sun, F., Zhang, W., Xie, X., and Cui, B. (2022). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37.

[Xu et al., 2019] Xu, F., Lian, J., Han, Z., Li, Y., Xu, Y., and Xie, X. (2019). Relation-aware graph convolutional networks for agent-initiated social e-commerce recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 529–538.

[Yang et al., 2014] Yang, X., Guo, Y., Liu, Y., and Steck, H. (2014). A survey of collaborative filtering based social recommender systems. *Computer communications*, 41:1–10.

[Yu et al., 2013] Yu, X., Ren, X., Gu, Q., Sun, Y., and Han, J. (2013). Collaborative filtering with entity similarity regularization in heterogeneous information networks. *IJCAI HINA*, 27.

[Zhao et al., 2020] Zhao, Z., Zhang, X., Zhou, H., Li, C., Gong, M., and Wang, Y. (2020). Hetnerec: Heterogeneous network embedding based recommendation. *Knowledge-based systems*, 204:106218.

[Zheng et al., 2017] Zheng, J., Liu, J., Shi, C., Zhuang, F., Li, J., and Wu, B. (2017). Recommendation in heterogeneous information network via dual similarity regularization. *International Journal of Data Science and Analytics*, 3(1):35–48.

[Zhou et al., 2020] Zhou, S., Bu, J., Zhang, Z., Wang, C., Ma, L., and Zhang, J. (2020). Cross multi-type objects clustering in attributed heterogeneous information network. *Knowledge-Based Systems*, 194:105458.