



Multiple clusterings of heterogeneous information networks

Shaowei Wei¹ · Guoxian Yu^{1,2}  · Jun Wang³ · Carlotta Domeniconi⁴ ·
Xiangliang Zhang⁵

Received: 21 November 2020 / Revised: 12 May 2021 / Accepted: 14 May 2021 /

Published online: 2 June 2021

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2021

Abstract

Traditional clustering algorithms focus on a single clustering result; as such, they cannot explore potential diverse patterns of complex real world data. To deal with this problem, approaches that exploit meaningful alternative clusterings in data have been developed in recent years. Existing algorithms, including single view/multi-view multiple clustering methods, are designed for applications with i.i.d. data samples, and cannot handle the data samples with dependency presented in networks, especially in heterogeneous information networks (HIN). In this paper, we propose a framework (NetMCs) that can explore multiple clusterings in HIN. Specifically, NetMCs adopts a set of meta-path schemes with different semantics on HIN, and considers each meta-path scheme as a base clustering aspect. Guided by the meta-path schemes, NetMCs then introduces a variation of the skip-gram framework that can jointly optimize multiple clustering aspects, and simultaneously obtain the respective embedding representations and individual clusterings therein. To reduce redundancy between alternative clusterings, NetMCs utilizes an explicit regularization term to control the embedding diversity of the same nodes among different clustering aspects. Experiments on benchmark HIN datasets confirm the performance of NetMCs in generating multiple clusterings with high quality and diversity.

Keywords Multiple clusterings · Heterogeneous information networks · Meta-path · Quality and diversity · Network embedding

1 Introduction

Clustering is an unsupervised learning task whose aim is to partition the data into a number of homologous clusters. Traditional clustering methods typically provide a single clustering, and fail to reveal the diverse patterns underlying the data. In fact, because of the multiplexes of real world data, several different clusterings may co-exist in a given dataset, and each may provide a meaningful grouping of the data (Bailey 2013; Niu et al. 2013;

Editors: Annalisa Appice, Sergio Escalera, Jose A. Gamez, Heike Trautmann.

✉ Guoxian Yu
guoxian85@gmail.com

Extended author information available on the last page of the article

Wang et al. 2018). For instance, a collection of films can be grouped based on genre, producer or on director; the authors in an academic network can be clustered by their research fields or by their organizations. These alternative clusterings are all meaningful. To mine the underlying structure of the data from different perspectives and present alternative clusterings of different aspects, the study of *multiple clusterings* has emerged during the last decade (Bailey 2013). Multiple clustering approaches not only focus on the quality of the clusterings, but also on their diversity. However, it is a known dilemma to reach a balance between quality and diversity (Bailey 2013; Wang et al. 2020).

The early approaches to multiple clusterings mainly focus on single-view data to generate alternative clusterings in non-redundant (independent) subspaces (Cui et al. 2007; Niu et al. 2013; Mautz et al. 2018; Wang et al. 2019b; Miklautz et al. 2020), by meta clustering of multiple base clusterings (Caruana et al. 2006), by reducing the redundancy with the already generated clusterings (Bae and Bailey 2006; Yang and Zhang 2017), or by simultaneously reducing the redundancy between all the to-be-generated clusterings (Wang et al. 2018; Yao et al. 2019b). Recently, some works extend multiple clusterings to multi-view data, which are naturally represented with heterogeneous feature views. For example, a film can be encoded by its audio, video and snapshots. Multi-view multiple clusterings (Yao et al. 2019a) first adapts self-representation learning (Luo et al. 2018) to extract the individuality and commonality information matrices of multi-view data, and then applies semi-nonnegative matrix factorization (Ding et al. 2010) on each combination of an individuality (for diversity) and the shared commonality (for quality) matrices to generate alternative clusterings. Deep matrix factorization based multi-view multiple clusterings (Wei et al. 2020b) factorizes the multi-view data matrices into multiple common subspaces layer-by-layer, and generates an alternative clustering per layer. Deep incomplete multi-view multiple clusterings (Wei et al. 2020a) seeks multiple clusterings by completing the missing data with multiple decoding networks. All these single-/multi-view multiple clusterings algorithms are designed for applications with i.i.d. data samples, and cannot handle the data samples with dependency presented with linkages in networks.

In the real world, many complex systems take the form of networks (Cui et al. 2019; Zhang et al. 2020; Yang et al. 2020), where the samples are nodes with dependency on others to some degree, reflected by the seen/unseen links between them. There are homogeneous networks with *only* one type of nodes and relationships. Yet a large number of networks are heterogeneous in nature, involving diverse types of node and/or relationships between nodes, such as social networks, biological networks, and academic networks. Traditional approaches perform clustering on networks by spectrum method (Mall et al. 2013; Li et al. 2019), by ranking learning (Sun et al. 2009; Chen et al. 2015), by matrix factorization method (Lin et al. 2016), by hierarchical approach (Pio et al. 2018), or by representation learning (Perozzi et al. 2014; Grover and Leskovec 2016; Dong et al. 2017). These methods only focus on a single clustering assignment for networks, while in reality a node in networks usually has multiple clustering assignments. As illustrated in Fig. 1, nodes (e.g., authors) in an academic publication network can be grouped by their research fields and by their organizations. Ensemble clustering also aims to generate diverse base clusterings to reach a consolidated clustering, but it still targets to output a single clustering. In addition, these base clusterings are often with high redundancy.

Several attempts have been made to generate multiple vector representations for each network node. Splitter (Epasto and Perozzi 2019) splits original network into multiple ego-networks and then learns a new representation from each ego-network. MNE (Yang et al. 2018) factorizes the network proximity matrix into several groups of embedding matrices to generate different representations. These two methods are designed

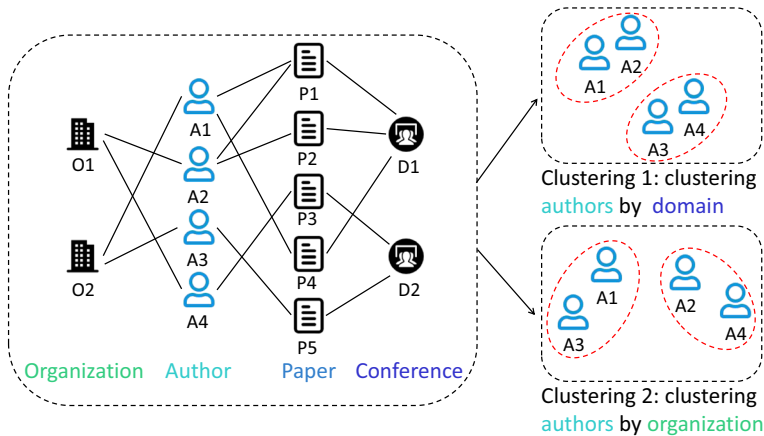


Fig. 1 An example of the academic heterogeneous network. The authors can be clustered according to alternative meaningful patterns (i.e., research areas and organizations)

for homogeneous network. ASPeM (Shi et al. 2018a) decomposes a HIN into multiple aspects and then learns representations of HIN. HEER (Shi et al. 2018b) embeds HIN via edge representations. But HEER defines the aspects based on the edges with predefined ground-truth labels, which are not always available in reality. Although these multi-facet network embedding methods can produce multiple representations, in which multiple clustering results can be generated consequently, they are not designed for multiple clusterings. Furthermore, they suffer the optimization inconsistency and the difficulty of redundancy control between alternative clusterings.

We propose an approach called NetMCs to explore multiple clusterings in HIN. NetMCs first adopts a set of meta-path schemes with different semantics on HIN and considers each meta-path scheme as a base clustering aspect. Then, guided by the set of meta-path schemes, NetMCs introduces a variation of the skip-gram framework that can jointly optimize multiple clustering aspects, and obtain respective embedding representations and individual clusterings therein. In addition, NetMCs explicitly controls the embedding diversity of the same node between different clustering aspects, and thus enhances the diversity between alternative clusterings. The main contributions of our work are summarized as follows:

- (i) To the best of our knowledge, NetMCs is the first effort to generate multiple clusterings with quality and diversity from a heterogeneous information network, which is an important and practical topic, but quite challenging and mostly overlooked by previous solutions.
- (ii) NetMCs introduces a variation of the skip-gram model to jointly optimize different clustering aspects to learn multiple diverse embeddings, and to generate multiple clusterings therein. The corresponding optimization procedure for the variation is also presented. As a result, NetMCs addresses the optimization inconsistency between representation learning and clustering, and can generate multiple clusterings of quality.

- (iii) NetMCs introduces the redundancy terms that can simultaneously minimize the overlap between embeddings and place nodes with similar embeddings into the same cluster, and therefore generates alternative clusterings of diversity.
- (iv) Experiments on real-world datasets and visualization examples demonstrate the effectiveness of NetMCs on mining multiple clusterings on heterogeneous networks.

2 Related works

Our work has close connections with three lines of related works, that is multiple clusterings, network clustering and multi-facet network embedding learning.

Multiple clusterings focuses on how to generate different clusterings with both high quality and diversity from the same dataset (Bailey 2013). It is less well studied than single/multi-view clustering and ensemble clustering (Jain 2010; Zhou 2012), especially network clustering, because of its demand on generating multiple groups of clusters, and the difficulties on well tradeoff the quality and diversity at the mean time. Based on hierarchical clustering, Bae and Bailey (2006) presented a multiple clusterings solution (COALA). The main idea of COALA is that instances with higher intra-class similarity in the first clustering still gather in the same cluster, while those with lower intra-class similarity are considered to be placed into different clusters for the next clustering. Jain et al. (2008) proposed Dec-kmeans to find multiple sets of mutually orthogonal cluster centroids and then generate diverse clusterings based on these centroids. Different from COALA and Dec-kmeans that directly control the diversity between clustering assignment, other solutions control the diversity between clustering subspaces and then generate different clusterings in these subspaces. Cui et al. (2007) projected the data matrix into orthogonal subspaces to get different feature representations and then found alternative clusterings in these subspaces. Mautz et al. (2018) also attempted to explore multiple mutually orthogonal subspaces, along with the optimization of classical k-means objective function, to find non-redundant clusterings. However, the orthogonal constraint is too strict to generate more than two clusterings. Wang et al. (2019b) generated multiple independent subspaces with semantic interpretation via independent subspace analysis and minimum description length, and then performed kernel matrix factorization-based clustering in these subspaces to explore diverse clusterings. Miklautz et al. (2020) combined the benefits of a deep neural network-based non-linear feature transformation with a non-redundant clustering objective to gain alternative clusterings. Yang and Zhang (2017) explicitly introduced a regularization term to quantify and minimize the redundancy between the already generated clustering and the to-be-generated one, and then plugged this regularization as the constraint of the next clustering based on nonnegative matrix decomposition (Ding et al. 2010) to find another clustering. Wang et al. (2018) and Yao et al. (2019b) directly reduced the redundancy between all the to-be-generated clusterings to simultaneously find all clusterings. Besides, Caruana et al. (2006) firstly generated a number of high-quality clusterings, and then grouped these clusterings at the meta-level, and thus allowed the user to select desired non-redundant clusterings for application. These multiple clusterings methods are designed only for **single-view** data.

Given the multiplicity of multi-view data, it is desirable but more difficult to generate multiple clusterings from the same *multi-view* data. Three approaches have been proposed for attacking this challenging task. MVMC (Yao et al. 2019a) first explores multiple clusterings on multi-view data by mining the individuality information encoding matrices and

the commonality information matrix shared across views by self-representation learning (Luo et al. 2018). It employs each individuality similarity matrix and the commonality similarity to generate a distinct clustering by matrix factorization-based clustering. However, given the cubic time complexity of the self-representation learning, MVMC can hardly be applicable on datasets with a large number of samples. To alleviate this drawback, DMClusts extends the deep matrix factorization (Trigeorgis et al. 2016) to collaboratively factorize the multi-view data matrices into multiple representational subspaces layer-by-layer, and seeks a different clustering of high quality per layer. In addition, it introduces a new balanced redundancy quantification term to enhance the diversity among these clusterings, and thus reduces the overlap between the produced clusterings. DiMVMC (Wei et al. 2020a) considers the incomplete multi-view data, which achieves the completion of data view and multiple shared representations simultaneously by optimizing multiple groups of decoder deep networks. It further minimizes a redundancy term to simultaneously control the diversity among these representations and among parameters of different networks to generate individual clusterings.

All these single-/multi-view multiple clusterings algorithms are vector-based methods, which assume that data samples can be directly represented by independent feature vectors. As such, they cannot be applied directly for extensive network data with inter-dependence. With the increase of network data, many network clustering works have been proposed. To name a few, Li et al. (2019) formulated the similarity matrix construction as an optimization problem and applies spectral clustering to network data. Lin et al. (2016) transformed different types of relations into a group of matrices and subsequently combined with a greedy search approach. Chen et al. (2015) proposed a probabilistic generative model that simultaneously achieves clustering and ranking on a heterogeneous network with arbitrary network schema. Zhou et al. (2019) proposed a recurrent meta-structure based framework to measure the similarity between nodes by integrating all the meta-paths and meta-structures, and applied it for clustering and ranking task on HIN. HENPC (Pio et al. 2018) extracts possibly overlapping and hierarchically-organized heterogeneous clusters and exploits them for predictive purposes, and it can take into account autocorrelation of HIN at different levels of granularity. Random walk based methods (Perozzi et al. 2014; Grover and Leskovec 2016; Dong et al. 2017) firstly obtain a pile of random walk sequences and subsequently put them into skip-gram framework to learn the node embedding, along with traditional clustering algorithms on the embedding. These methods only generate a single embedding or clustering for one network and lack of consideration for redundancy control.

Some multi-facet network embedding methods have been introduced, most of which focus on generating multiple vector representations for each network node. Based on a principled decomposition of the ego-network, Splitter (Epasto and Perozzi 2019) splits each node into multiple representations by performing local graph clustering, each representation encodes the role of the node in a different local community in which the nodes participate. MNE (Yang et al. 2018) factorizes the network proximity matrix into several groups of embedding matrices, and adds a diversity constraint to force different matrices focusing on different aspects of nodes. ASPEM (Shi et al. 2018a) decomposes a HIN into multiple aspects before learning embedding to obtain quality representations of HIN. HEER (Shi et al. 2018b) embeds HIN via edge representations that are further coupled with properly-learned heterogeneous metrics to capture the incompatible semantics of HIN.

Although these network embedding methods can generate different representation vectors, they cannot produce clustering results directly and thus suffer the optimization inconsistency between representation learning and clustering. Furthermore, they neglect the redundancy control among clusterings. To address these issues, our NetMCs generate

diverse HIN node embeddings by different meta-path based semantics and alternative clusterings based on these embeddings in a coherent way.

3 Our method

3.1 Preliminaries

For problem formulation, we first give related concepts and notations.

Definition 1 (HIN) A *Heterogeneous Information Network* is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, in which each node v and each edge e are associated with their mapping functions $\phi(v) : \mathcal{V} \rightarrow \mathcal{T}_V$ and $\psi(e) : \mathcal{E} \rightarrow \mathcal{T}_E$ respectively. \mathcal{T}_V and \mathcal{T}_E denote the sets of nodes and edge types, where $|\mathcal{T}_V| + |\mathcal{T}_E| \geq 2$.

Definition 2 (Meta-path) a *meta-path scheme* \mathcal{P} is defined as a path in the form of $V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \dots V_i \xrightarrow{R_i} V_{i+1} \dots \xrightarrow{R_{l-1}} V_l$, wherein $R_i \in \mathcal{T}_E$ represents an edge type and $R = R_1 \circ R_2 \circ \dots \circ R_{l-1}$ defines the composite relations between node types V_1 and V_l .

Definition 3 (Multiple clusterings on a HIN) Given a HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, a set of meta-paths $\mathcal{PS} = \{\mathcal{P}_m\}_{m=1}^M$, and a set of the number of clusters $\mathcal{K} = \{k_m\}_{m=1}^M$, the problem of multiple clusterings on \mathcal{G} is to partition nodes into M diverse clustering patterns. The nodes in the m -th clustering are partitioned into k_m disjoint clusters $\mathcal{C}_m = \{\mathcal{C}_{m,1}, \dots, \mathcal{C}_{m,k_m}\}$.

3.2 The proposed methodology

Random walk is a powerful approach for capturing information in networks, and extensively used for network embedding (Perozzi et al. 2014; Grover and Leskovec 2016; Dong et al. 2017). Considering its efficiency and effectiveness in handling large-scale networks, we adopt a random walk approach as the base of NetMCs. Thus, we briefly introduce random walk-based network embeddings first.

3.2.1 Generating node embeddings

Given a text corpus, Mikolov et al. (2013) proposed the word2vec to learn the distributed representations of words in the corpus. Inspired by it, previous network representation learning methods, such as DeepWalk (Perozzi et al. 2014) and node2vec (Grover and Leskovec 2016) view nodes in a graph as words in a corpus. Specifically, they both perform random walks on a graph and obtain a set of truncated random walks sequences \mathcal{W} . Then, the skip-gram model is introduced to learn the representation of each node. The objective is to maximize the likelihood of the context given the target node:

$$\max \prod_{u \in \mathcal{V}} \prod_{v \in \mathcal{N}(u)} p(v|u) \quad (1)$$

where $\mathcal{N}(u)$ denotes the neighbors of target node u and $p(v|u)$ is commonly defined as a softmax function as:

$$p(v|u) = \frac{\exp(\langle \mathbf{h}_u, \mathbf{h}_v \rangle)}{\sum_{v_r \in \mathcal{V}} \exp(\langle \mathbf{h}_u, \mathbf{h}_{v_r} \rangle)} \quad (2)$$

where \mathbf{h}_u represents the embedding vector for node u , \mathbf{h}_v represents the context embedding vector for node v and $\langle \mathbf{h}_u, \mathbf{h}_v \rangle$ means the inner product of \mathbf{h}_u and \mathbf{h}_v . By maximizing Eq. (1), nodes that frequently appear together within a context window will be trained to have similar embeddings. Equation (1) is limited to homogeneous networks. In other words, it cannot leverage the information of multi-type of nodes in HIN.

Meta-path can capture the structural and semantic correlations between different types of nodes in HIN (Sun and Han 2013). It is defined by a sequence of relations in the network, and can be described by a sequence of object types when there is no ambiguity. For example, A-O-A in Fig. 1 is a meta-path denoting the colleague relation between authors, and A-P-C is a meta-path denoting the publication relationship between the authors and conferences. `metapath2vec` (Dong et al. 2017) introduces a heterogeneous skip-gram model to perform meta-path based random walk in heterogeneous networks to learn the representation of the nodes. Formally, given a heterogeneous network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ with $|\mathcal{T}_v| > 1$, `metapath2vec` learns the representation of node u by maximizing the probability of having the heterogeneous context $\mathcal{N}_t(u)$ ($t \in \mathcal{T}_v$) generated by meta-path-based random walk:

$$\max_u \sum_{t \in \mathcal{T}_v} \sum_{v_j \in \mathcal{N}_t(u)} \log p(v|u) \quad (3)$$

where $\mathcal{N}_t(u)$ includes u 's neighbors with the t -th type of nodes.

For the work in this paper, we follow the concept of meta-path to describe the possible relations that can be derived from a heterogeneous network between different types of objects in a meta level. Most meta-path based methods usually use only one meta-path in a specific task. However, the semantic included in a single meta-path is too limited to mine the abundant information in HIN. To solve this problem, Zhao et al. (2017) introduced the concept of meta-graph which is an extension of meta-paths. Compared with the sequence structure of meta-path, meta-graph does not restrict the intermediate linked structure between the source and target nodes. Wang et al. (2019a) merged multiple meta-paths to learn the node similarity via weighting the meta-paths. Nevertheless, while both the meta-graph and merged meta-paths strive for richer semantic information, they may encounter the semantic incompatibility problem, due to heterogeneity of HIN (Shi et al. 2018b). Semantic incompatibility refers to the semantic inconsistency of meta paths with respect to the same node. If the semantics of these meta-paths are inconsistent, then projecting the related nodes into a uniform embedding space will result in the degradation of the embedding vectors. For example, Bob likes both musics and movies directed by Nolan. If these nodes were embedded to one metric space, Bob would be close to neither musics nor Nolan due to the dissimilarity between musics and Nolan, which results in information loss. Therefore, we adopt multiple different meta-paths to perform random walks on HIN to capture diverse semantic information. Specifically, guided by a set of diverse meta-paths \mathcal{PS} , we implement random walks on HIN. Each meta-path $\mathcal{P}_m \in \mathcal{PS}$ can generate truncated random walk sequences $\mathcal{W}_{\mathcal{P}_m}$, which are subsequently used as the input for the variant skip-gram framework. Thus, NetMCs generates an embedding for nodes in each selected meta-path as follows:

$$\max \sum_{\mathcal{P}_m \in \mathcal{PS}} \sum_{u \in \mathcal{W}_{\mathcal{P}_m}} \sum_{v \in \mathcal{N}_m(u)} \log \frac{\exp(\langle \mathbf{h}_u^m, \mathbf{h}_v^m \rangle)}{\sum_{v' \in \mathcal{V}} \exp(\mathbf{h}_u^m, \mathbf{h}_{v'}^m)} \tag{4}$$

where $\mathbf{h}_u^m \in \mathbb{R}^d$ is the embedding vector of node u in terms of the meta-path \mathcal{P}_m , and $\mathcal{N}_m(u)$ denotes the nodes within the context window of u . By optimizing the above equation, we can obtain individual embeddings of each node under different semantics guided by diverse meta-paths.

3.2.2 Enhancing diversity and quality

However, the semantics of different meta-paths may overlap. for instance, A-O-A and A-P-A may have the same semantics to some extent, because authors in the same institution are more likely to co-author, which may cause similar clustering results. We expect embeddings induced from different meth-paths being different. Thus, we introduce a regularization term to quantify and minimize the overlap between embedding vectors of the same node by their similarity of probability distributions. In addition, NetMCs should make each embedding space having different semantic information as much as possible and capture information from other embedding spaces also. In this way, a clustering is different from others by reducing the redundancy between embedding spaces respectively induced by individual meta-paths. For M different embeddings of node u , the regularization term is defined as:

$$\min \sum_{m'=1, m' \neq m}^M \log p(\mathbf{h}_u^{m'} | \mathbf{h}_u^m) \tag{5}$$

We can now jointly optimize Eqs. (4) and (5) to obtain multiple embeddings of the same node in the HIN and then execute clustering in the respective embedding spaces. However, as we pointed out, the sequential embedding and clustering paradigm may lead to optimization inconsistency, due to the distinct goals of embedding and clustering (this is confirmed in our experiments). Thus, we try to achieve multiple embeddings and multiple clusterings therein simultaneously by employing a transformation $f(\mathbf{h}_u^m)$ that generates a node’s clustering assignment from its embedding. To keep the model from getting too complicated, we simply adopt $\mathbf{z}_u^m = \text{softmax}(\mathbf{Q}_m \mathbf{h}_u^m)$ in this paper, where $\mathbf{z}_u^m \in \mathbb{R}^{k_m}$ is the soft assignment vector of node u in the m -th clustering, and $\mathbf{Q}_m \in \mathbb{R}^{k_m \times d}$ is a transformation matrix for the m -th clustering aspect. Because each meta-path can generate an embedding space and reflect a semantic pattern of HIN, we generate a clustering result in each embedding space. Thus, the number of clusterings M is equal to the number of meta-paths.

In addition, we assume that the assignment vectors of similar embeddings should be similar. Cross entropy is a common evaluation index in multi-classification tasks; it can measure the difference between two probability distributions. Therefore, we can naturally adopt the cross entropy metric to measure the similarity between different soft assignment vectors. Given a node u and its neighborhood v in the m -th clustering aspect, the cross entropy is minimized as:

$$\min \quad CE(\mathbf{z}_u^m, \mathbf{z}_v^m) = - \sum_{i=1}^{k_m} (\mathbf{z}_u^m)_i \log(\mathbf{z}_v^m)_i \tag{6}$$

where $(\mathbf{z}_u^m)_i$ is the i -th element of \mathbf{z}_u^m , and $v \in \mathcal{N}_m(u)$. By minimizing the cross entropy between the embedding vectors with high similarity, NetMCs can constrain these nodes also having similar clustering assignment and thus improve the clustering quality.

3.2.3 Unified objective

By integrating the above objectives we can define the comprehensive loss function of NetMCs. The optimization of Eq. (4) is computationally expensive, which requires the summation over the entire set of vertices in $\{\mathcal{W}_{P_m}\}_{m=1}^M$ when calculating the softmax function. In addition, the skip-gram model has a very large number of parameters. As a result, it is time-demanding to train such a large neural network using gradient descent. Furthermore, we need a lot of training data to adjust these weights to avoid over-fitting. To address this problem, we adopt the negative sampling strategy with linear-time computation (Mikolov et al. 2013) that samples multiple negative nodes according to some noisy distribution for each target node. The sampling distribution is proportional to the $3/4$ power of node degree. Therefore, in each iteration, we only need to care about the positive sample, its neighbor nodes, and its negative samples. In this way, the comprehensive loss function of NetMCs is defined as follows:

$$\mathcal{J} = \sum_{m=1}^M \sum_{u \in \mathcal{W}_{P_m}} \mathcal{J}_u^m(\mathbf{h}_u^m; \mathbf{z}_u^m),$$

where $\mathcal{J}_u^m(\mathbf{h}_u^m; \mathbf{z}_u^m) = \log \delta((\mathbf{h}_u^m)^T \mathbf{h}_v^m) + \sum_{j=1}^b \mathbb{E}_{v_j} [\log \delta(-(\mathbf{h}_u^m)^T \mathbf{h}_{v_j}^m)]$ (7)

$$+ \lambda \sum_{m'=1, m' \neq m}^M \log \delta(-(\mathbf{h}_u^m)^T \mathbf{h}_u^{m'}) - CE(\mathbf{z}_u^m, \mathbf{z}_v^m)$$

where λ is a trade-off parameter that controls the diversity extent of the embeddings of the same node in different embedding spaces, δ is the Sigmoid function, and b is the number of negative samples. The first term forces the embedding representations of u and its context v to be similar, while the second term is to force \mathbf{h}_u^m and the embeddings of negative samples $\mathbf{h}_{v_j}^m$ in the same clustering aspect to be different. The third term seeks different embedding representations of the same nodes for different clustering aspects, and the last term forces nodes with similar embeddings to be placed into the same cluster via the classical cross entropy. By maximizing the objective loss, NetMCs can generate multiple embeddings and multiple clusterings of nodes therein.

Algorithm 1 NetMCs: Multiple Clusterings of Heterogeneous Information Networks

Input: A heterogeneous information network dataset $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, scalar parameter λ , a set of meta-paths $\mathcal{PS} = \{\mathcal{P}_m\}_{m=1}^M$, a set of numbers of clusters $\mathcal{K} = \{k_m\}_{m=1}^M$, walks per node w , walk length l , and learning rate η .

Output: Multiple node embeddings $\mathcal{H} = \{\mathbf{h}_u^m | u \in \mathcal{V}\}_{m=1}^M$, and multiple clustering results $\mathcal{Z} = \{\mathbf{z}_u^m | u \in \mathcal{V}\}_{m=1}^M$.

MetaPathBasedRandomWalk($\mathcal{G}, \mathcal{PS}, u, l$)

```

1:  $MP = \{\}_{m=1}^M$ 
2: for  $m = 1 : M$  do
3:   for  $u \in \mathcal{V}$  do
4:      $mp[1] = u$ 
5:     for  $i = 2 : l$  do
6:       Sample  $v$  according to the meta-path  $\mathcal{P}_m \in \mathcal{PS}$ 
7:        $mp[i] = v$ 
8:     end for
9:      $MP[m] = MP[m] \cup mp$ 
10:  end for
11: end for
12: return  $MP$ 

```

VariantSkipGram($\mathcal{H}, \mathcal{Z}, MP$)

```

1: Initialize  $\mathcal{H}, \mathcal{Z}$ 
2: while not converged do
3:   for  $m = 1 : M$  do
4:     for  $u \in \mathcal{V}$  do
5:       Sample neighbor nodes  $v$  and negative samples  $v'$  of  $u$  from  $MP[m]$ 
6:        $\mathbf{h}_u^m \leftarrow \mathbf{h}_u^m - \eta \partial \mathcal{J}_u^m / \partial \mathbf{h}_u^m$ ;
7:        $\mathbf{z}_u^m \leftarrow \mathbf{z}_u^m - \eta \partial \mathcal{J}_u^m / \partial \mathbf{z}_u^m$ ;
8:     end for
9:   end for
10: end while
11: Return  $\mathcal{H}, \mathcal{Z}$ .

```

Algorithm 1 gives in details how our method works. NetMCs can be divided into preprocessing phase (MetaPathBasedRandomWalk) and training phase (VariantSkipGram). First of all, some parameters, such as scalar parameter λ , a set of meta-paths $\mathcal{PS} = \{\mathcal{P}_m\}_{m=1}^M$ and walks per node w , are initialized. Next, NetMCs executes meta-path based random walk on the input HIN, guided by the set of predefined meta-paths $\mathcal{PS} = \{\mathcal{P}_m\}_{m=1}^M$, and \mathcal{P}_m represents a meta-path schema in \mathcal{PS} . NetMCs starts random walks from the node whose type accords with the head of \mathcal{P}_m . Then the flow of the walker is conditioned on the predefined meta-path \mathcal{P}_m . When this preprocessing phase is over, M sets of random walk sequences MP are generated and then are sent into the next training phase. In the training phase, NetMCs firstly initializes node embeddings \mathcal{H}^m and clustering assignment \mathcal{Z}^m . After that, NetMCs samples target node u successively as well as its neighbor nodes v and negative nodes v' from random walk sequences MP as a tuple, along with the loss defined in Eq. (7) to optimize the variant skip-gram model to generate the diverse node embeddings \mathcal{H}^m and clustering results \mathcal{Z}^m .

Table 1 Statistics of DBLP, IMDb and YAGO networks

Datasets	Number of nodes $ \mathcal{V} $	Node type $ \mathcal{T}_v $
DBLP	37,582	4
IMDb	45,519	5
YAGO	579,721	7

4 Experimental results and analysis

In this section, we evaluate the effectiveness and efficiency of our proposed NetMCs on mining multiple clusterings on real-world HIN datasets.

4.1 Experimental setup

Datasets We use three publicly available real-world HIN datasets: DBLP, IMDb, and YAGO. **DBLP** is a bibliographical network in the computer science domain. We use its subnetwork collected by Lu et al. (2019), which contains four types of nodes: author (A), paper (P), venue (V), and term (T). The edge types include authors writing papers, papers published in venues, and papers belonging to terms. **IMDb** is a HIN built by linking the movie-attribute information from IMDb and the user-reviewing-movie relationship from MovieLens-100K. There are five types of nodes in the network: user (U), movie (M), actor (A), director (D), and genre (G). The edge types include: user reviewing movie, actor featuring in movie, director directing movie, and movie being of genre. **YAGO** (Suchanek et al. 2007) is a knowledge graph derived from merging Wikipedia, GeoNames and WordNet. YAGO dataset consists of 7 types of nodes: person (P), organization (O), location (L), prize (R), work (W), position (S), event (E), and 24 edge types. The statistics of these datasets are listed in Table 1.

Baselines We compare NetMCs against four recent multiple clusterings methods and four multi-facet network representation learning methods. **MNMF** (Yang and Zhang 2017) is a multiple clusterings solution based on vector data, it defines a regularization term to quantify and minimize the redundancy between the already generated clusterings and the to-be-generated one. **Nr-kmeans** (Mautz et al. 2018) tries to explore multiple mutually orthogonal subspaces from vector data, along with the optimization of k-means objective function, to find non-redundant clusterings. **MVMC** (Yao et al. 2019a) mines common and specific information of multi-view data with self-representation learning to achieve multiple clusterings. **DMClusters** (Wei et al. 2020b) employs deep matrix factorization and redundancy control to generate multiple subspaces from layer-wise and obtains different clustering results therein. **Splitter** (Epasto and Perozzi 2019) is the only compared method designed for homogeneous networks, while the others (**metapath2vec** (Dong et al. 2017), **ASPEM** (Shi et al. 2018a), **HEER** (Shi et al. 2018b)) are applied for heterogeneous networks. These multi-facet network embedding methods were introduced in the Introduction.

For all the embedding methods, we set the embedding dimension d to 128. For the random walk based methods (metapath2vec, Splitter and NetMCs), we use the following parameter values, number of walks per node w : 20; walk length l : 30; neighborhood size c : 3; and size of negative samples b : 5. The other input parameters of the compared methods are fixed (or optimized) as the authors suggested in their papers or shared codes. For

MNMF and Nr-kmeans, we take each row vector of the adjacency matrix of a HIN as the input feature vector of the single view compared methods. For MVMC and DMClusters, we adopt different node similarity matrices as data views, following the work in Pathsim (Sun et al. 2011) using different meta-paths of HIN. For the sake of fair comparison, the selected meta-path is the same as the following experimental setup. The codes of NetMCs are available at our website.¹

Evaluation metrics Multiple clusterings approaches aim to generate diverse clusterings of good quality. To measure quality, we use the Silhouette Coefficient (SC) and the Dunn Index (DI) as internal indices to quantify the compactness and separation of clusters. To measure redundancy, we use the Normalized Mutual Information (NMI) and the Jaccard Coefficient (JC) as external indices to quantify the similarity between clusters of two clusterings. These metrics have been extensively used in the multiple clusterings literature (Yang and Zhang 2017; Yao et al. 2019a; Wang et al. 2020). The formal definitions of these metrics are given as below.

Silhouette Coefficient is the mean silhouette value over all samples. The silhouette value of each sample is a measure of how similar the sample is to the points in its own cluster, when compared to the samples in other clusters. SC is computed as follows:

$$SC(C) = \frac{1}{N} \sum_{i=1}^N \frac{f(i) - a(i)}{\max\{a(i), f(i)\}} \tag{8}$$

where N is the number of samples, $a(i)$ is the average distance of the i -th sample to the other points in the same cluster, and $f(i)$ is the minimum average distance of the i -th sample to the points in a different cluster, minimized over the clusters.

Dunn Index measures the ratio between the minimum distance of two arbitrary clusters and the maximum inter-cluster distance. DI is defined as follows:

$$DI(C) = \frac{\min_{i \neq j} \{\delta(c_i, c_j)\}}{\max_{1 \leq l \leq k} \{\Delta(c_l)\}} \tag{9}$$

where $\delta(c_i, c_j)$ is the cluster-to-cluster distance of pairwise clusters and $\Delta(c_l)$ is the cluster diameter measure.

Normalized Mutual Information measures diversity based on the ratio of joint entropy and individual entropy of clusterings C and C^* . NMI is computed as follows:

$$NMI(C, C^*) = \frac{\sum_{i=1}^k \sum_{j=1}^{k^*} n_{ij} \log \frac{n_{ij}}{n_i n_j}}{\sqrt{(\sum_{i=1}^k n_i \log \frac{n_i}{n})(\sum_{j=1}^{k^*} n_j \log \frac{n_j}{n})}} \tag{10}$$

where n_i denotes the number of samples in cluster C_i , n_j is the number of points in C_j^* , and n_{ij} denotes the number of data in the intersection of clusters C_i and C_j^* .

Jaccard Coefficient measures the overlap between two different clusterings based on ‘pair-counting’ as follows:

$$JC(C, C^*) = \frac{n_{11}}{n_{11} + n_{10} + n_{01}} \tag{11}$$

¹ <http://sdu-idea.cn/codes.php?name=NetMCs>.

where n_{11} is the number of pairwise samples that are in the same cluster in both \mathcal{C} and \mathcal{C}^* ; n_{00} measures the number of pairs that are in different clusters in both \mathcal{C} and \mathcal{C}^* ; n_{01} and n_{10} are the numbers of pairs in the same cluster in one of \mathcal{C} and \mathcal{C}^* , but not in the other.

We want to remark that *higher* values of SC and DI signify a clustering of *higher* quality. On the other hand, *smaller* values of NMI and JC imply that two clusterings have a *smaller* redundancy, or a *higher* diversity. We take the average SC and DI of multiple clusterings, and the average NMI and JC of pairwise clusterings as the evaluation results.

4.2 Discovering multiple clusterings in HIN

For the first experiment, we set the number of clusterings $M = 2$. The nodes in DBLP are connected with 20 conferences, and the nodes in IMDb are connected with 23 movie genres. In addition, we extract nodes that are related to 10 prizes on YAGO dataset. In this way, we fix the number of clusters of individual clusterings on DBLP, IMDb and YAGO as 20, 23 and 10, respectively. Such configuration of the number of clusters is motivated by two factors. First, they are widely used clustering ways for these public heterogeneous network datasets. Second, it is necessary to unify the number of clusters for the computation of evaluation metrics and for quantitative comparison. The parameter λ of NetMCs is chosen from 10^{-3} to 10^3 . Because these compared methods (except MNMF, Nr-kmeans and MVMC) can not directly give the clustering results, we employ k -means to generate individual clusterings from respective embeddings of these methods. Since how to select meta-paths is a non-trivial work in heterogeneous network analysis and there is currently no principle way for selecting meta-paths (Meng et al. 2015; Zhou et al. 2019), we choose the meta-paths widely used in past works (Sun et al. 2013; Dong et al. 2017). Particularly, for DBLP we select two candidate meta-paths (A-P-C-P-A and A-P-T-P-A) for all applicable methods. The candidate meta-paths for IMDb are U-M-G-M-U and U-M-U; for YAGO are P-L-P, P-R-P and P-O-L-O-P. If there are many valid meta paths, we can just use some of them, choose from them by reducing the redundancy, or by referring to the clustering results in a wrapper way. We report the average results and standard deviations of ten independent runs of each method on generating two clusterings on the same datasets, and report the results in Table 2. ‘N/A’ means no experimental results, since Nr-kmeans can only produce one clustering on the YAGO dataset. In the experiments, we focus on the clustering of authors by attended conferences or paper topics in DBLP, of users by genres or taste for movies in IMDb, and of persons by awarded prizes or locations in YAGO. We can make the following observations:

- (i) *Quality of multiple clusterings* NetMCs often obtains a better quality than the other methods. Both MNMF and Nr-kmeans target at multiple clusterings on vector data, they suffer the difficulty to find subspaces of good quality from sparse network data. Although different feature views of the HIN were generated for multi-view multiple clusterings algorithms, our NetMCs frequently outperforms MVMC and DMClusts, which are less capable to capture semantic information and nonlinear structure of network data. As such, traditional vector-based multiple clusterings methods have a lower quality than NetMCs. This fact suggests the necessity of developing the network-based multiple clusterings solution. The other four compared methods first seek multiple embeddings and then generate alternative clusterings in the embedding spaces. They also lose to NetMCs, which suggests that the joint optimization of embeddings and clusterings is necessary. In other words, the stage-wise approaches

Table 2 Quality and Diversity of the various compared methods on generating multiple clusterings in HIN

	MNMF	Nr-kmeans	MVMC	DMClusts	ASPEM	Splitter	Metapath2vec	HEER	NetMCs	
DBLP	SC†	0.083 ± 0.007 ^o	0.002 ± 0.001 [•]	-0.001 ± 0.000 [•]	0.013 ± 0.000 [•]	0.005 ± 0.000 [•]	0.012 ± 0.001 [•]	0.009 ± 0.000 [•]	0.032 ± 0.001 ^o	0.024 ± 0.001
	DI†	0.179 ± 0.011 [•]	0.294 ± 0.006 ^o	0.003 ± 0.000 [•]	0.121 ± 0.003 [•]	0.122 ± 0.000 [•]	0.086 ± 0.000 [•]	0.018 ± 0.001 [•]	0.024 ± 0.002 [•]	0.223 ± 0.002
	NMI↓	0.105 ± 0.005 [•]	0.433 ± 0.005 [•]	0.005 ± 0.000	0.009 ± 0.000 [•]	0.523 ± 0.002 [•]	0.420 ± 0.001 [•]	0.210 ± 0.003 [•]	0.774 ± 0.007 [•]	0.005 ± 0.000
IMDb	JC↓	0.192 ± 0.001 [•]	0.357 ± 0.015 [•]	0.063 ± 0.000 [•]	0.082 ± 0.001 [•]	0.335 ± 0.001 [•]	0.227 ± 0.004 [•]	0.089 ± 0.001 [•]	0.452 ± 0.004 [•]	0.053 ± 0.000
	SC†	-0.166 ± 0.006 [•]	0.139 ± 0.005 ^o	0.012 ± 0.001 ^o	0.001 ± 0.000 [•]	-0.009 ± 0.000 [•]	-0.002 ± 0.000 [•]	0.014 ± 0.000 ^o	0.006 ± 0.001 [•]	0.009 ± 0.000
	DI†	0.044 ± 0.002 [•]	0.258 ± 0.009 [•]	0.015 ± 0.000 [•]	0.123 ± 0.004 [•]	0.210 ± 0.002 [•]	0.368 ± 0.001 ^o	0.135 ± 0.000 [•]	0.291 ± 0.003 [•]	0.315 ± 0.010
YAGO	NMI↓	0.174 ± 0.013 [•]	0.899 ± 0.001 [•]	0.031 ± 0.000 ^o	0.074 ± 0.001 [•]	0.172 ± 0.001 [•]	0.103 ± 0.000 [•]	0.130 ± 0.000 [•]	0.412 ± 0.002 [•]	0.066 ± 0.001
	JC↓	0.051 ± 0.001 [•]	0.930 ± 0.016 [•]	0.059 ± 0.000 [•]	0.067 ± 0.000 [•]	0.098 ± 0.000 [•]	0.199 ± 0.002 [•]	0.104 ± 0.001 [•]	0.132 ± 0.000 [•]	0.048 ± 0.000
	SC†	0.347 ± 0.012 ^o	N/A	0.002 ± 0.000 [•]	0.011 ± 0.000 [•]	0.005 ± 0.000 [•]	0.015 ± 0.000 [•]	0.020 ± 0.000 [•]	0.160 ± 0.001 ^o	0.031 ± 0.002
YAGO	DI†	0.138 ± 0.002 [•]	N/A	0.021 ± 0.000 [•]	0.091 ± 0.000 [•]	0.301 ± 0.000 ^o	0.137 ± 0.001 [•]	0.059 ± 0.002 [•]	0.009 ± 0.000 [•]	0.238 ± 0.007
	NMI↓	0.706 ± 0.013 [•]	N/A	0.023 ± 0.000 [•]	0.047 ± 0.000 [•]	0.237 ± 0.001 [•]	0.188 ± 0.002 [•]	0.141 ± 0.001 [•]	0.737 ± 0.003 [•]	0.020 ± 0.000
	JC↓	0.482 ± 0.020 [•]	N/A	0.042 ± 0.000 ^o	0.099 ± 0.000 [•]	0.164 ± 0.000 [•]	0.140 ± 0.000 [•]	0.101 ± 0.001 [•]	0.644 ± 0.003 [•]	0.055 ± 0.001

†(↓) indicates the preferred direction (the larger(smaller) the better) for the corresponding measure. •/° indicates whether our NetMCs is superior/inferior to the other method, with statistical significance checked by pairwise *t*-test at 95% level

Table 3 Comparison results of NetMCs and its variants

		NetMCs-nRC	NetMCs-nCE	NetMCs
DBLP	SC↑	-0.008 ± 0.001 •	0.005 ± 0.000 •	0.024 ± 0.002
	DI↑	0.333 ± 0.009 ◦	0.252 ± 0.003 ◦	0.223 ± 0.007
	NMI↓	0.013 ± 0.000 •	0.005 ± 0.000	0.005 ± 0.000
	JC↓	0.079 ± 0.002 •	0.127 ± 0.005 •	0.053 ± 0.001
IMDb	SC↑	0.030 ± 0.001 ◦	-0.041 ± 0.001 •	0.009 ± 0.000
	DI↑	0.363 ± 0.011 ◦	0.283 ± 0.004 •	0.315 ± 0.007
	NMI↓	0.113 ± 0.004 •	0.143 ± 0.002 •	0.066 ± 0.001
	JC↓	0.069 ± 0.000 •	0.049 ± 0.000 •	0.048 ± 0.000
YAGO	SC↑	0.038 ± 0.001 ◦	-0.016 ± 0.000 •	0.031 ± 0.001
	DI↑	0.267 ± 0.011 ◦	0.200 ± 0.006 •	0.238 ± 0.008
	NMI↓	0.057 ± 0.003 •	0.007 ± 0.000 ◦	0.020 ± 0.001
	JC↓	0.237 ± 0.010 •	0.152 ± 0.005 •	0.055 ± 0.003

NetMCs-nRC disregards the redundancy control term, and NetMCs-nCE disregards the cross entropy term. •/◦ indicates whether our NetMCs is superior/inferior to the variant, with statistical significance checked by pairwise *t*-test at 95% level

suffer from the optimization inconsistency between the sequential embedding and clustering steps. We observe that NetMCs sporadically has lower SC and DI values than some of the compared methods. This is due to the widely-recognized dilemma of obtaining alternative clusterings with both high diversity and high quality. Overall, the results prove the effectiveness of our proposed unified framework to generate multiple clusterings of quality.

- (ii) *Diversity of multiple clusterings* Both NMI and JC are canonically used to measure the diversity of multiple clusterings. The two clusterings generated by NetMCs often have a lower redundancy (higher diversity) than those generated by the compared methods. This is because NetMCs introduces a regularization term to specifically control the redundancy. Although MNMF considers the redundancy, it cannot control the clusterings' diversity well because of the sparseness and high dimensionality of the node feature vectors. Nr-kmeans does not explicitly consider the redundancy. MVMC and DMClusts also emphasize on redundancy control, so their diversity is relatively lower compared with other methods. Splitter cannot distinguish different types of nodes to eliminate semantic redundancy. metapath2vec cannot control the semantic redundancy between different meta-paths, and it can only utilize a single meta-path each time. ASPEM and HEER consider the inconsistency between different types of relations, but they cannot capture the diverse semantic information as meta-path based methods. For these reasons, the multiple clusterings generated by the compared methods have a lower diversity.

Besides the pairwise *t*-test, we further applied the nonparametric Wilcoxon signed-rank test to check the difference between NetMCs and other compared methods, all the *p*-values are smaller than 0.01. In conclusion, NetMCs outperforms the other methods across the benchmark HIN datasets on generating multiple clusterings in terms of quality and diversity.

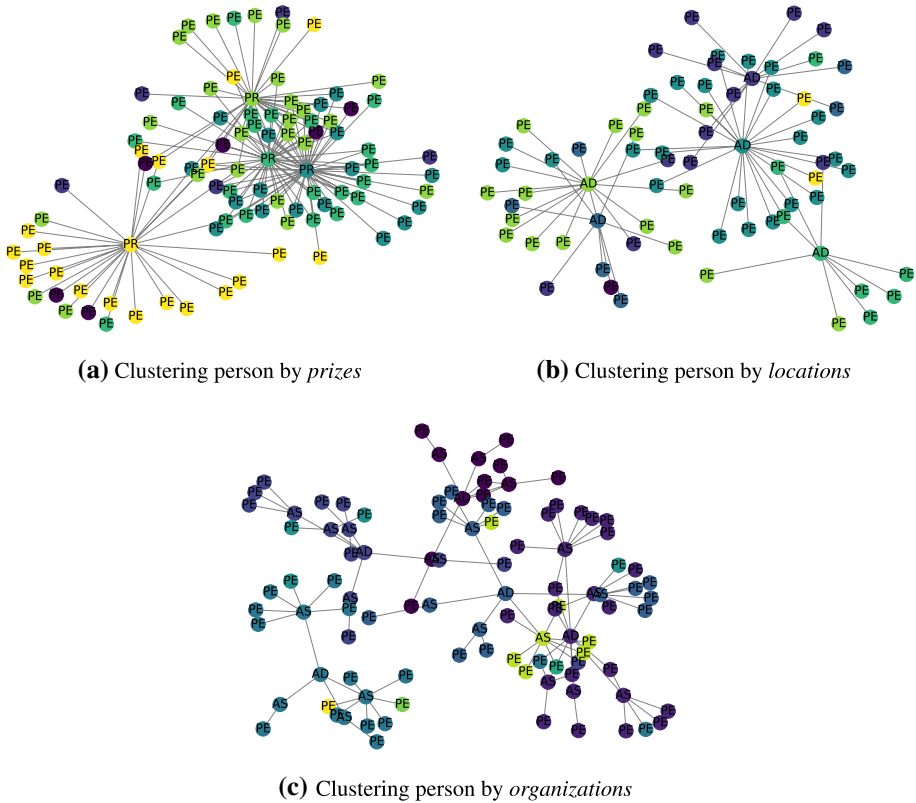


Fig. 2 Three alternative clusterings generated by NetMCs on the YAGO dataset. The various clusters in each clustering are distinguished by different node colors. The labels PE, PR, AD, and AS on the nodes represent different node types (person, prize, location, and organization)

4.3 Ablation study

We perform ablation study to investigate the contribution factors of NetMCs and report the results in Table 3. For this investigation, we introduce two variants of NetMCs. NetMCs-nRC means NetMCs without the redundancy control term (the third term in Eq. (7)). NetMCs-nCE means NetMCs without the cross entropy term (the last term in Eq. (7)). Because NetMCs-nCE can not directly give the clustering results, we perform k-means on the node embeddings generated by NetMCs-nCE. We can see that NetMCs-nRC often performs better than NetMCs in terms of the quality (SC and DI), but loses to NetMCs in terms of the diversity (NMI and JC) between clusterings. This result proves the effectiveness of the redundancy control term on improving the diversity, but it comes at the expense of compromising the quality, which coincides with known dilemma of quality and diversity of multiple clusterings. On the other hand, NetMCs always has a higher quality than NetMCs-nCE, while maintains a comparable diversity. This observation confirms that optimizing the objective of multi-facet embeddings and of multiple clusterings in a unified framework can alleviate the optimization inconsistency problem and produce alternative clusterings of better quality.

We also visualize three alternative clustering results discovered by NetMCs on the YAGO dataset in Fig. 2. The considered meta-paths are P-L-P, P-R-P and P-O-L-O-P. Different clusters in each clustering are distinguished by different node colors. From Fig. 2, we can see that the persons in three subgraphs are clustered by the *prize* they win, by their *locations* and by their affiliated *organizations*. All these three clusterings for the same set of persons are with different semantics but meaningful, which signifies the capability of NetMCs on generating more than two alternative clusterings. In addition, NetMCs can not only cluster a single type of nodes of a HIN, but also other types of nodes included in the meta-path, such as prizes in Fig. 2a, locations in Fig. 2b and organizations in Fig. 2c. As a result, NetMCs can generate multiple clusterings for more than one node type in HIN simultaneously. Finally, the semantics of different meta-path schemes may overlap to some extents, due to the rich semantics of HIN. For example, in the second and the third clusterings, the persons in the same locations are more likely to belong to the same organizations. To sum up, NetMCs can generate meaningful alternative clusterings of HIN from different perspectives.

4.4 Parameter, convergence, and complexity analysis

Parameter analysis To study the impact of redundancy term, we set $M = 2$ and vary λ from 10^{-3} to 10^3 with an exceptional condition $\lambda = 0$, and then plot the variation of quality (SC, the larger the better) and diversity (NMI, the smaller the better) of NetMCs on IMDb dataset in Fig. 3a. We see that: (i) the quality (SC) fluctuates within a certain range at first and then decreases rapidly as λ further increases; (ii) diversity (1-NMI) gradually increases and then keeps relatively stable. Overall, SC and NMI tend to decline as λ increases, and they are always below the starting point ($\lambda = 0$, no diversity control). This pattern is explainable, the larger λ is, the less similarity of probability distributions of the same node in different embedding spaces is. In addition, the enhancement of diversity between multiple clusterings are often accompanied by a decrease in quality. In summary, λ indeed helps to boost the diversity between clusterings. The best λ should give the highest value both in quality and diversity. Unfortunately, the best quality and diversity often can not be attained at the same time. As a result, a best λ is hard to choose. Users can adjust λ according to their prefer on diversity (large λ) or quality (small λ).

We study the impact of embedding dimension d of NetMCs. From Fig. 3b, we can observe that the quality (SC) fluctuates slightly, while the diversity (1-NMI) gradually decreases as the dimension increase. Overall, the embedding dimensions d impacts the quality of multiple clusters to some extent. The best balance between quality and diversity can be made when $d \approx 128 \sim 256$.

We vary M (number of alternative clusterings) from 2 to 6 on IMDb dataset to explore the variation of average quality (SC) and diversity (NMI) of multiple clusterings generated by NetMCs. In Fig. 3c, with the increase of M , the average quality (SC) decreases slowly while the diversity (NMI) fluctuates within a small range. Overall, NetMCs can generate $M \geq 2$ alternative clusterings of quality and diversity, it obtains a better performance than other compared methods across different input values of M in most cases. In fact, the number of alternative clusterings of NetMCs can be adjusted according to user preference or determined in advance by prior knowledge.

We also investigate the effect of walk length l on the performance of NetMCs. In Fig. 3d, we vary walk length l from 10 to 60 and plot the corresponding SC and NMI under each fixed l . We find that either a too short or too long walk length has a negative

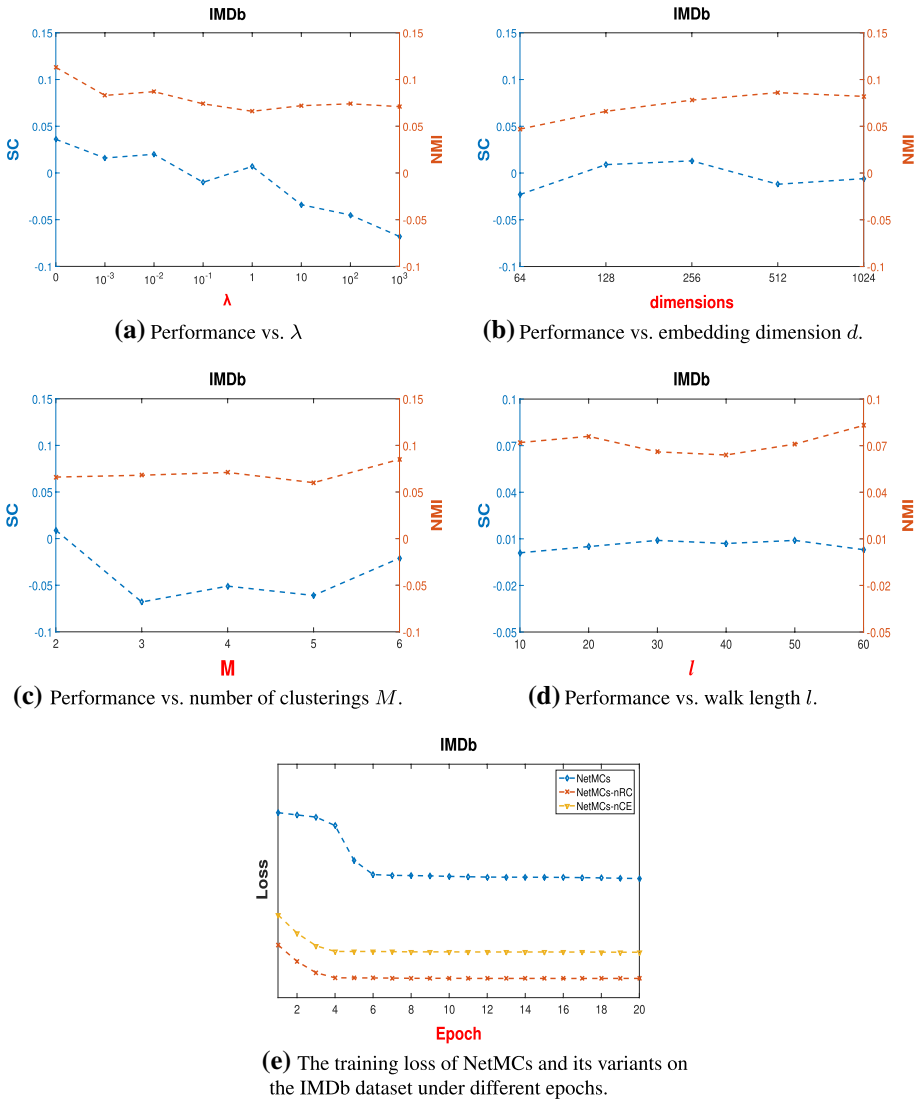


Fig. 3 Parameter and convergence analysis of NetMCs

impact on quality and diversity of multiple clusterings. That is because a too short walk length will affect the capture of semantic information, while a too long walk length will introduce noisy information.

Convergence analysis Figure 3e shows the loss value against the number of epochs for NetMCs and its variants. We see that at the beginning the loss value drops rapidly, and typically converges in around 5 epochs. This not only proves the efficiency of NetMCs, but also shows that the redundancy control term and cross entropy term do not significantly increase the complexity of the optimization procedure.

Table 4 Runtimes of compared methods (in min) on three network datasets

	MNMF	Nr-kmeans	MVMC	DMClusts	ASPEM	Splitter	Metapath2vec	HEER	NetMCs
DBLP	2769	6495	3811	293	39	7226	3956	52	4426
IMDb	33	417	6	2	41	6641	470	46	754
YAGO	981	N/A	6153	438	1463	15,793	4421	1700	5195
Total	3783	N/A	9970	733	1543	29,660	8847	1798	10,375

Complexity analysis The time complexity of NetMCs includes two parts. NetMCs takes $\mathcal{O}(Mlw|\mathcal{V}|)$ steps to obtain M groups of meta-path random walk sequences, and $\mathcal{O}(eMlw(c+b)d|\mathcal{V}|)$ steps to update the nodes' embeddings and clustering assignment vectors, where e , l , w , c , b , and d are the number of epoches for optimization, the walk length, the number of walks per node, the neighborhood size, the number of negative samples, and the embedding dimensions, respectively. Note that $eMlwcd \ll |\mathcal{V}|$, and the complexity of NetMCs is linear with respect to $|\mathcal{V}|$, while existing multiple clustering methods (Yang and Zhang 2017; Mautz et al. 2018; Yao et al. 2019a; Wei et al. 2020b) typically have a quadratic or cubic complexity in $|\mathcal{V}|$.

Table 4 gives the runtimes of the compared methods and of NetMCs. The experiments are conducted on a server.² All methods are implemented in Python supported by PyTorch machine learning framework, except for MNMF and MVMC that run on Matlab2014a. We observe that the three fastest methods are DMClusts, ASPEM and HEER. DMClusts has a linear time complexity. ASPEM and HEER both build on LINE (Tang et al. 2015), which considers the second order approximation of networks and ignores the semantic information embodied by meta-path; as such, they run faster than the other approaches. MNMF is also relatively fast, due to the decomposition of the sparse adjacency matrix into low dimensional ones. On the contrary, Nr-kmeans bears the curse of dimensionality and it cannot run on the large YAGO dataset. MVMC runs slowly due to the high time complexity of self-representation learning. NetMCs builds on metapath2vec, and the introduced terms (redundancy control and clustering assignment) of NetMCs do not increase the order of computational complexity, so NetMCs and metapath2vec have similar runtimes. Splitter is also a random walk based solution, but it needs to cluster each node based on its context to build the ego networks. Therefore, it has the highest runtime. Compared with other linear methods, NetMCs bears larger coefficient over $|\mathcal{V}|$, the walk length l and the number of walks per node w . In addition, NetMCs has to generate both embeddings and clustering results simultaneously, while other compared methods do not. Thus, its running times are high with respect to most of the competitors. In conclusion, the runtime of NetMCs is in the medium range, and it runs slightly slower than metapath2vec, since it has to control the redundancy and execute clustering assignments.

² Configuration: OS Ubuntu 16.04, Intel Xeon8163, 1TB RAM with NVIDIA TITAN V.

5 Conclusion and future work

In this paper, we introduced the NetMCs model to explore alternative clusterings from the ubiquitous heterogeneous information networks, which is an interesting, practical but overlooked clustering topic that conjoins multiple embeddings and multiple clusterings of the same network. NetMCs can seek multiple embeddings and multiple clustering results simultaneously by a variation of the skip-gram model under different semantic meta-paths. It further introduces a redundancy term to improve the diversity between alternative clusterings. Experimental results confirm the advantage of NetMCs to state-of-the-art competitive multiple embeddings/clusterings solutions. We will investigate a principle to automatically choose the meta-paths (Sun et al. 2013; Zhou et al. 2019; Meng et al. 2015) and number of alternative clusterings, and multiple clusterings on dynamic network (Loglisci et al. 2012).

Acknowledgements This work is partially supported by NSFC (Nos. 62031003, 62072380 and 61872300).

References

- Bae, E., & Bailey, J. (2006). Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *Proceedings of IEEE international conference on data mining* (pp. 53–62).
- Bailey, J. (2013). Alternative clustering analysis: A review. In *Data clustering: Algorithms and applications* (pp. 535–550).
- Caruana, R., Elhawary, M., Nguyen, N., & Smith, C. (2006). Meta clustering. In *Proceedings of IEEE international conference on data mining* (pp. 107–118).
- Chen, J., Dai, W., Sun, Y., & Dy, J. G. (2015). Clustering and ranking in heterogeneous information networks via gamma-Poisson model. In *Proceedings of SIAM international conference on data mining* (pp. 424–432).
- Cui, P., Wang, X., Pei, J., & Zhu, W. (2019). A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5), 833–852.
- Cui, Y., Fern, X. Z., & Dy, J. G. (2007). Non-redundant multi-view clustering via orthogonalization. In *Proceedings of IEEE international conference on data mining* (pp. 133–142).
- Ding, C. H., Li, T., & Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1), 45–55.
- Dong, Y., Chawla, N. V., & Swami, A. (2017). metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 135–144).
- Epasto, A., & Perozzi, B. (2019). Is a single embedding enough? Learning node representations that capture multiple social contexts. In *Proceedings of international conference on world wide web* (pp. 394–404).
- Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 855–864).
- Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8), 651–666.
- Jain, P., Meka, R., & Dhillon, I. S. (2008). Simultaneous unsupervised learning of disparate clusterings. *Statistical Analysis and Data Mining*, 1(3), 195–210.
- Li, X., Kao, B., Ren, Z., & Yin, D. (2019). Spectral clustering in heterogeneous information networks. In *AAAI conference on artificial intelligence* (pp. 4221–4228).
- Lin, W., Yu, P. S., Zhao, Y., & Deng, B. (2016). Multi-type clustering in heterogeneous information networks. *Knowledge and Information Systems*, 48(1), 143–178.
- Loglisci, C., Ceci, M., & Malerba, D. (2012). Discovering evolution chains in dynamic networks. In *International workshop on new frontiers in mining complex patterns* (pp. 185–199).
- Lu, Y., Shi, C., Liu, Z., & Mei Hu, L. (2019). Relation structure-aware heterogeneous information network embedding. In *AAAI conference on artificial intelligence* (pp. 4456–4463).

- Luo, S., Zhang, C., Zhang, W., & Cao, X. (2018). Consistent and specific multi-view subspace clustering. In *AAAI conference on artificial intelligence* (pp. 3730–3737).
- Mall, R., Langone, R., & Suykens, J. A. K. (2013). Kernel spectral clustering for big data networks. *Entropy*, 15(5), 1567–1586.
- Mautz, D., Ye, W., Plant, C., & Böhm, C. (2018). Discovering non-redundant k-means clusterings in optimal subspaces. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1973–1982).
- Meng, C., Cheng, R., Maniu, S., Senellart, P., & Zhang, W. (2015). Discovering meta-paths in large heterogeneous information networks. In *Proceedings of the 24th international conference on world wide web* (pp. 754–764).
- Miklautz, L., Mautz, D., Altinigneli, M. C., Bohm, C., & Plant, C. (2020). Deep embedded non-redundant clustering. In *AAAI conference on artificial intelligence* (pp. 5174–5181).
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).
- Niu, D., Dy, J. G., & Jordan, M. I. (2013). Iterative discovery of multiple alternative clustering views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 1340–1353.
- Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 701–710).
- Pio, G., Serafino, F., Malerba, D., & Ceci, M. (2018). Multi-type clustering and classification from heterogeneous networks. *Information Science*, 425, 107–126.
- Shi, Y., Gui, H., Zhu, Q., Kaplan, L. M., & Han, J. (2018a). Aspem: Embedding learning by aspects in heterogeneous information networks. In *Proceedings of SIAM international conference on data mining* (pp. 144–152).
- Shi, Y., Zhu, Q., Guo, F., Zhang, C., & Han, J. (2018b). Easing embedding learning by comprehensive transcription of heterogeneous information networks. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2190–2199).
- Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: A core of semantic knowledge. In *Proceedings of international conference on world wide web* (pp. 697–706).
- Sun, Y., & Han, J. (2013). Mining heterogeneous information networks: A structural analysis approach. *SIGKDD Explorations*, 14(2), 20–28.
- Sun, Y., Han, J., Yan, X., Yu, P. S., & Wu, T. (2011). Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11), 992–1003.
- Sun, Y., Norrick, B., Han, J., Yan, X., PHILIP, S. Y. U., & Xiao, Y. U. (2013). Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data*, 7(3), 1–23.
- Sun, Y., Yu, Y., & Han, J. (2009). Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 797–806).
- Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). LINE: Large-scale information network embedding. In *Proceedings of international conference on world wide web* (pp. 1067–1077).
- Trigeorgis, G., Bousmalis, K., Zafeiriou, S., & Schuller, B. W. (2016). A deep matrix factorization method for learning attribute representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3), 417–429.
- Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., & Yu, P. S. (2019a). Heterogeneous graph attention network. In *Proceedings of international conference on world wide web* (pp. 2022–2032).
- Wang, J., Wang, X., Yu, G., Domeniconi, C., Yu, Z., & Zhang, Z. (2020). Discovering multiple co-clusterings with matrix factorization. *IEEE Transactions on Cybernetics*, 99(1), 1–14.
- Wang, X., Wang, J., Yu, G., Domeniconi, C., Xiao, G., & Guo, M. (2019b). Multiple independent subspace clusterings. In *AAAI conference on artificial intelligence* (pp. 5353–5360).
- Wang, X., Yu, G., Domeniconi, C., Wang, J., Yu, Z., & Zhang, Z. (2018). Multiple co-clusterings. In *Proceedings of IEEE international conference on data mining* (pp. 1308–1313).
- Wei, S., Wang, J., Yu, G., Domeniconi, C., & Zhang, X. (2020a). Deep incomplete multi-view multiple clusterings. In *Proceedings of IEEE international conference on data mining* (pp. 651–660).
- Wei, S., Wang, J., Yu, G., Domeniconi, C., & Zhang, X. (2020b). Multi-view multiple clusterings using deep matrix factorization. In *AAAI conference on artificial intelligence* (pp. 6348–6355).
- Yang, C., Xiao, Y., Zhang, Y., Sun, Y., & Han, J. (2020). Heterogeneous network representation learning: Survey, benchmark, evaluation, and beyond. arXiv preprint [arXiv:2004.00216](https://arxiv.org/abs/2004.00216).

- Yang, L., Guo, Y., & Cao, X. (2018). Multi-facet network embedding: Beyond the general solution of detection and representation. In *AAAI conference on artificial intelligence* (pp. 499–506).
- Yang, S., & Zhang, L. (2017). Non-redundant multiple clustering by nonnegative matrix factorization. *Machine Learning*, 106(5), 695–712.
- Yao, S., Yu, G., Wang, J., Domeniconi, C., & Zhang, X. (2019a). Multi-view multiple clustering. In *Proceedings of international joint conference on artificial intelligence* (pp. 4121–4127).
- Yao, S., Yu, G., Wang, X., Wang, J., Domeniconi, C., & Guo, M. (2019b). Discovering multiple co-clusterings in subspaces. In *Proceedings of SIAM international conference on data mining* (pp. 423–431).
- Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2020). Network representation learning: A survey. *IEEE Transactions on Big Data*, 6(1), 3–28.
- Zhao, H., Yao, Q., Li, J., Song, Y., & Lee, D. L. (2017). Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 635–644).
- Zhou, Y., Huang, J., Sun, H., Sun, Y., Qiao, S., & Wambura, S. (2019). Recurrent meta-structure for robust similarity measure in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data*, 13(6), 1–33.
- Zhou, Z. H. (2012). *Ensemble methods: Foundations and algorithms*. Boca Raton: CRC Press.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Shaowei Wei¹ · Guoxian Yu^{1,2}  · Jun Wang³ · Carlotta Domeniconi⁴ · Xiangliang Zhang⁵

Shaowei Wei
swwei2019@email.swu.edu.cn

Jun Wang
kingjun@sdu.edu.cn

Carlotta Domeniconi
carlotta@cs.gmu.edu

Xiangliang Zhang
xiangliang.zhang@kaust.edu.sa

¹ College of Computer and Information Sciences, Southwest University, Chongqing, China

² School of Software, Shandong University, Jinan, China

³ Joint SDU-NTU Centre for Artificial Intelligence Research, Shandong University, Jinan, China

⁴ Department of Computer Science, George Mason University, Fairfax, VA, USA

⁵ Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia