



Article

Multi-Layer Feature Fusion-Based Community Evolution Prediction

Zhao Wang [†], Qingguo Xu [†] and Weimin Li ^{*}

School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; a1914411803@163.com (Z.W.); qgxu@t.shu.edu.cn (Q.X.)

^{*} Correspondence: wml@i.shu.edu.cn

[†] These authors contributed equally to this work.

Abstract: Analyzing and predicting community evolution has many important applications in criminology, sociology, and other fields. In community evolution prediction, most of the existing research is simply calculating the features of the community, and then predicting the evolution event through the classifier. However, these methods do not consider the complex characteristics of community evolution, and only predict the community's evolution from a single level. To solve these problems, this paper proposes an algorithm called multi-layer feature fusion-based community evolution prediction, which obtains features from the community layer and node layer. The final community feature is the fusion of the two layer features. At the node layer, this paper proposes a global and local-based role-extraction algorithm. This algorithm can effectively discover different roles in the community. In this way, we can distinguish the influence of nodes with different characteristics on the community evolution. At the community layer, this paper proposes to use the community hypergraph to obtain the inter-community interaction relationship. After all the features are obtained, this paper trains a classifier through these features and uses them in community evolution prediction. The experimental results show that the algorithm proposed in this paper is better than other algorithms in terms of prediction effect.

Keywords: community evolution prediction; non-negative matrix factorization; multi-layer features; role extraction



Citation: Wang, Z.; Xu, Q.; Li, W. Multi-Layer Feature Fusion-Based Community Evolution Prediction. *Future Internet* **2022**, *14*, 113. <https://doi.org/10.3390/fi14040113>

Academic Editor: Vijayakumar Varadarajan

Received: 13 March 2022

Accepted: 1 April 2022

Published: 6 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many complex networks show the structure of communities. A community is a set of nodes, where connections within the community are relatively dense, and connections between communities are relatively sparse [1]. The community is the basic structure of the network, and has many important applications, such as influence maximization [2,3] and rumor tracing. Due to the wide application of community, ever since the concept of community has been proposed, many scholars have devoted themselves to related research on community. These studies include community discovery [4,5], community evolution analysis [6], and community evolution prediction.

Among many research directions related to communities, community evolution prediction is an important direction. In dynamic networks, communities change over time, which we have called community evolution. Community evolution prediction can be defined as predicting the future evolution of a community when its historical evolution is known.

Community evolution prediction has many important applications in many fields. For example, in criminology [7], a criminal group can be regarded as a community, and tracking and predicting its changes can help to control them over time. In an infectious disease transmission network, by tracking the changes in infected communities, the structural characteristics of highly infectious communities can be found to predict the future spread of infectious diseases.

Although community evolution prediction has important applications in many fields, there are few related studies. Most of the existing research simply calculates community features based on manually selected features and then uses classifiers to make predictions. Most of the differences among these algorithms lie in the way they calculate features [8–15]. For example, in references [8,11], the authors simply calculate features based on manual selection. In references [9,10], the authors take into account the influence of the leader node on the evolution of the community. In the reference [15], the author automatically selects relevant features through pre-training classifiers.

Although the existing algorithms can achieve good effects, they still have some shortcomings. First, the existing research does not consider the impact of the nodes with different features on the evolution of the community. To solve this problem, in references [9,10], the authors introduced leader nodes when calculating community features. However, in reality, there are not only leader nodes in the community, but also other types of nodes. These nodes also affect the community evolution. For example, in a community, there may be nodes that want to separate. If there are many such nodes, the community is likely to split in the future. Therefore, when predicting the evolution of a community, it is not enough to only consider the features of influence nodes. Second, the existing algorithms do not consider mutual influence between communities when calculating community features. Obviously, the relationship between communities will affect the evolution of the community. For example, if there are more edges between two communities, these two communities are more likely to merge in the future.

To solve the above problems, this paper proposes multi-layer feature fusion-based community evolution prediction (MCEP). MCEP obtains features from the node layer and the community layer. Then MCEP merges the obtained features and uses the merged features to predict the evolution of the community. At the node layer, we give community roles to different types of nodes in the community. In contrast to the traditional definition of the role, community roles need to consider both the global scope and local scope features of nodes. Therefore, the traditional role-discovery algorithm [16] cannot be used to discover community roles. To accurately discover community roles, we propose the global and local-based community role-extraction algorithm. The algorithm first constructs the global and local feature matrices of nodes in the same community. Then MCEP performs non-negative matrix factorization (NMF) on these two matrices to obtain the role of the node. At the community layer, because the community is a mesoscopic structure, we call the features of the mutual influence between communities the community mesoscopic features. To obtain the community mesoscopic features, we regard the community as a hypernode and use the hypernode to construct the hypergraph. Then, we perform NMF on the adjacency matrix of the hypergraph to obtain the community mesoscopic features. After all the features are obtained, we merge the obtained features. The merged feature is the final feature representation of the community. In addition, to incorporate time information into the training set, we integrate the features of the community into multiple snapshots. Finally, MCEP constructs a training set according to the obtained features and then trains a classifier as the final community evolution prediction model. Experimental results show that the algorithm proposed in this paper is better than other algorithms.

The main contributions of this paper can be summarized as follows:

- We propose to obtain features from the node layer and the community layer, respectively. In this way, the features of nodes and the influence of inter-community relationships on community evolution can be better captured.
- We propose the global and local-based community role-extraction algorithm. The algorithm first obtains the features of the node from the global scope and the local scope. Then, global and local feature matrices are constructed based on the obtained node features. Finally, we obtain the role attribution matrix of the node through NMF. Through the role attribution matrix, we can assign community roles to each node.
- We propose the obtaining of the community mesoscopic features through a community hypergraph. To obtain the community mesoscopic features, we first construct the

hypergraph according to the similarity between the communities. Then we perform NMF on the adjacency matrix corresponding to the hypergraph to obtain the mutual influence between the communities.

The rest of the paper is organized as follows: Section 2 introduces the overall process of community evolution prediction and related work. Section 3 introduces the MCEP algorithm proposed in this paper in detail. Section 4 shows the results of the comparative experiment and other related experiments. Section 5 summarizes the full paper.

2. Related Work

2.1. Community Evolution

In a dynamic network, a community will evolve with a change in the network. To study the changes in a community in a relevant snapshot, scholars have proposed relevant community evolution models. These models define community evolution events. For the specific definition of community evolution events, different models will be different, and the most commonly used model is group evolution discovery (GED). Since the algorithm proposed in this paper analyzes community evolution through GED, this section will briefly introduce community evolution events in GED.

In GED, community evolution events are divided into seven categories:

- Birth: a community exists in the current snapshot, but in the previous snapshot, no community could match that community.
- Survive: two communities in the adjacent snapshot are identical or only a small number of nodes are different.
- Dissolve: a community existed in the previous snapshot, and no community can match it in the current snapshot.
- Grow: in the current snapshot, some new nodes join the community, and make the size of community bigger than the previous snapshot.
- Shrink: shrinking is the opposite process growing; a community is reduced in size compared to the previous snapshot.
- Merge: a community in the current snapshot is formed by merging several communities in the previous snapshot.
- Split: a community divides into several small communities in the current snapshot.

2.2. Community Evolution Prediction

Most of the existing research on community evolution prediction uses four steps to make predictions. The four steps are network data split, community discovery, evolution analysis, obtaining relevant features, and making predictions. In these four steps, most of the papers focus on the last step, which is the obtaining of features.

The existing algorithms can be divided into two types: (a) methods that do not contain historical information; and (b) methods that contain historical information. The difference between the two types of method is whether to use historical information.

The method that does not contain historical information only uses the community of the current snapshot for prediction. The advantage of these methods is that it is relatively simple, and at the same time, the obtained feature dimension is lower, so the prediction can be made more quickly. Takaffoli et al. [12] proposed the concept of the meta-community. Through the meta-community, the evolutionary relationship of the non-adjacent snapshot can be tracked. In addition, they used properties of influential nodes, properties of the community, and temporal changes in these properties predict community evolution. Ilhan et al. [15] pre-trained a multi-classifier through many synthetic networks. The multi-classifier can automatically select features useful for prediction from a feature set before prediction.

Community evolution is a sequential process. The historical evolution of the community will affect the current evolution. The method that contains historical information considers the historical evolution, so the effect is better than the method that does not contain historical information. Diakidis et al. [13] hypothesized that communities in social

networks are the result of user interaction. Therefore, they selected structure, content, contextual features, and the previous state of community as community features. However, their method does not consider the split and merge events of the community. İlhan et al. [14] used ARIMA (autoregressive integrated moving average) to predict the future changes in community features. Then, they spliced the predicted features with the community features of the historical snapshot to obtain the final feature representation. Dakiche et al. [10] considered historical community features and the influence of leader nodes on community evolution prediction, and used the change rate of community features to predict community evolution. The above methods are mainly based on GED (group evolution discovery) [17] for community evolution analysis. Gliwa et al. [8] proposed a new community evolution model called stable group change identification (SGCI) and used it to predict community evolution. SGCI used leadership, community density, cohesion, and group size to predict the future evolution of the community.

Although the above methods have been able to achieve good results, these algorithms do not consider the complex characteristics of the community evolution. Specifically, the existing algorithms have the following drawbacks: first, although existing algorithms consider the influence of leader nodes on the prediction results, it is obvious that there are other types of nodes in the community. The influence of these nodes on the evolution of the community has not been considered. Second, the existing algorithms do not consider the influence between communities. The relationship between communities is an important factor affecting the evolution of the community.

To solve the above problems, this paper proposes multi-layer feature fusion-based community evolution prediction (MCEP).

3. Multi-Layer Feature Fusion-Based Community Evolution Prediction

This section will introduce the MCEP algorithm in detail. As mentioned in the related work section, community evolution prediction is mainly divided into four steps. The main work of this paper is the last step, which obtains the corresponding features.

The structure of this section is as follows. Section 3.1 will briefly introduce the features used in the MCEP. Section 3.2 will introduce the global and local-based community role-extraction algorithm proposed in this paper. Section 3.3 will introduce the algorithm for discovering the community mesoscopic features. Section 3.4 briefly introduces the prediction process of the MCEP algorithm.

Before introducing MCEP, we first briefly introduce the related symbols used in this paper.

In this paper, the dynamic network is defined as $G = (V_t, E_t)$, where V_t is the node in the network at time t , E_t is the edge in the network at time t , $t = 1, 2, \dots, T$, and T is the number of the snapshot in the network. $N_t = |V_t|$ represents the number of nodes in the network at time t . C_t represents the community structure at time t , $C_{(t,i)}$ represents the i -th community at time t , and $Role_C$ represents the division of roles in community C . The main symbols used in this article are given in Table 1.

Table 1. Notations.

Symbol	Meaning
G_t	The network of snapshot t
C_t	Community set at snapshot t
$C_{t,i}$	Community j at snapshot t
$\xi_C \in R^{ C_i \times M}$	The global feature matrix of community C
$\ell_C \in R^{ C_i \times M}$	The local feature matrix of community C
$F_C \in R^{ C \times L}$	The mesoscopic feature of community C
$W_C \in R^{K \times M}$	The global feature contribution matrix
$H_C \in R^{K \times M}$	The local feature contribution matrix
$\Re_C \in R^{ C_i \times K}$	The role attribution matrix

In the above table, $|C|$ is the number of communities in the network, $|C_i|$ is the number of nodes in the community C , K is the number of roles, and L is the number of community mesoscopic features.

3.1. Community Features

Similar to other community evolution prediction algorithms, MCEP also needs to construct a training set by calculating related features. This section will introduce the features used in MCEP.

The features used in MCEP can be divided into three types: overall community features, node features, and community mesoscopic features. Compared with the first two features, the calculation of community mesoscopic features is more complicated. We will introduce it in detail in Section 3.3. This section mainly introduces the overall community features and node features used in MCEP. It is worth pointing out that because different networks have different properties, finding a suitable feature set to make it effective for all networks is difficult. The experimental part will also illustrate this point. The features listed below are just some of the general features. In practical applications, the user should make a choice based on specific conditions.

The overall community features measure the community from an overall perspective. This measurement mainly reflects the features of the community at the macro level. The overall community features used by MCEP are community size, density, and cohesion. The community size measures the number of nodes in the community. Both the density and the cohesion are measures of the closeness of the connections within the community. The density of the community pays more attention to the connections within the community. The cohesion considers the number of internal connections while also considering the connections outside the community.

The overall community features are macroscopic reflections of the community features. In addition to the macro features, the community also has its micro features. Micro features refer to the features of nodes in the community. To obtain the micro features, MCEP calculates the features of each node in the community and then calculates its average value. The average value is the micro features of the community. The node features used in MCEP are degree, clustering coefficient [18], betweenness centrality [19], closeness centrality, and eigenvector centrality [20]. Among them, clustering coefficient and closeness centrality mainly measure the tightness of connections around nodes. Centrality and eigenvector centrality mainly measure the importance of nodes in the network.

The relevant calculation formulas and symbol meanings of the above features are given in Table 2.

Table 2. Community features.

Feature	Calculation Formula
degree	$D_v = Nei_v $
Size	$S = V $
Density	$den(C) = \frac{\sum_v D_v}{ C \times (C - 1)}$
Cohesion	$\zeta_c = \frac{den(C)}{O_c}$
Clustering Coefficient	$cluser(v) = \frac{PL_v}{L_v}$
Betweenness Centrality	$BC_v = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$
Closeness centrality	$CC_v = \frac{1}{\sum_{u \in G} Dis(v, u)}$
Eigenvector Centrality	$EC_v = x_v$

In the above table, Nei_v is the set of neighbor nodes of node v of node v , O_c is the number of edges between nodes in community C and external nodes, and PL_v represents the maximum number of possible connections of node v 's neighbors, L_v represents the actual number of connections of the neighbors of node v , σ_{st} represents the number of shortest paths from node s to node t , $\sigma_{st}(v)$ represents the number of paths that the shortest path number passes through v , $Dis(v, u)$ represents the distance from node v to node u , and x_v represents the feature vector of node v .

3.2. Global- and Local-Based Community Role Extraction

This section introduces the global and local-based community role-extraction algorithm in detail.

As mentioned above, in addition to leader nodes, there are many other types of nodes in the community, which will also affect the evolution of the community. We refer to different types of nodes in the community as community roles, which are specifically defined as follows:

Definition 1 (Community role). *The community role classifies nodes according to the structural features of nodes. Community role describes the different features of nodes in the community. Nodes belonging to different roles will have different effects on the evolution of the community.*

In contrast to traditional role extraction, the discovery of community role not only needs to consider the network structure features of the node, but also needs to consider the community structure features of the node. Therefore, it is difficult to accurately discover community roles using traditional role-extraction algorithms. To accurately discover the community role, we propose the global- and local-based community role-extraction algorithm (GLRE). In this algorithm, global refers to the whole network, and local refers to the community where the node is located.

GLRE obtains the community role mainly by obtaining the community role attribution matrix. The definition of the community role attribution matrix is as follows:

Definition 2 (Community role attribution matrix). *The community role attribution matrix R reflects the probability that a node belongs to each community role. The role can be assigned to nodes in the community through this matrix.*

To obtain the community role attribution matrix, GLRE first calculates the structural features of the node from the global scope and the local scope. Then, GLRE constructs the global and local feature matrix. The features used in GLRE are the node features mentioned in Section 3.1. After obtaining the global and local feature matrices, GLRE decomposes the two matrices simultaneously through NMF to obtain the community role attribution matrix. The rest of this section will introduce the process of the GLRE algorithm in detail.

First, GLRE calculates the features of nodes in the global and local scopes and constructs the global and local feature matrix ξ_C and ℓ_C .

It is worth pointing out that when calculating the features of a node from a local scope, GLRE will ignore nodes that do not belong to the community. Figure 1 is an example. Among them, nodes of the same color belong to the same community. For node 1 in the graph, its degree is 4. Among the neighbors of node 1, two nodes belong to the same community as node 1, and two nodes belong to different communities from node 1. When computing node features from the global scope, the degree of node 1 is 4, as shown on the left. When computing node features from a local scope, the degree of node 1 is 2, as shown on the right.

After obtaining the global and local feature matrices of the community, GLRE uses NMF to decompose matrix ξ_C and ℓ_C to obtain the community role attribution matrix.

To consider both the global and local features of the node in the NMF, we design an effective loss function so that GLRE can decompose matrix ξ_C and ℓ_C at the same time in the NMF. The loss function is as follows:

$$Cost = ||\ell_C - \Re_C H_C||_F^2 + ||\xi_C - \Re_C W_C||_F^2 \quad (1)$$

where ξ_C is the global feature matrix of community C , ℓ_C is the local feature matrix of community C , $W_C \in R^{M \times K}$ is the global feature contribution matrix of nodes in C , and $H_C \in R^{M \times K}$ is the local feature contribution matrix of nodes in C , and $\Re_C \in R^{M \times K}$ is the role attribution matrix of nodes in community C .

After obtaining the loss function, the GLRE algorithm uses the multiplicative update rule [21] to minimize the loss function to obtain W_C , H_C and $\Re_C$. The final update formula is as follows:

$$H_C = H_C \frac{\mathfrak{R}_C^T \ell_C}{\mathfrak{R}_C^T \mathfrak{R}_C H_C} \quad (2)$$

$$\mathfrak{R}_C = \mathfrak{R}_C \frac{\ell_C H_C^T + \xi_C W_C^T}{\mathfrak{R}_C H_C H_C^T + \mathfrak{R}_C W_C W_C^T} \quad (3)$$

$$W_C = W_C \frac{\mathfrak{R}_C^T \xi_C}{\mathfrak{R}_C^T \mathfrak{R}_C W_C} \quad (4)$$

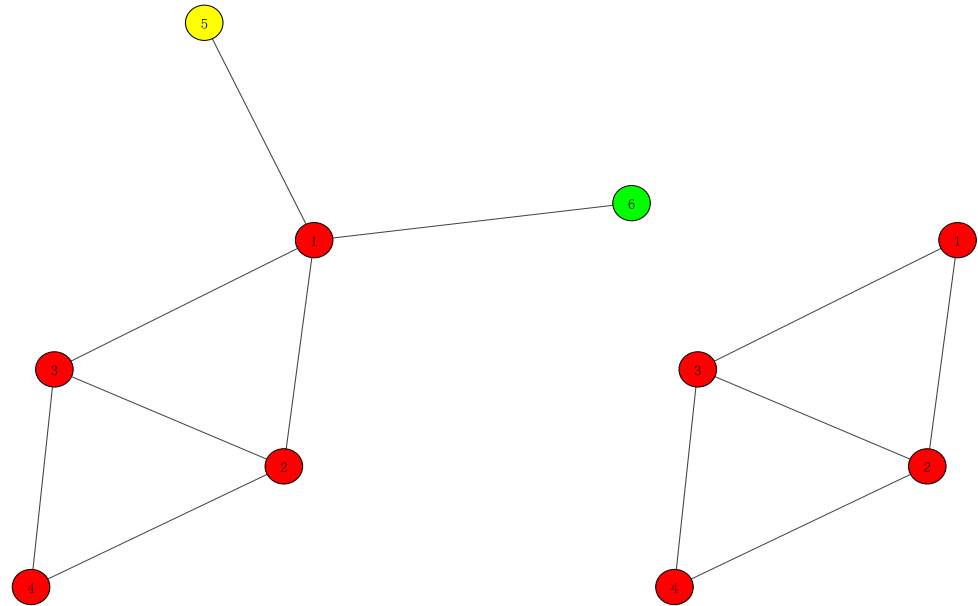


Figure 1. Calculate degree from global scope and local scope.

After obtaining the community role attribution matrix $\mathfrak{R}_C$, we can obtain the community role of each node. Each row in $\mathfrak{R}_C$ represents the probability that the node belongs to each community role. The nodes in the community can be assigned roles through the elements in $\mathfrak{R}_C$:

$$\mathfrak{R}_{C,i} = \underset{k}{\operatorname{argmax}} \mathfrak{R}_{C,ik} \quad (5)$$

The process of the GLRE algorithm is shown in Algorithm 1.

Algorithm 1 GLRE

Require: network structure G , community C

- 1: Calculate the node features of each node from the global and local, respectively
 - 2: Construct matrices ξ_C and ℓ_C according to the obtained node features
 - 3: Get H_C , W_C , $\mathfrak{R}_C$, according to Formulas (2)–(4)
 - 4: Obtain the role distribution $\mathfrak{R}_{C,i}$ of each node according to Formula (5)
 - 5: Output division result $\mathfrak{R}_C$
-

After obtaining the roles of nodes in the community through GLRE, for nodes with the same role, MCEP calculates the node features and uses the average value as the features of this type of role.

3.3. Community Mesoscopic Feature Mining

This section will first introduce the definition of community mesoscopic features proposed in this paper. Then, this section will propose the corresponding mesoscopic feature-mining algorithm based on the community hypergraph.

Community is a mesoscopic structure in the network. In the real world, there is much interaction between communities. These interactions will also have an impact on the future evolution of the community. To consider the influence relationship between communities, we define the community mesoscopic feature, which is defined as follows:

Definition 3 (Community mesoscopic feature). *Community mesoscopic feature describes the mutual influence relationship between communities. It is a concrete manifestation of the influence of other communities on this community. In this paper, the community mesoscopic feature of community C is represented by F_C . The mesoscopic features of all communities are represented by matrix F .*

Since the interaction between communities is a complex and abstract effect, the mesoscopic features of the community cannot be obtained through simple calculation. To accurately discover the community mesoscopic features, we propose an algorithm for mining mesoscopic features. This algorithm first treats the community as a hypernode in the network, then constructs a community hypergraph based on the similarity between hypernodes. Finally, the algorithm obtains the community mesoscopic features through NMF.

We first give the definition of the community hypergraph, as follows:

Definition 4 (Community hypergraph). *Hypergraph is a concept opposite to the simple graph. A node in a hypergraph is a set of nodes, and the edges in a hypergraph describe the relationship between the set of nodes. A community hypergraph is a hypergraph constructed by treating a community as a node. We refer to the community in the hypergraph as community hyper-nodes.*

Figure 2 shows an example of a community hypergraph. In Figure 2, the network on the left is the original network, in which the nodes of the same color belong to the same community. The network on the right is the community hypergraph constructed from the original network, and the relationship between the hypernode and the community is given by the solid arrow. As can be seen from Figure 2, by constructing the community hypergraph, the algorithm can ignore the complexity of the internal connections of the community, and it can focus on mining the influence relationship between the communities. In addition, in the original network, there is no direct connection between the blue community and the green community. But in the hypergraph, the blue node and the green node are connected, because in the real network, the influence between the community can be generated not only through the interconnection of nodes between communities, but also through other means, such as mutual attention between communities and so on.

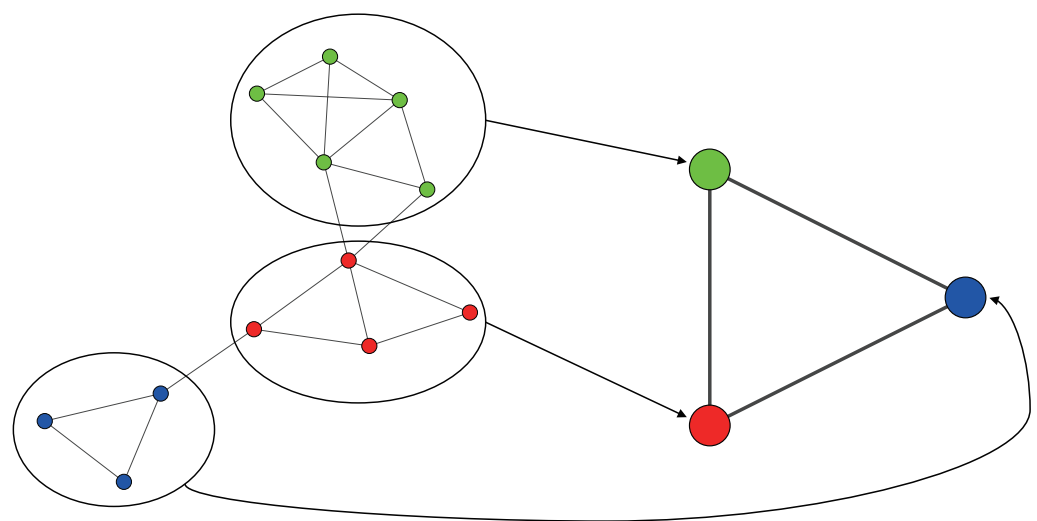


Figure 2. The example of hypergraph.

In this paper, the community hypergraph is constructed by calculating the similarity of community hyper-nodes. The weight of the edges in the hypergraph is the corresponding hypernode similarity. The definition of hypernode similarity is as follows:

Definition 5 (Hypernode similarity). *Hypernode similarity describes the degree of similarity between community hyper-nodes. The calculation of the hypernode similarity of the community hypernode C_i and C_j is as follows:*

$$S_{C_i, C_j} = \frac{F_{C_i} F_{C_j}^T}{\sqrt{\|F_{C_i}\|^2} \sqrt{\|F_{C_j}\|^2}} \quad (6)$$

where S_{C_i, C_j} is the hypernode similarity of the community hyper-nodes C_i and C_j , and F_{C_i} and F_{C_j} are the features of F_{C_i} and F_{C_j} , which are obtained by the GLRE algorithm.

The reason for using similarity to construct a community hypergraph is as follows: a community is not isolated in the network. Its future evolution will be affected by other communities. This impact can be direct or indirect. If the two communities are more similar, they are more likely to influence each other. For example, in a social network, if two communities have the same interests, the two communities may follow each other. Therefore, even if the two communities are not connected, the evolution of one community is likely to lead to corresponding changes in the other community.

Finally, MCEP decomposes the similarity matrix S of the community hypergraph through NMF. After NMF, MCEP obtains the community mesoscopic features according to mesoscopic feature matrix. The optimization goals of NMF are as follows:

$$\min_{M \geq 0, F \geq 0} Cost = \|S - MF\|_F^2 \quad (7)$$

where S is the hypergraph similarity matrix, F is the community mesoscopic feature matrix, each row in F is the mesoscopic feature of the corresponding community, and M is the mapping matrix, which reflects the mapping relationship from S to F .

3.4. Community Evolution Prediction

This section will explain the feature set construction method used for prediction, and the overall process of MCEP.

Suppose that, for the community i at time t , the overall community features obtained by MCEP are represented by $TF_{t,i}$, and the community role features are represented by $RF_{t,i,1}$, $RF_{t,i,2}$, $RF_{t,i,3}$ (assuming that the number of roles is 3), the community mesoscopic feature is represented by $MF_{t,i}$. The final community feature is obtained by splicing the above features:

$$\mathbb{F} = [TF_{t,i}, RF_{t,i,1}, RF_{t,i,2}, RF_{t,i,3}, MF_{t,i}] \quad (8)$$

Because the future evolution of the community will be affected by its historical evolution, to take time information into account, MCEP obtains the evolution sequence by tracking the evolution process of the community. Then MCEP splices the corresponding community features and their evolution events to obtain the features for training and prediction. The feature structure of community i at time t is as follows:

$$T_{t,i} = [\mathbb{F}_{t-2,i}, E_{t-2,i}, \mathbb{F}_{t-1,i}, E_{t-1,i}, \mathbb{F}_{t,i}] \quad (9)$$

where $T_{t,i}$ represents the total feature of community i at time t , $\mathbb{F}_{t,i}$ is the feature of community i at time t , and $E_{t,i}$ is the evolution event of the community i at time t . Because MCEP uses binary classification in evolution prediction, i.e., for each evolution event, MCEP trains a separate classifier to predict. Therefore, $E_{t,i}$ has only two values: $E_{t,i}$ is 1 when the corresponding evolution event occurs, and $E_{t,i}$ is 0 when it does not occur.

After the feature set is constructed, MCEP uses the classifier for training and prediction. For each community evolution event, MCEP constructs a corresponding training set. Then MCEP uses logistic regression for training and prediction. Although MCEP uses logistic regression as the classifier, and the main focus of MCEP is the obtaining of community features and the construction of the training set, so MCEP can also use other classifiers.

The process of the MCEP algorithm is shown in Algorithm 2.

Algorithm 2 MCEP

Require: Network structure G

- 1: Split the network to obtain $G_1, G_2, \dots, G_t$
 - 2: Perform community discovery on different snapshots, and get the community structure of each snapshot $C_1, C_2, \dots, C_t$
 - 3: Use the GED algorithm to obtain the evolutionary relationship between communities
 - 4: For the community $C_{t,i}$, calculate the overall community features $TF_{t,i}$
 - 5: Use the GLRE to obtain the role features of the community $RF_{t,i,1}, RF_{t,i,2}, \dots, RF_{t,i,r}$
 - 6: Obtain the community mesoscopic feature $MF_{t,i}$ according to the algorithm described in Section 4.3
 - 7: Combine the obtained features to obtain community features $\mathbb{F}_{t,i}$
 - 8: Construct feature set according to Formula (9)
 - 9: Train the classifier based on the constructed feature set
-

4. Experiment

This section will show the effectiveness of the MCEP through experiments. In this paper, five representative algorithms are chosen as baselines. The five algorithms are CFCR [10], SGCI [8], PATOC [9], node2vec [22], and LINE [23]. For the first two algorithms, since the author did not give the abbreviations of the algorithm, for the convenience of subsequent presentation, we have named these two algorithms. In addition, since the representation of the community can also be obtained through graph representation learning, we use two well-known graph representation learning algorithms as comparison algorithms; these are node2vec and LINE. To prevent confusion with the original algorithm, we use Cnode2vec and CLINE to represent the corresponding algorithms. To use these two algorithms for community evolution prediction, after obtaining the node representation, we calculate the average value of the node representation of each community. The average value is the feature corresponding to the community; we can use these features to make predictions.

We conducted experiments on three real datasets, which are DBLP [24], ia-reality-call <https://networkrepository.com/ia-reality-call.php> (accessed on 12 March 2022), and sx-askubuntu-c2q <http://snap.stanford.edu/data/sx-askubuntu-c2q.txt.gz> (accessed on 12 March 2022). Table 3 shows the dataset used in the experiment. As can be seen from Table 3, the three networks are small, medium and large, respectively. In addition, the types of these three networks are also different, where DBLP is a citation network, ia-reality-call is a mobile phone network, and sx-askubuntu-c2q is a question-answering network. Since networks vary in size and type, we believe using these three datasets is sufficient to demonstrate the effectiveness of our proposed algorithm.

All algorithms use logistic regression as the classifier. To increase the number of community events, ia-reality-call has one snapshot overlap, and sx-askubuntu-c2q has two snapshots overlap. To ensure the stability of the experimental results, the experiment used ten-fold cross-validation and was repeated 20 times; each time, the order of the data was randomly shuffled. The final experimental result was an average of 20 times. The experiment uses accuracy as a measure.

Since the main difference of the comparison algorithm lies in feature selection, to ensure a fair comparison, all algorithms use the same steps in the first three steps of community evolution prediction. The algorithm used in the community discovery part is LFM [25]. The algorithm used in the evolution analysis part is GED [17]. The two parameters in the GED algorithm are both 0.5. Since the final training set has the problem of imbalance

between positive and negative samples, we use SMOTE [26] to balance the positive and negative samples before training the classifier.

Table 3. Datasets.

Dataset	Number of Nodes	Number of Edges	Maximum Degree	Minimum Degree	Average Degree	Density
DBLP	2423	54,001	144	1	40.52	0.01
ia-reality-call	6809	52,050	261	1	15.28	0.002
sx-askubuntu-c2q	79,155	327,513	3116	1	8.27	0.0001

4.1. Parameter Selection

MCEP has two parameters, the number of community roles K and the number of community mesoscopic features L . Obviously, choosing an appropriate number of roles and community mesoscopic features can effectively improve the effect of the algorithm. When the number of roles is too small, the algorithm cannot completely divide the community roles in the community, which will reduce the effect of the algorithm. When the number of roles is too large, the algorithm may easily divide the roles of the nodes incorrectly, and the number of mesoscopic features. Therefore, an appropriate number of roles and mesoscopic features are very important for the algorithm.

This section studies the influence of the number of roles and the number of community mesoscopic features on the MCEP. The datasets used are DBLP and ia-reality-call. The results of the experiment are shown in Tables 4 and 5.

Table 4. The accuracy of MCEP different parameters: DBLP dataset.

Event	$K = 3$				$K = 5$				$K = 7$			
	$L = 5$	$L = 10$	$L = 15$	$L = 20$	$L = 5$	$L = 10$	$L = 15$	$L = 20$	$L = 5$	$L = 10$	$L = 15$	$L = 20$
Survive	94.90%	96.29%	97.10%	97.05%	96.94%	97.21%	98.18%	97.99%	97.15%	97.30%	97.82%	97.79%
Dissolve	71.14%	72.20%	75.60%	75.70%	75.54%	74.22%	75.19%	76.02%	72.20%	73.58%	72.49%	75.27%
Grow	82.83%	84.71%	83.17%	85.43%	83.33%	85.54%	86.74%	84.81%	82.63%	83.75%	84.23%	85.64%
Merge	85.52%	83.05%	86.64%	88.05%	85.82%	87.10%	88.47%	90.31%	86.38%	86.45%	88.66%	89.05%
Shrink	95.66%	98.80%	99.41%	99.40%	98.50%	99.01%	99.44%	99.46%	99.19%	99.30%	99.47%	99.88%
Split	83.21%	84.25%	85.08%	86.75%	86.30%	86.33%	87.14%	88.93%	87.77%	87.77%	88.83%	89.89%
Aver	85.54%	86.55%	87.83%	88.73%	87.74%	88.24%	89.19%	89.59%	88.03%	88.03%	88.58%	89.59%

Table 5. The accuracy of MCEP different parameters: ia-reality-call dataset.

Event	$K = 3$				$K = 5$				$K = 7$			
	$L = 5$	$L = 10$	$L = 15$	$L = 20$	$L = 5$	$L = 10$	$L = 15$	$L = 20$	$L = 5$	$L = 10$	$L = 15$	$L = 20$
Survive	88.87%	93.44%	92.01%	94.62%	93.98%	94.27%	94.36%	94.53%	92.63%	93.56%	94.34%	94.98%
Dissolve	70.62%	72.15%	76.06%	78.48%	77.33%	78.81%	80.74%	83.46%	78.83%	79.68%	83.55%	84.16%
Grow	69.33%	69.72%	69.88%	73.84%	66.89%	68.26%	75.64%	75.11%	71.06%	70.33%	73.21%	72.27%
Merge	92.32%	95.91%	97.63%	97.52%	95.75%	96.37%	97.82%	97.78%	96.07%	97.82%	97.21%	97.89%
Shrink	68.05%	69.66%	71.35%	75.59%	75.31%	73.72%	81.92%	79.76%	76.14%	79.98%	81.57%	89.51%
Split	70.11%	71.15%	69.49%	75.07%	71.27%	79.08%	77.85%	76.43%	74.44%	74.32%	73.90%	76.26%
Aver	76.55%	78.67%	79.40%	82.52%	80.09%	82.25%	84.72%	84.51%	81.53%	82.61%	83.96%	84.84%

As shown in Tables 4 and 5, we can see that with the increase of K and L , the effect of the MCEP algorithm presents a gradual upward trend on the average. From the perspective of the number of roles, when K is 5, the MCEP algorithm can achieve the best effect. When K is greater than 5, the effect of MCEP increases slowly. Therefore, the value of K in the subsequent experiments in this paper is 5. From the perspective of the number of community mesoscopic features, when the number of roles is fixed at 5, the best community mesoscopic features on the two datasets are slightly different. On the DBLP dataset, when $L = 20$, the prediction effect is the best. On the ia-reality-call dataset, when $L = 15$, the prediction effect is the best. Regardless of the dataset, when the number of mesoscopic features

is large, the effect of MCEP is better. This may be because the community mesoscopic features involve complex relationships among communities, so MCEP needs a larger value to capture the community mesoscopic features. In the subsequent experiments, the number of features between communities is selected from 15 and 20.

From the perspective of the prediction accuracy of a single community event, the number of roles and the number of community mesoscopic features that make the evolutionary event achieve the best results are different in different events and different datasets, and there is no obvious rule. This shows that different community evolution events have their unique characteristics, so it is difficult to find a fixed pattern that can achieve the best results in all events.

4.2. Comparison Results between MCEP and Baselines

This section shows the effects of MCEP and other algorithms on three datasets. The experimental results are shown in Figures 3–8. The last column in each figure is the average value of each algorithm on the corresponding dataset. On the sx-askubuntu-c2q dataset, because the number of negative samples of survival events is too small, the prediction accuracy of the events cannot be compared. The experimental results show that the MCEP achieves the best results in most of the evolutionary events of the three datasets, which shows the effectiveness of the MCEP. For some evolutionary events, such as the survival event of ia-reality-call, the effect of MCEP algorithm is slightly worse than other algorithms. This may be because different community evolution events often present different features, and these features will also be different on the different datasets. Therefore, it is difficult to find a common feature set that enables the MCEP to have the best effect on all evolutionary events.

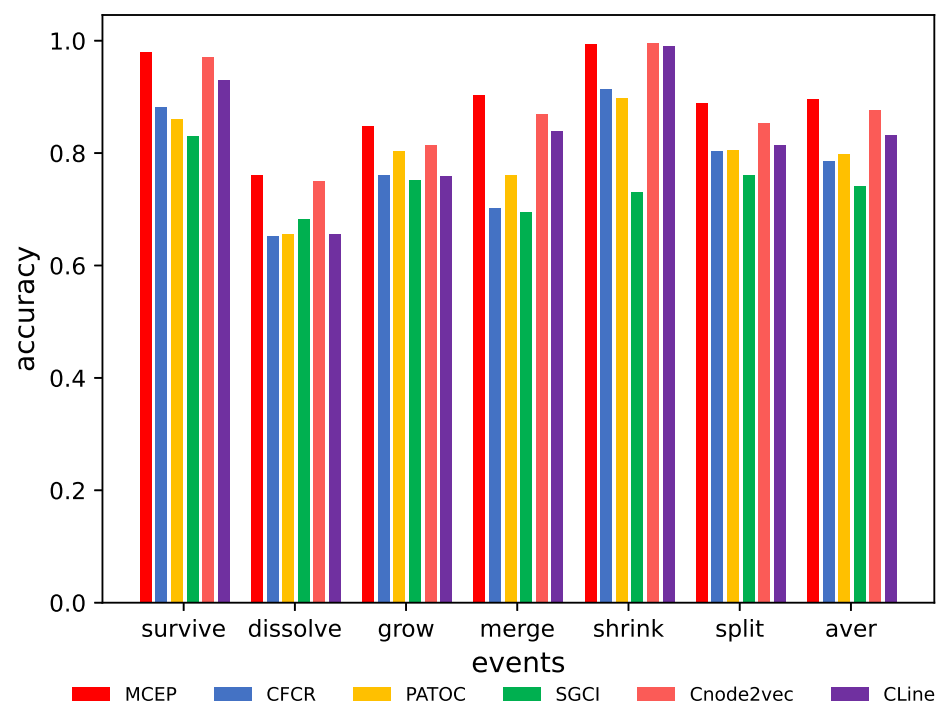


Figure 3. Comparing the accuracy of the algorithms on the DBLP dataset.

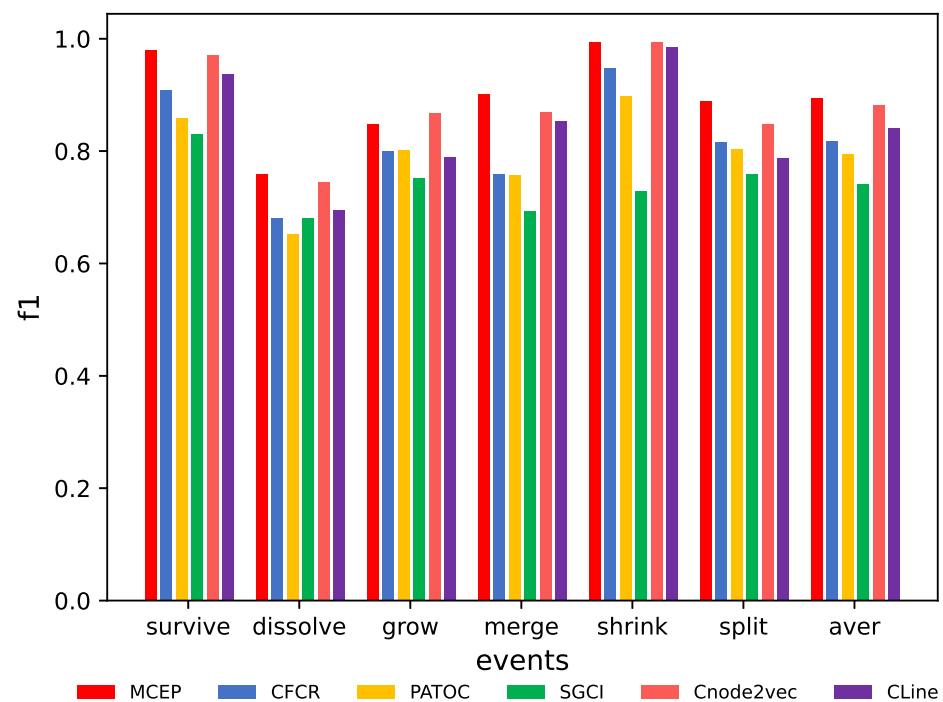


Figure 4. Comparing the f1 score of the algorithms on the DBLP dataset.

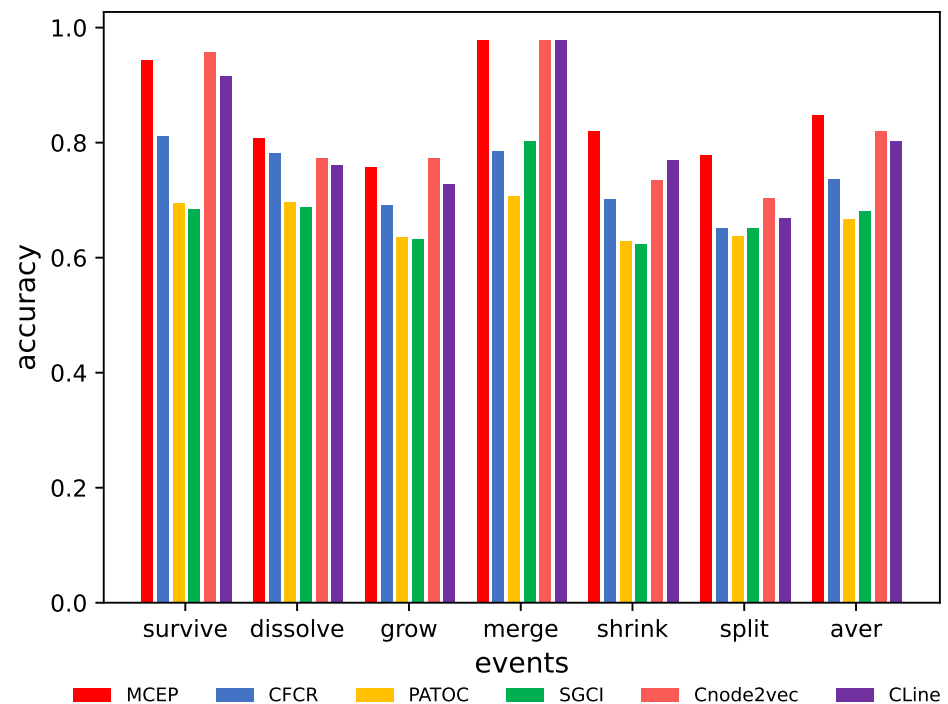


Figure 5. Comparing the accuracy of the algorithms on ia-reality-call dataset.

As can be seen from Figures 3–8, the effect of the Cnode2vec and CLINE algorithms is slightly lower than the effect of the MCEP algorithm on the three datasets. But the effect of the Cnode2vec and CLINE algorithm is better than other algorithms. The common point of MCEP, Cnode2vec, and CLINE algorithms is that these three algorithms use a relatively complex method to construct feature sets. MCEP is based on community roles and mesoscopic features, and Cnode2vec and CLINE are based on graph representation learning. In addition, the dimensionality of the features obtained by these three methods is also more than other algorithms, so it is more able to capture the characteristics of

community evolution. This shows that due to the complex characteristics of the community evolution, it is difficult to achieve a satisfactory effect for community evolution prediction only by selecting a few suitable features.

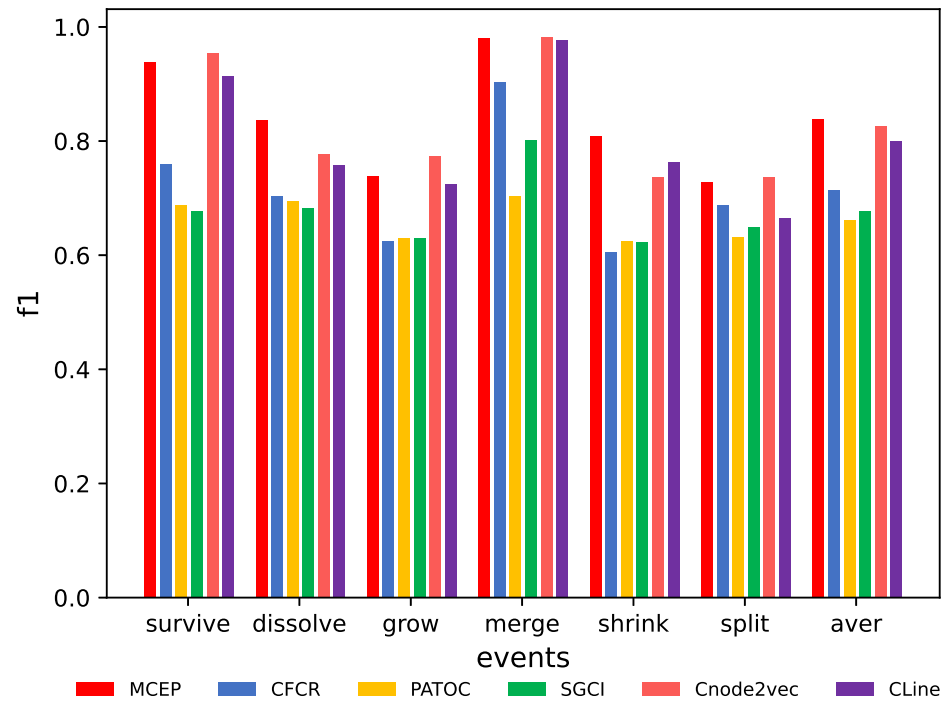


Figure 6. Comparing the f1 score of the algorithms on ia-reality-call dataset.

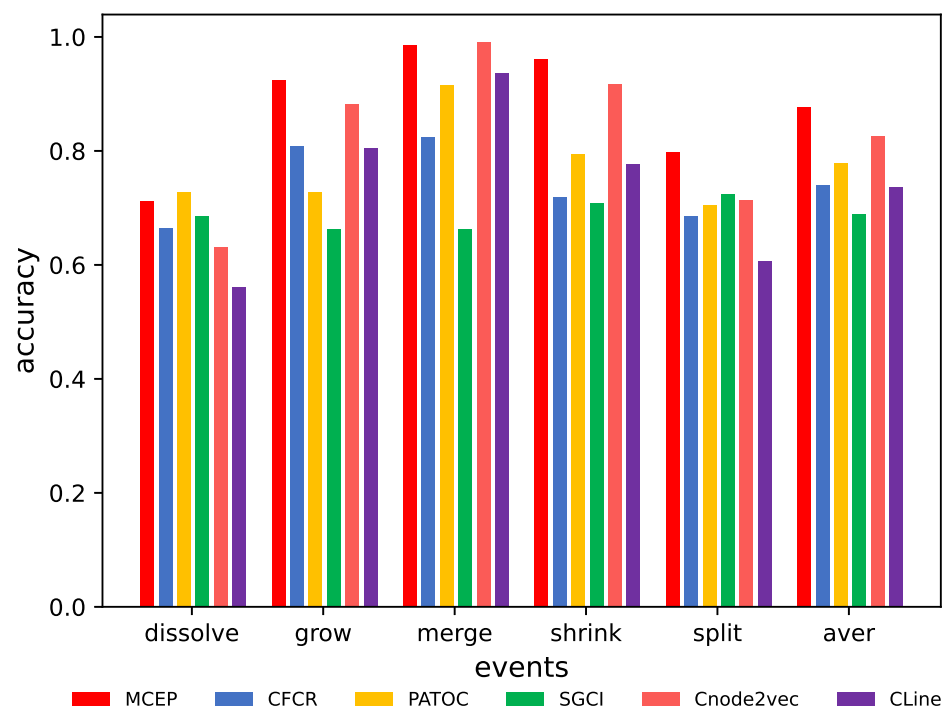


Figure 7. Comparing the accuracy of the algorithms on sx-askubuntu-c2q dataset.

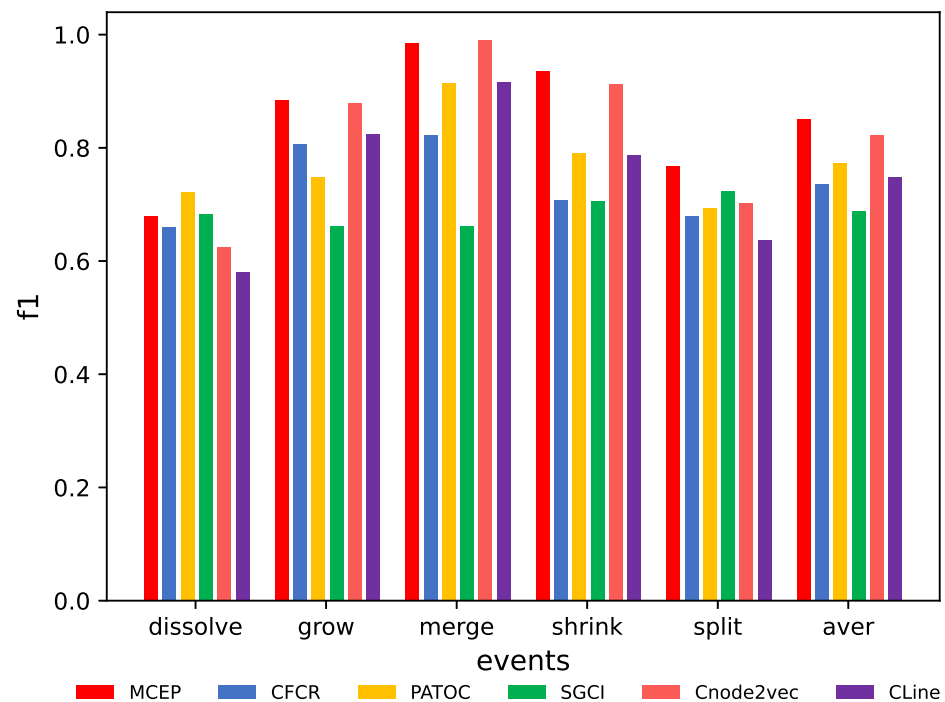


Figure 8. Comparing the f1 score of the algorithms on sx-askubuntu-c2q dataset.

4.3. Ablation Experiments

Compared to the previous algorithm, MCEP mainly made two improvements. First, MCEP proposed the GLRE for discovering the community role. Second, MCEP uses NMF to discover the community mesoscopic features. This section will prove the effectiveness of the above two improvements through ablation experiments.

First, we construct three algorithms by deleting the related improvements. In order to distinguish these three algorithms from the original MCEP algorithm, we name these algorithms as follows:

1. GL-B-MCEP: The GLRE algorithm is used, and the community mesoscopic features are considered.
2. G-B-MCEP: The traditional role-extraction algorithm is used and the community mesoscopic features are considered.
3. GL-MCEP: The GLRE algorithm is used, and the community mesoscopic features are not considered.
4. G-MCEP: The traditional role-extraction algorithm is used, and the community mesoscopic features are not considered.

Then, we conduct comparative experiments on the three datasets. To reduce the influence of other factors on the experimental results, for each algorithm, we use the same number of roles and mesoscopic features. On the DBLP dataset, the number of roles is 5 and the number of mesoscopic features is 20. On the ia-reality-call dataset and sx-askubuntu-c2q dataset, the number of roles is 5 and the number of mesoscopic features is 15. The experimental results are shown in Tables 6–8.

Table 6. Results of ablation experiments: DBLP.

Event	GL-B-MCEP	GL-MCEP	G-B-MCEP	G-MCEP
Survive	97.99%	96.48%	97.52%	95.28%
Dissolve	76.02%	72.37%	76.01%	71.48%
Grow	84.81%	81.09%	86.71%	82.39%
Merge	90.31%	85.05%	89.61%	83.45%
Shrink	99.46%	98.35%	99.02%	98.67%
Split	88.93%	85.55%	87.34%	85.26%
Aver	89.59%	86.48%	89.37%	86.09%

Table 7. Results of ablation experiments:ia-reality-call.

Event	GL-B-MCEP	GL-MCEP	G-B-MCEP	G-MCEP
Survive	94.36%	90.81%	94.71%	88.31%
Dissolve	80.74%	76.83%	83.30%	74.90%
Grow	75.64%	67.54%	75.18%	68.10%
Merge	97.82%	94.08%	97.79%	96.05%
Shrink	81.92%	73.49%	73.95%	69.64%
Split	77.85%	70.26%	74.92%	71.36%
Aver	84.72%	78.84%	83.31%	78.06%

Table 8. Results of ablation experiments: sx-askubuntu-c2q.

Event	GL-B-MCEP	GL-MCEP	G-B-MCEP	G-MCEP
Dissolve	71.16%	72.49%	70.54%	72.02%
Grow	92.42%	84.02%	89.16%	87.69%
Merge	98.45%	97.12%	97.86%	96.24%
Shrink	96.11%	91.14%	92.77%	89.49%
Split	79.68%	73.98%	72.98%	71.76%
Aver	87.56%	83.75%	84.66%	83.44%

The experiment result shows that the GL-B-MCEP algorithm can achieve the best results on the three datasets. Due to the complex characteristics of community evolution events, GL-B-MCEP did not achieve the best results in all community evolution events, which is the same as the results of the comparative experiment. But from an average point of view, the effect of GL-B-MCEP is the best.

In addition, on the DBLP dataset, the introduction of community mesoscopic features has greatly improved the prediction effect of the algorithm, while the improvement of the GLRE algorithm is not obvious. This may be because the DBLP dataset has only 2423 nodes. On this small-scale network, the local features of the community are not obvious. On the other two datasets, it can be seen that with the increase of network scale, the improvement of the prediction effect of the GLRE algorithm is gradually obvious, which shows that GLRE can be more effective on large-scale networks.

4.4. Different Community Discovery Algorithms

In the above experiment, we use LFM as the community discovery algorithm. In addition to the LFM algorithm, there are many other efficient community discovery algorithms. This section studies the performance of MCEP on several commonly used community discovery algorithms. These community discovery algorithms are the LFM algorithm [25], the Cluster Percolation method [27], the greedy algorithm [28], and the Louvain algorithm [29]. Among these four algorithms, the first two are overlapping community discovery algorithms, and the latter two are non-overlapping community discovery algorithms. The experimental results are shown in Table 9. We conducted a comparative experiment on the DBLP dataset. For the greedy algorithm and the Louvain algorithm, because the number of positive samples of shrink events is too small, there is no way to predict its accuracy, so there is no result in that part. The experimental results are shown in Table 9.

Table 9. The results of MCEP under different community algorithms.

Event	LFM	CPM	Greedy	Louvain
Survive	97.99%	81.18%	96.32%	95.88%
Dissolve	76.02%	80.78%	80.12%	87.30%
Grow	84.81%	97.15%	95.10%	87.60%
Merge	90.31%	93.39%	94.53%	94.19%
Shrink	99.46%	96.90%	-	-
Split	88.93%	99.82%	90.80%	91.92%
Aver	89.59%	91.54%	91.38%	91.38%

From Table 5, we can see that different community discovery algorithms have a significant impact on the results of MCEP. For example, when using the CPM algorithm, MCEP prediction accuracy of survival events is significantly lower than the other three algorithms, and its prediction accuracy of split events is significantly higher than other algorithms. This fact shows that the communities discovered by different community discovery algorithms have their characteristics. The choice of community discovery algorithms will also have a significant impact on the prediction effect. Therefore, when making community evolution prediction, the choice of community discovery algorithm should be decided according to specific conditions.

4.5. Dimensionality Reduction

In the above experiment, we do not restrict the final dimension of the training set obtained by each algorithm, i.e., the feature dimension of the training set obtained by the above experiment is different. Different algorithms use different methods for obtaining the training set, so it is normal for the dimensions of the training set to be different.

This section compares the performance of different algorithms when the dimensions of the training set are the same. The specific method is that after obtaining the final features, we use PCA to reduce the dimensionality of the obtained features. The result after dimensionality reduction is used for prediction. We compare the effects of different algorithms in 5 different dimensions. We conducted experiments on the DBLP dataset and the ia-reality-call dataset. The experimental results are shown in Figures 9 and 10, which only shows the average value of the final prediction results. In addition, since the dimensionality of the training set obtained by the SGCI algorithm is only 15 dimensions, when the dimensions exceed 15 dimensions, we set the effect of SGCI to be the same as the effect of 15 dimensions.

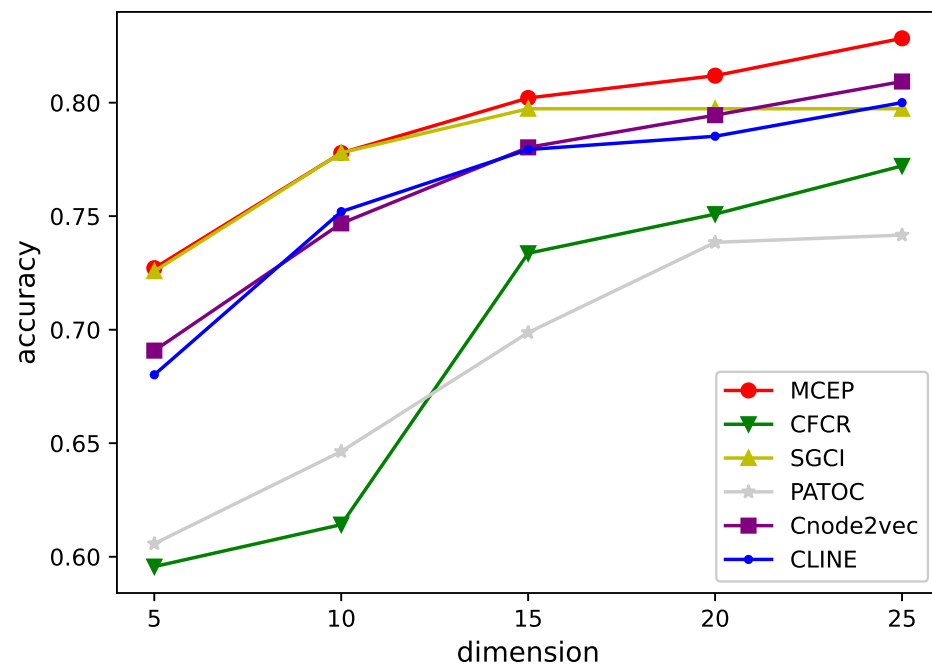


Figure 9. Comparison of algorithm effects in different dimensions: DBLP.

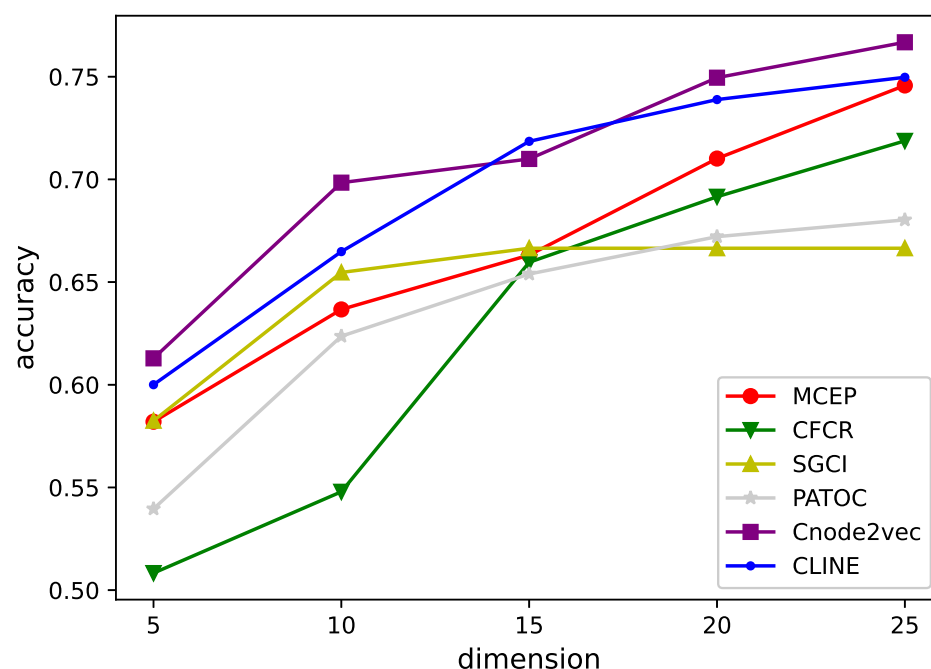


Figure 10. Comparison of algorithm effects in different dimensions: ia-reality-call.

5. Conclusions and Future Work

In this paper, we studied a novel community evolution prediction algorithm. The existing community evolution prediction algorithm only obtains the features of the community from a single level and then uses it for community evolution prediction. These approaches are neither comprehensive nor detailed enough, so they cannot obtain good results. To solve these problems, the MCEP algorithm is proposed to obtain features from two layers: the node layer, and the community layer. At the node layer, we propose the global and local community role-extraction algorithm. MCEP can accurately divide the nodes with different characteristics in the community using GLRE. Second, at the community layer, MCEP discovers community mesoscopic features by constructing a community hypergraph and using NMF. The related experimental results verify the effectiveness of the MCEP algorithm.

Although the MCEP can improve the effectiveness of community evolution prediction, the algorithm still has some shortcomings. First, in the stage of calculating community features, the algorithm needs to manually select relevant features. Obviously, this is a very inefficient way, and it is difficult to obtain the best results. Second, the MCEP algorithm has two parameters: the number of roles, and the number of mesoscopic features. These two parameters also need to be set in advance. In many cases, choosing an optimal parameter is quite difficult. Therefore, how to automatically select effective features and how to automatically select algorithm parameters is the future research direction from this paper.

Author Contributions: Conceptualization, Z.W. and W.L.; Data curation, Q.X.; Formal analysis, Z.W.; Funding acquisition, W.L.; Investigation, Q.X.; Methodology, Z.W.; Project administration, W.L.; Resources, Z.W.; Software, Z.W.; Supervision, Q.X. and W.L.; Validation, Q.X.; Visualization, Q.X.; Writing—original draft, Z.W.; Writing—review & editing, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research presented in this paper is supported by the National Key R&D Program of China (No. 2017YFE0117500) and the National Natural Science Foundation of China (No. 61762002).

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <https://networkrepository.com/ia-reality-call.php> (accessed on 12 March 2022); <http://snap.stanford.edu/data/sx-askubuntu-c2q.txt.gz> (accessed on 12 March 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Girvan, M.; Newman, M.E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [[CrossRef](#)] [[PubMed](#)]
- Li, J.; Cai, T.; Deng, K.; Wang, X.; Sellis, T.; Xia, F. Community-diversified influence maximization in social networks. *Inf. Syst.* **2020**, *92*, 101522. [[CrossRef](#)]
- Cai, T.; Li, J.; Mian, A.S.; Sellis, T.; Yu, J.X. Target-aware holistic influence maximization in spatial social networks. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 1993–2007. [[CrossRef](#)]
- Zhuang, D.; Chang, J.M.; Li, M. DynaMo: Dynamic community detection by incrementally maximizing modularity. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 1934–1945. [[CrossRef](#)]
- Xu, Z.; Rui, X.; He, J.; Wang, Z.; Hadzibeganovic, T. Superspreaders and superblockers based community evolution tracking in dynamic social networks. *Knowl.-Based Syst.* **2020**, *192*, 105377. [[CrossRef](#)]
- Li, W.; Zhu, H.; Li, S.; Wang, H.; Dai, H.; Wang, C.; Jin, Q. Evolutionary community discovery in dynamic social networks via resistance distance. *Expert Syst. Appl.* **2021**, *171*, 114536. [[CrossRef](#)]
- Calvó-Armengol, A.; Zenou, Y. Social networks and crime decisions: The role of social structure in facilitating delinquent behavior. *Int. Econ. Rev.* **2004**, *45*, 939–958. [[CrossRef](#)]
- Gliwa, B.; Bródka, P.; Zygmunt, A.; Saganowski, S.; Kazienko, P.; Koźlak, J. Different approaches to community evolution prediction in blogosphere. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Niagara Falls, ON, Canada, 25–28 August 2013; pp. 1291–1298.
- Shahriari, M.; Gunashekar, S.; Domarus, M.V.; Klamma, R. Predictive analysis of temporal and overlapping community structures in social media. In Proceedings of the 25th International Conference Companion on World Wide Web, Montreal, QC, Canada, 11–15 May 2016; pp. 855–860.
- Dakiche, N.; Tayeb, F.B.S.; Slimani, Y.; Benatchba, K. Community evolution prediction in dynamic social networks using community features' change rates. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol, Cyprus, 8–12 April 2019; pp. 2078–2085.
- Bródka, P.; Kazienko, P.; Kołoszczyk, B. Predicting group evolution in the social network. In Proceedings of the International Conference on Social Informatics, Alexandria, VA, USA, 14–16 December 2012; pp. 54–67.
- Takaffoli, M.; Rabbany, R.; Zaiane, O.R. Incremental local community identification in dynamic social networks. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), Niagara Falls, ON, Canada, 25–28 August 2013.
- Diakidis, G.; Karna, D.; Fasarakis-Hilliard, D.; Vogiatzis, D.; Paliouras, G. Predicting the evolution of communities in social networks. In Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics, Larnaca, Cyprus, 13–15 July 2015; pp. 1–6.
- İlhan, N.; Ögüdücü, Ş.G. Predicting community evolution based on time series modeling. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Paris France, 25–28 August 2015; pp. 1509–1516.
- İlhan, N.; Ögüdücü, Ş.G. Feature identification for predicting community evolution in dynamic social networks. *Eng. Appl. Artif. Intell.* **2016**, *55*, 202–218. [[CrossRef](#)]
- Henderson, K.; Gallagher, B.; Eliassi-Rad, T.; Tong, H.; Basu, S.; Akoglu, L.; Koutra, D.; Faloutsos, C.; Li, L. Rolx: Structural role extraction & mining in large graphs. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 1231–1239.
- Bródka, P.; Saganowski, S.; Kazienko, P. GED: The method for group evolution discovery in social networks. *Soc. Netw. Anal. Min.* **2013**, *3*, 1–14. [[CrossRef](#)]
- Watts, D.J.; Strogatz, S.H. Collective dynamics of 'small-world' networks. *Nature* **1998**, *393*, 440–442. [[CrossRef](#)] [[PubMed](#)]
- Freeman, L.C. A set of measures of centrality based on betweenness. *Sociometry* **1977**, *40*, 35–41. [[CrossRef](#)]
- Newman, M.E.J. The mathematics of networks. *New Palgrave Encycl. Econ.* **2008**, *2*, 1–12.
- Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [[CrossRef](#)] [[PubMed](#)]
- Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
- Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
- Bhat, S.Y.; Abulaish, M. HOCTracker: Tracking the evolution of hierarchical and overlapping communities in dynamic social networks. *IEEE Trans. Knowl. Data Eng.* **2014**, *27*, 1013–1019. [[CrossRef](#)]
- Lancichinetti, A.; Fortunato, S.; Kertész, J. Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* **2009**, *11*, 033015. [[CrossRef](#)]
- Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [[CrossRef](#)]
- Palla, G.; Derényi, I.; Farkas, I.; Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **2005**, *435*, 814–818. [[CrossRef](#)] [[PubMed](#)]

-
28. Clauset, A.; Newman, M.E.; Moore, C. Finding community structure in very large networks. *Phys. Rev. E* **2004**, *70*, 066111. [[CrossRef](#)] [[PubMed](#)]
 29. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [[CrossRef](#)]