Knowl Inf Syst (2016) 48:143–178
DOI 10.1007/s10115-015-0869-9

REGULAR PAPER

Multi-type clustering in heterogeneous information networks

Wangqun Lin[1] · Philip S. Yu[2,3] · Yuchen Zhao[2] · Bo Deng[1]

Received: 26 August 2014 / Revised: 24 June 2015 / Accepted: 2 August 2015 /
Published online: 19 October 2015

Abstract Heterogeneous information networks have drawn much attention in recent years due to their significant applications, such as text mining, e-commerce, social networks, and bioinformatics. Clustering different types of objects simultaneously based upon not only their relations of the same type, but also the relations between different types of objects can improve the clustering quality mutually. In this paper, we propose a general model, in which both the homogeneous and heterogeneous relations are considered simultaneously, to describe the structure of the heterogeneous information networks and devise a novel parametric free multi-type overlapped clustering approach. In this model, different types of relations between different types of objects are represented by a group of matrices. In this way, we transfer the multi-type clustering problem into the information compression problem. Subsequently, greedy search approaches, which aim at describing the group of relational matrices with least bits, are proposed. Moreover, by discovering the discriminative clusters among different types of objects, we devise effective parameter-free strategies to discover either overlapping or non-overlapping structure among different types of clusters. Extensive experiments on real-world and synthetic data sets demonstrate our methods are effective and efficient.

Keywords Heterogeneous information network · Multi-type clustering · Overlapping · Cluster

✉ Wangqun Lin
linwangqun2005@gmail.com

Philip S. Yu
phsyu@uic.edu

Yuchen Zhao
yuzhao@cs.uic.edu

Bo Deng
bodeng@vip.tom.com

1 Beijing Institute of System Engineering, Beijing, China

2 University of Illinois at Chicago, Chicago, IL, USA

3 Institute of Data Science, Tsinghua University, Beijing, China

1 Introduction

Clustering is one of the fundamental data mining tasks and has been studied for many years. Traditional network clustering methods [1,35,41,42,49] mainly focus on the relations among the same type of objects in homogeneous networks. However, in many real-world applications, such as text mining, e-commerce, social networks, and bioinformatics, there are many different types of objects interacting with each other and forming a heterogeneous information network. Usually, one type of objects interacts with another type of objects in a heterogeneous information network. And different objects have relations with each other. Moreover, the relations exist not only among objects within the same type, but also among objects of different types. In this case, information from different views reflects relationship of different types of objects and helps understand the structure of the network. Some cases of such application scenarios are as follows.

In a scientific digital library, such as DBLP,[1] the relations among different authors, papers, conferences, and other related textual objects form a heterogeneous information network. In this heterogeneous information network, different authors co-author with each other on some papers. Each paper cites some other related papers and publishes on a specific conference. Moreover, the topic of each paper is described by the related textual objects such as title, keywords, and abstract. In this heterogeneous network, each type of objects can provide additional information on the other types of objects. For instance, if we want to cluster the authors into different communities, which have similar research interests, it is not enough by only considering the co-author relations among authors. It is not unusual for an author mainly focused on artificial intelligence to co-author a paper with another author who is mainly interested in data mining. In such a case, on the one hand, if we only consider the co-author relations and drop other relation information such as relations between papers and conferences in the heterogeneous information network, we will lose the important research field information. On the other hand, if we cluster the authors by considering not only the co-author relations, but also the relations among different types of objects such as authors, papers, and conferences, the clustering quality can be improved greatly since different types of objects provide information from different views. More concretely, the textual information of each paper indicates the topic of this paper, and the conference, where the paper is published, provides information of the research field of this paper. Making use of such information improves the quality of author communities.

Clustering different types of objects in heterogeneous information networks is very important, but the task is very challenging. Firstly, clustering objects in heterogeneous information networks is different from the traditional clustering in homogeneous networks and cannot be accomplished by the traditional clustering methods. That is because different types of objects may have different types of relations. In other words, we need to consider not only the relations among objects within same type, but also the relations among objects of different types. Moreover, an appropriate framework to unify different types of objects and relations is not only critical, but also challenging. Secondly, most of the existing clustering methods [6,8,30,32] for heterogeneous information networks assume that the number of clusters in different types of objects is given. Nonetheless, these parameters are often difficult to obtain in reality. Designing an approach, which does not require user-specified parameters, is quite challenging yet much desired. Thirdly, overlapping cluster structures exist in not only homogenous networks, but also heterogeneous information networks. Discovering the overlapping structure in heterogeneous information networks can further improve the clustering quality, but

[1] http://www.informatik.uni-trier.de/ley/db/.

has been neglected by most existing multi-type clustering methods [6,8,17,23,25,30,32] in heterogeneous information networks. How to make use of different information from different types of objects and design an appropriate overlapping strategy is significant and challenging.

We notice that co-clustering a bipartite graph can get higher clustering quality than clustering two types of the objects of the bipartite separately in most situations [7,10,48]. That is because the clustering results of one type of objects can be enhanced by making use of the information of the other type of objects during the co-clustering process, and vice versa. If we can co-cluster different relation matrices of the heterogeneous network simultaneously considering not only the homogeneous links but also the heterogeneous links, the information of different types of objects can be used to enhance the quality of multi-type clustering.

In this paper, we study the problem of clustering in heterogeneous information networks. We first propose a general model to unify the heterogeneous information network, which contains both homogeneous and heterogeneous relations. Then, we encode the heterogeneous network by the MDL approach [4]. Through designing an appropriate objective function, we transfer the clustering problem into the problem of minimizing the total information used to describe the heterogeneous network. Subsequently, two greedy search methods, which are used to optimize the objective function, are designed. In order to discover the discriminative clusters in different types of objects, we devise a density guided principle, based on which a novel multi-type clustering method which can support either overlapping or non-overlapping clustering is proposed. To the best of our knowledge, we explore the first work on overlapping multi-type clustering in heterogeneous networks.

The rest of the paper is organized as follows. We introduce the related works in Sect. 2. In Sect. 3, we give our problem formulation. In Sect. 4, three different clustering methods for heterogeneous information networks are proposed. We analyze the time complexity of our proposed methods in Sect. 5. In Sect. 6, we present the experimental results. Next, a case study is presented in Sect. 7. We compare the running time of different methods in Sect. 8. Finally, we conclude in Sect. 9.

2 Related work

Traditional network clustering methods mainly focus on homogeneous networks [12,18,19, 26,34,36,39]. Many clustering approaches were proposed based on various criteria including modularity [35], normalized cut [42], structural density [49], and partition density [1]. Modularity defines the objective function as the substraction of the sum of inner link density and the sum of outer link density of the clusters. Since optimizing the modularity is NP-hard [35], many heuristic approaches were proposed. Such approaches include greedy agglomeration [46], spectral clustering [41], simulated annealing [20], sampling techniques [40], etc. Besides, there are some clustering methods based on the attributes of the nodes in homogenous network. One of the most famous algorithms is k-means. Tsai et al. [45] proposed a feature weight self-adjustment mechanism for k-means clustering. Tian et al. [44] proposed OLAP-style aggregation approaches to summarize large graphs by grouping nodes based on user-selected attributes and relationships.

If we view the heterogeneous information networks from different aspects, they are constructed by different types of bipartite graphs. Since co-clustering methods are often used for mining the clustering structures for bipartite graphs, we firstly study the related co-clustering algorithms [29]. Co-clustering makes use of the adjacency matrix of a bipartite graph and

clusters two types of objects simultaneously by exploiting the clear duality between rows and columns of the adjacency matrix [13,33]. Dhillon [13] proposed a spectral co-clustering algorithm. In this algorithm, the second left and right singular vectors of an appropriately scaled matrix are computed to yield ideal co-clusters. Dhillon and Guan [14] viewed the adjacency matrix of the bipartite graph as an empirical joint probability distribution of two discrete random variables. Given the number of clusters in each type of objects, the co-clusters can be easily detected by maximizing the mutual information between the clustered random variables. Banerjee et al. [3] proposed a general co-clustering model, in which the approximation error was measured by a large class of loss functions called Bregman divergences. Then, a new minimum Bregman information (MBI) principle was introduced to generalize the maximum entropy and standard least square principles. This general framework is an extension of some existing co-clustering methods, such as [10,15], as the special cases of this model. Chakrabarti et al. [7] viewed the process of co-clustering as the problem of how to condense the matrix with the least bits. By minimizing the total information used to describe the matrix, the co-clusters can be detected effectively. Later, Papadimitriou et al. [38] extended this method to the hierarchical situation.

Co-clustering is also very popular for analyzing the gene expression data. Cho et al. [10] used the mean squared residue score as the criterion of the result of co-clustering. Then, two different functions for measuring the residue were designed. Cheng and Church [9] proposed the sequential biclustering model. Based on this model, an algorithm, which finds out the co-clusters iteratively, was devised. Later, Lazzeroni and Owen [28] proposed a plaid model for directly finding the overlapping co-clusters, but cannot identify multiple co-clusters simultaneously. Recently, Wang et al. [48] proposed a method similar to k-means by making use of the correlations between users and tags in social media. However, this method is only tailored for social media domain and is ineffective for the general case of overlapping structures.

Some clustering models focused on heterogeneous networks were also proposed [2,43]. Wang et al. [47] presented ReCom to improve the quality of clusters through the iterative reinforcement process. Gao et al. [16] designed an algorithm named consistent bipartite graph co-partitioning (CBGC) based on consistency theory. Because CBGC takes semi-definite programming to solve the clustering problem, it is very time-consuming and does not fit for large-scale data sets. Later, Gao et al. [17] extended the pairwise information theoretic model, which was proposed by Dhillon and Guan [14], to heterogeneous networks, and proposed a star model to describe the structure of the heterogeneous network. However, in this star model, the homogenous relations among the objects of the same type were not considered. Long et al. [32] formulated the multi-type clustering as collective factorization on the relational matrices and proposed spectral relational clustering (SRC). Since SRC requires solving the eigenvector problem, the space and time consumptions are very high. Another follow-up model devised by Long et al. [31] is relational summary network (RSN) model for clustering multi-type objects in heterogeneous networks by making use of Bregman divergences. Based on pairwise interactions between variables, Bekkerman and Mccallum [5] aimed at maximizing the sum of the mutual information between clustered random variables and introduced a multi-type distributional clustering approach named MDC. Another work of Bekkerman and Jeon [6] is combinatorial Markov random field (CMRF) algorithm. In this approach, each type of objects is viewed as a single combinatorial random variable of Markov random field. However, the theoretical proof of the effectiveness and correctness of CMRF is not presented. Ienco et al. [25] devised a co-clustering method for heterogeneous networks by optimizing Goodman–Kruskal's τ which was used to describe the quality of co-clustering results. Chen et al. [8] proposed a semi-supervised nonnegative matrix factorization framework (NMF) for

Table 1 Comparison for related works

	Homogeneous	Bipartite	Heterogeneous
Non-overlapping with parameters	[34,41,44,45]	[3,7,9,10,13–15,33]	[2,5,8,17,31,32,43,47]
Non-overlapping parameter-free	[12,20,35,39,40,46]	[7,38]	[6,23,25]
Overlapping with parameters	[36]	[28,48]	None
Overlapping parameter-free	[18,19,49]	[29]	None

heterogeneous networks. This work extended nonnegative matrix factorization to multi-view data and was shown to be effective. However, in order to learn a Mahalanobis matrix, the "must" link and "cannot" link information have to be provided in advance.

Though this paper is partly inspired by Chakrabarti et al. [7], it is still very different from it. Firstly, the model used in Ref. [7] can only fit for bipartite graphs. As we known, heterogeneous information networks are more common exists in our real world. But, the heterogeneous information networks, which include more different types of objects and more complex network structures, are much harder to impress than traditional bipartite graphs. This paper proposes a more general model which can fit for almost all types of heterogeneous information networks. Secondly, the algorithms used on Ref. [7] can only fit for mining clusters in bipartite graph. But the clustering structure in heterogeneous network is much more difficult to discover. This paper further improves the original two greedy co-clustering algorithms to heterogeneous networks and uses them for mining the cluster structures in heterogeneous information networks. Moreover, it not only provides the theoretical analysis but also the experimental results to improve the effectiveness and efficiency of our algorithms. Thirdly, to the best of our knowledge, this paper explores the first work on overlapping algorithm for heterogeneous information networks.

We notice that most of the existing multi-type clustering methods [6,8,17,32] need to provide the number of clusters for each type of objects and cannot cluster the networks in which both the homogeneous and heterogeneous relations exist. Besides, none of these multi-type clustering methods [8,23,25,32] can detect the overlapping structures in heterogeneous networks. We present the comparison of our related works in Table 1.

3 Problem formulation

3.1 Problem statement

A heterogeneous information network is denoted as an undirected and unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, where $\mathcal{V} = \{V^i\}_{i=1}^k$ represents a set of different types of objects, $\mathcal{R} = \{R^{s,t}\}_{s,t=1}^k$ represents a set of link relation matrices between different types of objects. Each $|V^s| \times |V^t|$, binary matrix $R^{s,t} \in \mathcal{R}$ $(1 \leq s \leq t \leq k)$ describes the link relations between the objects of type T^s and type T^t. Specifically, if s equals to t, $R^{s,t}$ stands for the homogenous relations among objects of the same type. Otherwise, $R^{s,t}$ denotes the heterogeneous relations between objects of type T^s and T^t. Moreover, if object $v_p^s \in V^s$ has a link with object $v_q^t \in V^t$, the element at the pth row and qth column in relation matrix $R^{s,t}(s \leq t)$ equals to 1. Otherwise, it equals to 0.

We give an common example of heterogeneous information network in Fig. 1. In this heterogeneous information network, we have three different types of object sets $V^1 =$

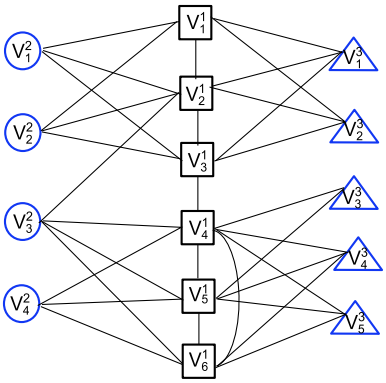


Fig. 1 A heterogeneous network with three different types of objects

$\{v_1^1, v_2^1, v_3^1, v_4^1, v_5^1, v_6^1\}$, $V^2 = \{v_1^2, v_2^2, v_3^2, v_4^2\}$ and $V^3 = \{v_1^3, v_2^3, v_3^3, v_4^3, v_5^3\}$. The homogenous relation existing in V^1 is described by $R^{1,1}$, while the heterogeneous relations between V^1, V^2, and V^3 are described by matrix $R^{1,2}$ and $R^{1,3}$, respectively. The details of relation matrices $R^{1,1}$, $R^{1,2}$, and $R^{1,3}$ are as follows:

$$R^{1,1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad R^{1,2} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad R^{1,3} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Given a heterogeneous information network, we aim at clustering different types of objects into different clusters based on the link relations with regard to not only the homogenous types but also the heterogeneous types.

We give the main notation system of this paper in Table 2.

3.2 Problem transformation

Inspired by Chakrabarti et al. [7], Papadimitriou et al. [37,38], we view the process of multi-type clustering as condensing the whole heterogeneous network without information loss. Since a heterogeneous information network is described by a group of relation matrices, the objective of clustering the heterogeneous network turns into minimizing the total information used to describe the group of relation matrices. In order to accomplish this objective, given a heterogeneous network $\mathcal{G} = (\mathcal{R}, \mathcal{V})$, we first concatenate the related relation matrices into larger ones based on the row dimensions they shared. An example of this process in a heterogeneous information network of four types of objects is given in Fig. 2. In this heterogeneous information network (shown in Fig. 2a), we assume that the objects of the same type have the homogeneous links, and the objects between different types of objects have heterogeneous links. Then, based on the ascending order of the object type indices, we get 4 new matrices (shown in Fig. 2b) by concatenating the relation matrices into larger ones. In order to further condense the group of relation matrices, we adjust the position of the rows and columns of the group of concatenated relation matrices. Then, we describe each concatenated matrix by a two-part code, which is schema description complexity and code length.

Table 2 Main notation system

Symbol	Definition
$\mathcal{V} = \{V^i\}_{i=1}^k$	Set of different types of objects in $\mathcal{G}$
$\mathcal{R} = \{R^{s,t}\}_{s,t=1}^k$	Link relation matrices between different types of objects in $\mathcal{G}$
$\mathcal{G} = (\mathcal{V}, \mathcal{R})$	Heterogeneous information network
$R^{s,t} \in \mathcal{R}$	Link relations between the objects of types T^s and T^t
T^i	The ith type of objects
$\{\Psi_i\}_{i=1}^k$	Schema in the concatenated relation matrices
J^i	Set of clusters of type T^i
J_p^i	The pth cluster in J^i
$R_{p,q}^{i,j}$	Link relations between the specified p-th group of objects of type T^i and the specified qth group of objects of type T^j
$n_p^{i,j}$ and $n_q^{i,j}$	Row and column dimension of $R_{p,q}^{i,j}$, respectively
$N(R_{p,q}^{i,j})$	Number of elements in matrix $R_{p,q}^{i,j}$
$H(R_{p,q}^{i,j})$	Binary Shannon entropy for matrix $R_{p,q}^{i,j}$
$\mathcal{T}_s(\mathcal{R})$	Cost for encoding the schema of $\mathcal{R}$
$\mathcal{T}_c(\mathcal{R})$	Cost for encoding all of the sub-matrices in $\mathcal{R}$
$\mathcal{T}(\mathcal{R})$	Cost for encoding $\mathcal{R}$
$T(R)$	Objective function for encoding matrix R with the least bits
$N_h(R_{p,q}^{i,j})$	Number of elements whose value equal to h in $R_{p,q}^{i,j}$
$P_h(R_{p,q}^{i,j})$	Density of "h" in $R_{p,q}^{i,j}$
$w(p, q, j)$	Discriminative column objective function
$w'(q, p, j)$	Discriminative row objective function

Before presenting the encoding schema in detail, we introduce the notation system used in this paper as follows. There are k types of objects, e.g., $T^1, T^2, \ldots, T^k$ in the heterogeneous network. For each type of T^i, there are m^i objects will be clustered into l^i clusters. Let us suppose $\{\Psi_i\}_{i=1}^k$ represents the schema of adjustment in the concatenated relation matrices, e.g., $\Psi_i : \{1, 2, \ldots, m^i\} \to \{1, 2, \ldots, l^i\}(1 \le i \le k)$. J^i denotes the set of clusters of type T^i, and J_p^i denotes the pth cluster in J^i. Besides, $R_{p,q}^{i,j}(1 \le i \le j \le k, 1 \le p \le l_i, 1 \le q \le l_j)$ stands for the sub-matrix created by the elements at the crossing of pth group of rows and the qth group of columns in the relation matrix $R^{i,j}$. Moreover, $n_p^{i,j}$ and $n_q^{i,j}$ denote the row and column dimensions of $R_{p,q}^{i,j}$, respectively. From the view of objects, $R_{p,q}^{i,j}$ expresses the link relations between the specified pth group of objects of type T^i and the specified qth group of objects of type T^j.

3.2.1 Description complexity for schema

Encoding the whole schema information includes the following six parts. We first need to record the number of types of objects, which costs $\log^* k$ bits [15].

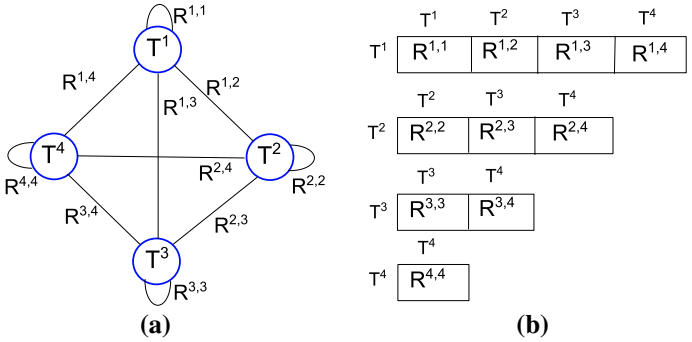


Fig. 2 General model for a heterogeneous information network with four types of objects and the corresponding concatenated relation matrices. **a** The different relations among different types of objects. **b** The concatenated relation matrices

Secondly, we need to record the row dimensions and column dimensions of all the concatenated relation matrices. Since the column dimension of concatenated relation matrix is combined by different relation matrices, the column dimension of each concatenated relation matrix needs to be recorded separately. Therefore, encoding the dimension information of all of the concatenated relation matrices costs $\sum_{t=0}^{k-1}\sum_{i=t+1}^{k}(\log^* m^i)$ bits.[2] Thirdly, encoding the permutations of rows and columns in the concatenated relation matrices costs $\sum_{t=0}^{k-1}\sum_{i=t+1}^{k}(m^i\lceil\log m^i\rceil)$ bits. Fourthly, encoding the number of groups, which stand for the number of clusters, i.e., $l_i (1 \le l \le k)$, in different dimensions of the relation matrices costs $\sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\lceil\log m^i\rceil$ bits. Fifthly, if we assume the mapping of m^i objects into l_i clusters is equally likely, recording the number rows or columns in all of the groups belonging to T^i type, i.e., the number of objects within each cluster of the same type T^i, costs $\left\lceil\log\binom{m^i}{m^i_1,\ldots,m^i_{l_i}}\right\rceil$ bits. Consequently, encoding all the numbers of rows or columns in all of concatenated relation matrix groups costs $\sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\left\lceil\log\binom{m^i}{m^i_1,\ldots,m^i_{l_i}}\right\rceil$ bits. Finally, recording the number of 1's of each sub-matrix $R^{i,j}_{p,q}$ costs $\lceil\log(n^{i,j}_p n^{i,j}_q+1)\rceil$ bits, where $n^{i,j}_p$ and $n^{i,j}_q$ are the row and column dimensions of $R^{i,j}_{p,q}$, respectively. Therefore, recording the total 1's of all the sub-matrices in the concatenated relation matrices costs the $\sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\sum_{j=i}^{k}\sum_{p=1}^{l_i}\sum_{q=1}^{l_j}\lceil\log(n^{i,j}_p n^{i,j}_q+1)\rceil$ bits.

In total, encoding the schema of the concatenated relation matrices costs $\mathcal{T}_s(\mathcal{R})$ bits that equal to summing all above values and is described as follows.

$$\begin{aligned}\mathcal{T}_s(\mathcal{R}) = \log^* k &+ \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\left(\log^* m^i\right) + \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\left(m^i\left\lceil\log m^i\right\rceil\right)\\ &+ \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\left\lceil\log m^i\right\rceil + \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\left\lceil\log\binom{m^i}{m^i_1,\ldots,m^i_{l_i}}\right\rceil\\ &+ \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\sum_{j=i}^{k}\sum_{p=1}^{l_i}\sum_{q=1}^{l_j}\left\lceil\log(n^{i,j}_p n^{i,j}_q+1)\right\rceil\end{aligned} \tag{1}$$

[2] All logarithms are based on 2 in this paper.

3.2.2 Code length for concatenated matrices

Assume that each element in sub-matrix $R_{p,q}^{i,j}$ is independent and identically distributed (IID) drawn from a Bernoulli distribution [7]. According to the Shannon entropy theory, encoding sub-matrix $R_{p,q}^{i,j}$ costs $N(R_{p,q}^{i,j})H(R_{p,q}^{i,j})$ bits, where $N(R_{p,q}^{i,j})$ denotes the number of elements in $R_{p,q}^{i,j}$ and $H(R_{p,q}^{i,j})$ denotes the entropy of $R_{p,q}^{i,j}$. Hence, it takes $\sum_{p=1}^{l_i}\sum_{q=1}^{l_j} N(R_{p,q}^{i,j})H(R_{p,q}^{i,j})$ bits to describe all sub-matrices in $R^{i,j}$. Therefore, encoding all of the sub-matrices in the concatenated relation matrices costs $\mathcal{T}_c(\mathcal{R})$ bits described as follows:

$$\begin{aligned}\mathcal{T}_c(\mathcal{R}) &= \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\sum_{j=i}^{k}\sum_{p=1}^{l_i}\sum_{q=1}^{l_j} N\left(R_{p,q}^{i,j}\right) H\left(R_{p,q}^{i,j}\right) \\ &= \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\sum_{j=i}^{k}\sum_{p=1}^{l_i}\sum_{q=1}^{l_j}\sum_{h=0}^{1} N_h(R_{p,q}^{i,j}) \log\left(\frac{N\left(R_{p,q}^{i,j}\right)}{N_h\left(R_{p,q}^{i,j}\right)}\right) \\ &= \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\sum_{j=i}^{k}\sum_{p=1}^{l_i}\sum_{q=1}^{l_j}\sum_{h=0}^{1} N_h\left(R_{p,q}^{i,j}\right) \log\left(\frac{1}{P_h\left(R_{p,q}^{i,j}\right)}\right)\end{aligned} \tag{2}$$

where $N_h(R_{p,q}^{i,j})$ represents the number of elements whose value equal to h in $R_{p,q}^{i,j}$ and $P_h(R_{p,q}^{i,j})$ represents the density of "h" in $R_{p,q}^{i,j}$.

3.2.3 Objective function

To conclude, encoding the set of relation matrices $\mathcal{R}$ of the heterogeneous information network $\mathcal{G} = (\mathcal{V}, \mathcal{R})$ costs $\mathcal{T}(\mathcal{R})$ bits described as:

$$\begin{aligned}\mathcal{T}(\mathcal{R}) &= T_s(\mathcal{R}) + T_c(\mathcal{R}) \\ &= \log^* k + \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\left(\log^* m^i\right) + \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\left(m^i\left\lceil\log m^i\right\rceil\right) \\ &\quad + \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\left\lceil\log m^i\right\rceil + \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\left\lceil\log\left(\begin{smallmatrix} m^i \\ m_1^i,\ldots,m_{l_i}^i\end{smallmatrix}\right)\right\rceil \\ &\quad + \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\sum_{j=i}^{k}\sum_{p=1}^{l_i}\sum_{q=1}^{l_j}\left\lceil\log\left(n_p^{i,j}n_q^{i,j}+1\right)\right\rceil \\ &\quad + \sum_{t=0}^{k-1}\sum_{i=t+1}^{k}\sum_{j=i}^{k}\sum_{p=1}^{l_i}\sum_{q=1}^{l_j}\sum_{h=0}^{1} N_h\left(R_{p,q}^{i,j}\right)\log\left(\frac{1}{P_h\left(R_{p,q}^{i,j}\right)}\right)\end{aligned} \tag{3}$$

For the simplicity of the discussion, in the rest of the paper, we focus on a basic type of heterogeneous network, i.e., star structure heterogeneous information network, in which the homogenous relation only exists within the objects of the central type, and the heterogeneous relations exist between the objects of the central type and the other types. That is to say, only $R^{1,j}$ needs to be considered as $R^{i,j}(i \neq 1)$ equals to zero. However, all of the following discussions of the paper can be easily extended to our general model. An example of the

star structure heterogeneous information network is shown in Fig. 1. As a result, we have Eq. (4) for computing the total bits used to describe the star structure heterogeneous network $G = (V, R)$.

$$\begin{aligned}\mathcal{T}(R) = &\log^* k + \sum_{i=1}^{k}\left(\log^* m^i\right) + \sum_{i=1}^{k}\left(m^i \lceil \log m^i \rceil\right) \\ &+ \sum_{i=1}^{k}\left\lceil \log m^i \right\rceil + \sum_{i=1}^{k}\left\lceil \log \binom{m^i}{m_1^i, \ldots, m_{l_i}^i} \right\rceil \\ &+ \sum_{j=1}^{k}\sum_{p=1}^{l_1}\sum_{q=1}^{l_j}\left\lceil \log\left(n_p^{1,j} n_q^{1,j} + 1\right)\right\rceil \\ &+ \sum_{j=1}^{k}\sum_{p=1}^{l_1}\sum_{q=1}^{l_j}\sum_{h=0}^{1} N_h\left(R_{p,q}^{1,j}\right) \log\left(\frac{1}{P_h\left(R_{p,q}^{1,j}\right)}\right)\end{aligned} \tag{4}$$

We notice that the first to the fourth terms in Eq. (4) are constant. The more important part is the last term of Eq. (4), which is used to describe the code length of the concatenated matrix, which dominates the whole equation. Moreover, the larger scale of the heterogeneous information network, the more dominant of the last term in Eq. (4). Therefore, we give our objective function as follows.

$$T(R) = \min\left\{\sum_{j=1}^{k}\sum_{p=1}^{l_1}\sum_{q=1}^{l_j}\sum_{h=0}^{1} N_h\left(R_{p,q}^{1,j}\right) \log\left(\frac{1}{P_h\left(R_{p,q}^{1,j}\right)}\right)\right\} \tag{5}$$

In Fig. 3, we give two different multi-type clustering results for the heterogeneous information network in Fig. 1. Obviously, the multi-type clustering result in Fig. 3b is much better than the multi-type clustering result in Fig. 3a. That is because the clusters in Fig. 3b not only have the dense connection among homogeneous objects, e.g., the clusters on object set V^1, but also have the dense connection among heterogeneous objects, e.g., the clusters on object sets V^2 and V^3. If we only consider the homogeneous relation of Fig. 1, the quality of clustering result in Fig. 3a is acceptable. However, if we consider both the homogeneous and heterogeneous relations of Fig. 3, the quality of multi-type clustering in Fig. 3a, such as the clusters in object sets V^2 and V^3, is not so attractive.

We notice that, on the one hand, based on the multi-type clustering result of Fig. 3a, the total bits computed by Eq. (5) are 64.1. On the other hand, based on the multi-type clustering result of Fig. 3b, the total bits computed by Eq. (5) are 27.3, which is much less than the bits based on the multi-type clustering result in Fig. 3a. This demonstrates that the lesser the value of Eq. (5), the better multi-type clustering results in the corresponding heterogeneous information network. Hence, our objective function provides a good criterion for the quality of the multi-type clustering in the heterogeneous information network.

4 The framework of multi-type clustering in heterogeneous information network

Given the objective function in Eq. (5), finding the best schema, i.e., a set of $\{\Psi_i\}_{i=1}^{k}$, in a heterogeneous information network is NP-hard. That is because it is even NP-hard to find the best schema in the bipartite graph [7,10]. Therefore, we design local schema search (LSS-H)

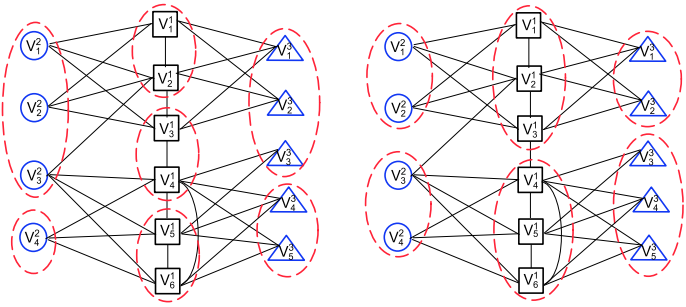


Fig. 3 Two different types of clustering results on the heterogeneous information network shown in Fig. 1. **a** One possible clustering result only based on the homogeneous relation. **b** The clustering result based on both homogeneous and heterogeneous relations

and global schema search (GSS-H) to find the appropriate optimization in the heterogeneous information network. LSS-H minimizes the objective function based on the assumption that the numbers of clusters for different types of objects, i.e., $\{l_i\}_{i=1}^{k}$, are given, while for GSS-H, the numbers of clusters for different types of objects are discovered automatically. Based on the appropriate optimization discovered by LSS-H or GSS-H, the overlapping schema search (OSS-H), which is used to detect the overlapping structures of clusters, is proposed.

4.1 Local schema search

In the algorithm of LSS-H, we assume the set of relation matrix of the heterogeneous information network, i.e., $\{R^{1,j}\}_{j=1}^{k}$, and the number of clusters for each type of objects in the heterogeneous information network, i.e., $l_i (1 \leq i \leq k)$, is given. Initially, we concatenate the relation matrices $R^{1,1}, R^{1,2}, \ldots, R^{1,k}$ to form a $m^1 \times \sum_{i=1}^{k} m^i$ matrix $R^{1,\ldots,k}$. Then, we initialize the schema $\{\Psi_i\}_{i=1}^{k}$ by randomly assigning $m^i (1 \leq i \leq k)$ objects of T^i type into l_i clusters. This initialization step also incurs the adjustment of rows and columns in the concatenated relation matrix $R^{1,\ldots,k}$. In order to minimize the objective function described by Eq. (5) and get the appropriate optimization schema, we adjust the columns and rows of $R^{1,\ldots,k}$ by iteratively performing the following steps.

Firstly, at iteration t, we fix the row mapping, i.e, $\Psi_1^{(t)}$, of $(R^{1,\ldots,k})^{(t)}$. Then, for each column c in each relation matrix, i.e., $(R^{1,j})^{(t)} (2 \leq j \leq k)$, we compute the new column group $\Psi_j^{(t+1)}(c)$ for column c by the equation below:

$$\Psi_j^{(t+1)}(c) = \arg \min_{1 \leq q \leq l_j} \left\{ -\sum_{p=1}^{l_1} \sum_{h=0}^{1} N_h(c_p) \log \left(P_h \left(\left(R_{p,q}^{1,j} \right)^{(t)} \right) \right) \right\} \tag{6}$$

where c_p represents the pth part of column c split by the row mapping $\Psi_1^{(t)}$. Next, we assign column c into the new column group $\Psi_j^{(t+1)}(c)$, which reduces the value of objective function.

Secondly, at iteration $(t+1)$, except for the column mapping, i.e., $\Psi_1^{(t+1)}$, of relation matrix $(R^{1,1})^{(t+1)}$, we fix all of the column mappings, i.e., $\Psi_i^{(t+1)} (2 \leq i \leq k)$, in the concatenated matrix $(R^{1,\ldots,k})^{(t+1)}$. Then, for each row r of the concatenated matrix $(R^{1,\ldots,k})^{(t+1)}$, we assign row r into the new row group $\Psi_1^{(t+2)}(r)$ computed by

$$\begin{aligned}\Psi_1^{(t+2)}(r) = \arg\min_{1\le p\le l_1}\Bigg\{ & -\sum_{j=1}^{k}\sum_{q=1}^{l_j}\sum_{h=0}^{1} N_h\left(r_q^{1,j}\right)\log\left(P_h\left(\left(R_{p,q}^{1,j}\right)^{(t+1)}\right)\right) \\ & -\sum_{q=1}^{l_1}\sum_{h=0}^{1} N_h\left(r_q^{1,1}\right)\log\left(P_h\left(\left(R_{p,q}^{1,1}\right)^{(t+1)}\right)\right) \\ & +\sum_{h=0}^{1} 2N_h(d_{r,r})\log\left(P_h\left(\left(R^{1,1}_{p,\Psi_1^{(t+1)}(r)}\right)^{(t+1)}\right)\right) \\ & -\sum_{h=0}^{1}\log\left(P_h\left(\left(R_{p,p}^{1,1}\right)^{(t+1)}\right)\right)\Bigg\}\end{aligned} \tag{7}$$

where $r^{1,j}(1 \le j \le k)$ is the part of row r located in relation matrix $(R^{1,j})^{(t+1)}$, $r_p^{1,j}$ is the pth part of $r^{1,j}$ spliced by column mapping $\Psi_j^{(t+1)}(2 \le j \le k)$, and $d_{r,r}$ is the element at the crossing of row $r^{1,1}$ and column $c^{1,1}$. Here, $c^{1,1}$ is the symmetrical column of $r^{1,1}$ in $(R^{1,1})^{(t+1)}$. Because $R^{1,1}$ is symmetrical matrix that reflects the homogeneous relation of object type T^1, the mapping $\Psi_1^{(t+1)}$ is also operated on the column dimension of $(R^{1,1})^{(t+1)}$ in the concatenated matrix $(R^{1,\ldots,k})^{(t+1)}$.

We keep in mind that there are two types of relation matrices, which form the concatenated relation matrix. The first one is symmetric matrix $R^{1,1}$, which describes the central type of objects in the star schema. This type of matrix reflects the link relation between the same type of objects. The second type of matrices such as $R^{1,2}$, $R^{1,3}$, …, $R^{1,k}$ are asymmetric, which describe the relations between different types of objects in the heterogeneous network. Then, two types of relation matrices form the concatenated matrix $R^{1,\ldots,k}$ according to the rule described in Fig. 2b. Hence, it is easy to understand that in Eq. (7), the first term denotes the cost of shifting row r to the new row group $\Psi_1^{(t+1)}(r)$, while the second term denotes the cost of shifting column c, which is symmetrical to row r in $(R^{1,1})^{(t+1)}$, to the new column group $\Psi_1^{(t+1)}(r)$, and the last two terms are the "double-counting" of the element $d_{r,r}$ in $(R^{1,1})^{(t+1)}$.

The above two steps are repeated iteratively until the convergence of the objective function. The description of LSS-H is presented in Algorithm 1. For Algorithm 1, we have the following theorems.

Theorem 1 *At iteration t ($t \ge 1$), assigning any column c of the relation matrix $(R^{1,j})^{(t)}$ ($2 \le j \le k$) into the new column group $\Psi_j^{(t+1)}(c)$ defined by* Eq. (6) *decreases the objective function, i.e.,*

$$T\left(R^{(t+1)}\right) \le T\left(R^{(t)}\right).$$

Proof

$$\begin{aligned}T(R^{(t)}) &= \sum_{j=1}^{k}\sum_{p=1}^{l_1}\sum_{q=1}^{l_j}\sum_{h=0}^{1} N_h\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right)\log\left(\frac{1}{P_h\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right)}\right) \\ &= \sum_{j=1}^{k}\sum_{p=1}^{l_1}\sum_{q=1}^{l_j}\sum_{h=0}^{1}\left[\sum_{c:\Psi_j^{(t)}(c)=q} N_h(c_p)\right]\log\left(\frac{1}{P_h\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right)}\right)\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{k}\sum_{q=1}^{l_j}\sum_{c:\Psi_j^{(t)}(c)=q}\left[\sum_{p=1}^{l_1}\sum_{h=0}^{1} N_h(c_p)\log\left(\frac{1}{P_h\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right)}\right)\right] \\
&\overset{(1)}{\geq} \sum_{j=1}^{k}\sum_{q=1}^{l_j}\sum_{c:\Psi_j^{(t)}(c)=q}\left[\sum_{p=1}^{l_1}\sum_{h=0}^{1} N_h(c_p)\log\left(\frac{1}{P_h\left(\left(R_{p,\Psi_j^{(t+1)}(c)}^{1,j}\right)^{(t)}\right)}\right)\right] \\
&= \sum_{j=1}^{k}\sum_{q=1}^{l_j}\sum_{c:\Psi_j^{(t+1)}(c)=q}\left[\sum_{p=1}^{l_1}\sum_{h=0}^{1} N_h(c_p)\log\left(\frac{1}{P_h\left(\left(R_{p,\Psi_j^{(t+1)}(c)}^{1,j}\right)^{(t)}\right)}\right)\right] \\
&= \sum_{j=1}^{k}\sum_{p=1}^{l_1}\sum_{q=1}^{l_j}\sum_{h=0}^{1}\left[\sum_{c:\Psi_j^{(t+1)}(c)=q} N_h(c_p)\right]\log\left(\frac{1}{P_h\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right)}\right) \\
&= \sum_{j=1}^{k}\sum_{p=1}^{l_1}\sum_{q=1}^{l_j}\sum_{h=0}^{1} N_h((R_{p,q}^{1,j})^{(t+1)})\log\left(\frac{1}{P_h\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right)}\right) \\
&\overset{(2)}{\geq} \sum_{j=1}^{k}\sum_{p=1}^{l_1}\sum_{q=1}^{l_j}\sum_{h=0}^{1} N_h((R_{p,q}^{1,j})^{(t+1)})\log\left(\frac{1}{P_h\left(\left(R_{p,q}^{1,j}\right)^{(t+1)}\right)}\right) \\
&= T(R^{(t+1)})
\end{aligned}
\tag{8}
$$

In the above deduction of E. (8), inequality (1) holds for the Eq. (6), and inequality (2) holds for the nonnegativity of the Kullback–Leibler distance.

Theorem 2 *At iteration* $(t+1)(t \geq 1)$*, assigning any row* r *of the concatenated matrix* $(R^{1,\ldots,k})^{(t+1)}$ *into the new row group* $\Psi_1^{(t+1)}(r)$ *defined by* Eq. (7) *decreases the objective function, i.e.,*

$$T\left(R^{(t+2)}\right) \leq T\left(R^{(t+1)}\right).$$

Proof The proof of Theorem 2 is very similar to Theorem 1, and the proof is omitted due to space limitation.

Theorem 3 *The objective function defined in* Eq. (5) *converges to a local minimization.*

Proof On the one hand, the lower bound of the objective function is 0. That is because any relation matrix needs at least 0 bits to describe its code length. On the other hand, Theorems 1 and 2 guarantee the objective function keeps not increasing during the iterative process in Algorithm 1. Hence, considering these two respects, the objective function converges to a local minimization in Algorithm 1.

Input : Relation matrix set $\{R^{1,j}\}_{j=1}^{k}$;
Cluster number set $\{l_i\}_{i=1}^{k}$.
Output : Appropriately optimal schema $\{\Psi_i\}_{i=1}^{k}$

Concatenate all relation matrices in $\{R^{1,j}\}_{j=1}^{k}$ to form a $m^1 \times \sum_{i=1}^{k} m^i$ matrix $R^{1..k}$;
Set $t = 0$; Initialization $\{\Psi_i^{(t)}\}_{i=1}^{k}$;
Update $(R^{1..k})^{(t)}$ according to $\{\Psi_i^{(t)}\}_{i=1}^{k}$;
repeat
  Fixing the row mapping $\Psi_1^{(t)}$ for $(R^{1..k})^{(t)}$;
  foreach $(R^{1,j})^{(t)} (2 \le j \le k)$ *in* $(R^{1..k})^{(t)}$ **do**
    foreach *column c in* $(R^{1,j})^{(t)}$ **do**
      Compute column mapping $\Psi_j^{(t+1)}(c)$ according to Equation(6) ;
    end
  end
  $\Psi_1^{(t+1)} \leftarrow \Psi_1^{(t)}$;
  Update $(R^{1..k})^{(t+1)}$ according to $\{\Psi_j^{(t+1)}\}_{j=2}^{k}$;
  Fixing the column mapping set $\{\Psi_i^{(t+1)}\}_{i=2}^{k}$ for $(R^{1..k})^{(t+1)}$;
  foreach *row r in* $(R^{1..k})^{(t+1)}$ **do**
    Compute row mapping $\Psi_1^{(t+2)}(r)$ according to Equation(7) ;
  end
  $\{\Psi_j^{(t+2)}\}_{j=2}^{k} \leftarrow \{\Psi_j^{(t+1)}\}_{j=2}^{k}$;
  Update $(R^{1..k})^{(t+2)}$ according to $\Psi_1^{(t+2)}$;
  $t = t + 2$;
until *convergence*;

Algorithm 1: Local schema search (LSS-H)

4.2 Global schema search

GSS-H takes a top-down greedy search strategy to automatically detect the clusters for each type of objects. The description of GSS-H is presented in Algorithm 2. Similar to LSS-H, all of the relation matrices $R^{1,j}$ are concatenated to form a $m^1 \times \sum_{i=1}^{k} m^i$ matrix $R^{1,\ldots,k}$ at the beginning. Then, each type of objects is initialized as a single cluster. In other words, for each relation matrix $R^{1,j}$ in $R^{1,\ldots,k}$, all of the columns in $R^{1,j}$ form a single-column group, and all of the rows in $R^{1,\ldots,k}$ form a single-row group. Each iteration of GSS-H can be divided as the following steps.

Initially, at iteration t, for a specific type of cluster set, let say $J^j (1 \le j \le k)$, we add a new empty cluster $J^j_{l_j+1}$–J^j. Then, we select the cluster which has the maximum average code length per object among all clusters in J^j. For clusters of $T^j (2 \le j \le k)$ type, we compute the column cluster index q, whose cluster has the average maximum code length per object by Eq. (9). Since the objects of T^1 type have relations with all of the other types of objects, the cluster index p, whose cluster has the maximum average entropy per object, is computed by Eq. (10).

$$q = \arg \max_{1 \le q \le l_j} \sum_{p=1}^{l_1} \sum_{h=0}^{1} \frac{N_h\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right) \log \frac{1}{P_h\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right)}}{n_q^{1,j}} \tag{9}$$

Input : Relation matrix set $\{R^{1,j}\}_{j=1}^{k}$
Output : Appropriately optimal schema $\{\Psi_i\}_{i=1}^{k}$;
Cluster number set $\{l_i\}_{i=1}^{k}$

Concatenate relation matrices $\{R^{1,j}\}_{j=1}^{k}$ to form a $m^1 \times \sum_{i=1}^{k} m^i$ matrix $R^{1,\ldots,k}$;
Set $t = 0$; $l_1 = l_2 \cdots = l_k = 1$;
Initialization $\{\Psi_i^{(t)}\}_{i=1}^{k}$ according to $\{l_i\}_{i=1}^{k}$;
Update $(R^{1,\ldots,k})^{(t)}$ according to $\{\Psi_i^{(t)}\}_{i=1}^{k}$;
repeat
  foreach *cluster set* $J^j (1 \le j \le k)$ **do**
    repeat
      Add an empty cluster $J^j_{l_j+1}$ to J^j ;
      if $j > 1$ **then**
        Compute index q by Eq. (9) ;
        $MeaFunc \leftarrow$Eq. (11) ;
      else
        Compute index q by Eq. (10) ;
        $MeaFunc \leftarrow$Eq. (12) ;
      end
      Randomly select half of the objects from J^j_q and place them into $J^j_{l_j+1}$;
      `Switching`$(J^j_q, J^j_{l_j+1}, MeaFunc)$;
      Update $(R^{1..k})^{(t)}$ and l_j ;
      $\{\Psi_i^{(t+1)}\}_{i=1}^{k} \leftarrow$ `LSS`$((R^{1..k})^{(t)}, \{l_i\}_{i=1}^{k})$;
      Update $(R^{1..k})^{(t+1)}$;
      $t = t + 1$;
    until *the total cost does not increase*;
  end
until *convergence*;

Algorithm 2: Global schema search (GSS-H)

$$p = \arg \max_{1 \le p \le l_1} \sum_{j=1}^{k} \sum_{q=1}^{l_j} \sum_{h=0}^{1} \frac{N_h\left(\left(R^{1,j}_{p,q}\right)^{(t)}\right) \frac{1}{P_h\left(\left(R^{1,j}_{p,q}\right)^{(t)}\right)}}{n_p^{1,1}} \tag{10}$$

Subsequently, we randomly choose half of the objects from J^j_q and move them into the new cluster $J^j_{l_j+1}$. In order to make the objects in J^j_q have denser linking relation and decrease the total code length, we evaluate the decrease in objective function by moving any object $c \in J^j_q$ to the new cluster $J^j_{l_j+1}$. In this test, we need to consider clusters of $T^j (2 \le j \le k)$ type and clusters of T^1 type separately. For clusters of T^j type, the decrease in objective function by moving object c from cluster J^j_q to cluster $J^j_{l_j+1}$ is computed by

$$\begin{aligned} \Delta T_{c \to \left(J^j_{l_j+1}\right)^{(t)}} = \sum_{p=1}^{l_1} \Bigg\{ & N\left(\left(R^{1,j}_{p,q}\right)^{(t)}\right) H\left(\left(R^{1,j}_{p,q}\right)^{(t)}\right) \\ & + N\left(\left(R^{1,j}_{p,l_j+1}\right)^{(t)}\right) H\left(\left(R^{1,j}_{p,l_j+1}\right)^{(t)}\right) \end{aligned}$$

$$
\begin{aligned}
&- N\left(\left(R_{p,q}^{1,j}\right)_{-c}^{(t)}\right) H\left(\left(R_{p,q}^{1,j}\right)_{-c}^{(t)}\right) \\
&- N\left(\left(R_{p,l_j+1}^{1,j}\right)_{+c}^{(t)}\right) H\left(\left(R_{p,l_j+1}^{1,j}\right)_{+c}^{(t)}\right)\Bigg\} \qquad (11)
\end{aligned}
$$

where $(R_{p,q}^{1,j})_{-c}^{(t)}$ is the matrix $R_{p,q}^{1,j}$ without column c, and $(R_{p,l_j+1}^{1,j})_{+c}^{(t)}$ is the matric $R_{p,l_j+1}^{1,j}$ with column c at iteration t. Different from the clusters of $T^j (2 \le j \le k)$ type, the decrease in total code length by moving object $r \in J_p^1$ to $J_{l_1+1}^1$ can be computed by the following equation.

$$
\begin{aligned}
\Delta T_{r\to\left(J_{l_1+1}^1\right)^{(t)}} = \sum_{j=1}^{k}\sum_{q=1}^{l_j}\Bigg\{ & N\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right) H\left(\left(R_{p,q}^{1,j}\right)^{(t)}\right) \\
& + N\left(\left(R_{l_1+1,q}^{1,j}\right)^{(t)}\right) H\left(\left(R_{l_1+1,q}^{1,j}\right)^{(t)}\right) \\
& - N\left(\left(R_{p,q}^{1,j}\right)_{-r}^{(t)}\right) H\left(\left(R_{p,q}^{1,j}\right)_{-r}^{(t)}\right) \\
& - N\left(\left(R_{l_1+1,q}^{1,j}\right)_{+r}^{(t)}\right) H\left(\left(R_{l_1+1,q}^{1,j}\right)_{+r}^{(t)}\right)\Bigg\} \qquad (12)
\end{aligned}
$$

All of the objects in J_q^j or J_p^1, whose assignment can decrease the objective function, are assigned into the new cluster $J_{l_j+1}^j$ or $J_{l_1+1}^j$. Similar adjustment is also repeated for each object

Input : Cluster J_p and cluster J_q;
Measure function `MeaFunc`.
Output : Cluster J_p' and Cluster J_q' which has less total code length than J_p and J_q

```
Initialize each c ∈ J_p and c' ∈ J_q as unvisited ;
repeat
    foreach c ∈ J_p ∩ c.visited == false do
        ΔT_{c→J_q} ← MeaFunc(c, J_q, J_p) ;
        if ΔT_{c→J_q} > 0 then
            Move c from J_p to J_q ;
            c.visited = true ;
        else
            foreach c' ∈ J_q ∩ c'.visited == false do
                ΔT_{c'→J_p} ← MeaFunc(c', J_p, J_q) ;
                if ΔT_{c'→J_p} > 0 then
                    Move c' from J_q to J_p ;
                    c'.visited = true ;
                else
                    break ;
                end
            end
        end
    end
until each c ∈ J_p and c' ∈ J_q have been visited;
J_p' ← J_p; J_q' ← J_q;
return J_p', J_q';
```

Algorithm 3: Switching

$c' \in J^j_{l_j+1}$ or $r' \in J^1_{l_1+1}$. The detail description of switching objects between two clusters is given in Algorithm 3. This split process makes the number of clusters of $T^j (1 \leq j \leq k)$ type increase from l_j to $(l_j + 1)$.

Finally, the LSS-H is applied to further adjust the assignment of objects among different clusters. The above three stages are repeated iteratively until the total coding cost of the heterogeneous information network does not increase any more. Then, we choose another type of cluster set and continue the above steps.

All of the above steps are repeated until the convergence of the objective function. We notice that the convergence of GSS-H can be guaranteed. On the one hand, the splitting process (steps 8–17) in Algorithm 2 decreases the objective function. That is because splitting a matrix decreases the total code length has already been proved in [7]. Moreover, the process of switching objects. i.e., Algorithm 3, also decreases the total code length. On the other hand, each LSS-H further (step 19) decreases the objective function. Since the total code length dominates the whole coding cost of the heterogeneous information network, the continuous decreasing in the code length converges the total coding cost to a local optimization. Thus, Algorithm 2 decreases the objective function at each iteration and converges to a local optimization.

4.3 Overlapping schema search

As we can see, LSS-H and GSS-H co-cluster different types of objects simultaneously in two different situations, respectively. In other words, after the non-overlapping multi-type clustering processes, the objects which have closer relations in the heterogeneous network are co-clustered together, while the objects which have the looser relations are separated into different clusters. From the viewpoint of the relation matrix, this process makes some sub-matrices of the relation matrix very dense and the other sub-matrices very sparse. In order to clearly explain our overlapping strategy, a single relation matrix $R^{1,j} (2 \leq j \leq k)$, before and after the multi-type clustering, is given in Fig. 4a and b, respectively.

It is clear that there are four row clusters of T^1 type and four column clusters of $T^j (j > 1)$ type in Fig. 4b. We notice that, for any pair of row clusters, there must be some pairs of blocks, in which each pair of block is in the same column group, having relatively opposite density. That is because, for any pair of row clusters, if any pair of blocks in the same column group have very similar density, the two row clusters have to be merged to form a single-row cluster during the multi-type clustering process, because the objects in these two row clusters have

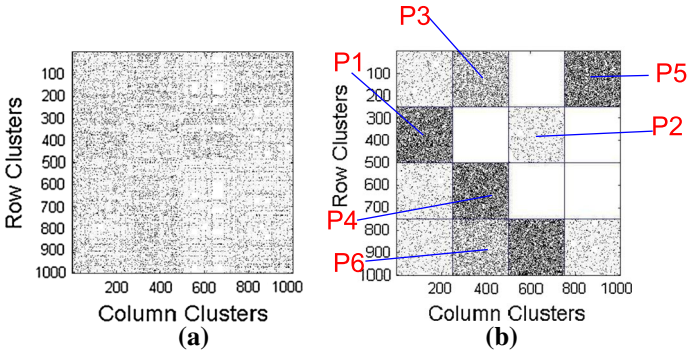


Fig. 4 A relation matrix before and after multi-type clustering. **a** Original matrix, **b** co-clustered matrix

similar link relations between different column objects. Similar principle also holds for any pair of column clusters. For example, in Fig. 4b, it is obvious that we cannot find any pair of row clusters, in which all of the blocks in the same column group have very similar density. In other words, there are at least one pair of blocks, which are in the same column group, having relatively opposite density. Otherwise, the two row clusters should be combined. We notice that, compared to the other column objects, column objects located in $P3$ and $P5$ have denser links with the row objects of the first row clusters. Moreover, column objects located in $P5$ are more important than those column objects in $P3$ for distinguishing the first row clusters from other row clusters. That is because, firstly, $P5$ has higher density than $P3$, which means the column objects located in $P5$ have closer relation with the row objects of the first row cluster than these column objects located in $P3$. Secondly, from the viewpoint of different row clusters, the link relation between the column objects in $P5$ and row objects in the first row clusters is very close, while the link relation between the column objects in $P5$ and row objects of the other row clusters is relatively loose. However, this situation is different for block $P3$. In other words, even though the row objects of the first row cluster have relative close relation with the column objects located in $P3$, the objects in the third and fourth row clusters also have very close relation with the column objects in $P3$. For example, $P6$ is almost as dense as $P3$, while $P4$ is even much denser than $P3$. Similar analysis process tells us that the column objects in $P1$ are more important than column objects in $P2$ for separating the second row cluster from the other row clusters, and row objects in $P4$ are more important than row objects in $P3$ for separating the second column clusters from the other column clusters. The above density analysis process is referred to as the *density guided principle* for discriminative clusters.

Given a co-clustered relation matrix $R^{1,j}(1 \leq j \leq k)$, we measure the importance of objects in row cluster $J_p^1(1 \leq p \leq l_1)$ for separating column cluster $J_q^j(1 \leq q \leq l_j)$ from the other column clusters by using *discriminative column objective function* as follows:

$$w(p, q, j) = P_1\left(R_{p,q}^{1,j}\right) - \frac{1}{l_j}\sum_{i=1}^{l_j} P_1\left(R_{p,i}^{1,j}\right) \tag{13}$$

where $P_1(R_{p,q}^{1,j})$ measures the density of "1" in the sub-matrix $R_{p,q}^{1,j}$.

Symmetrically, we have *discriminative row objective function*, which is described in Eq. (14), for evaluating the importance of objects in column cluster $J_q^j(1 \leq q \leq l_j)$ for separating row cluster $J_p^1(1 \leq p \leq l_1)$ from the other row clusters.

$$w'(q, p, j) = P_1\left(R_{p,q}^{1,j}\right) - \frac{1}{l_1}\sum_{i=1}^{l_1} P_1\left(R_{i,q}^{1,j}\right) \tag{14}$$

Obviously, both $w(p, q, j)$ and $w'(q, p, j)$ range from -1 to 1. In fact, $w(p, q)$ reflects the contribution of the links between objects in cluster J_p^1 and objects in cluster J_q^j for the cohesion of the objects in cluster J_q. Moreover, the larger value of $w(p, q)$ means the more contribution of the links between objects in J_p^1 and objects in J_q^j for the forming of cluster J_q^j. For example, in Fig. 3b, if we consider all of the clusters of T^1 type, i.e., $\{v_1^1, v_2^1, v_3^1\}$, $\{v_4^1, v_5^1, v_6^1\}$, and the second cluster of T^2 type, i.e., $\{v_3^2, v_4^2\}$, we get $w(1, 2, 2) = -\frac{5}{6}$ and $w(2, 2, 2) = \frac{5}{6}$. That is to say, compared to the links between objects in $\{v_1^1, v_2^1, v_3^1\}$ and objects in $\{v_3^2, v_4^2\}$, the links between objects in $\{v_4^1, v_5^1, v_6^1\}$ and objects in $\{v_3^2, v_4^2\}$ contribute more for objects v_3^2 and v_4^2 to form a cluster. Similar analysis can be easily applied to $w'(q, p, j)$.

Definition 1 (*Discriminative column cluster*) For a heterogeneous information network $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, given the row cluster $J_p^1 \in J^1 (1 \leq p \leq l_1)$ and the column cluster $J_q^j \in J^j (2 \leq j \leq k, 1 \leq q \leq l_j)$, the cluster J_q^j is defined as the discriminative column cluster of row cluster J_p^1 iff. J_q^j contributes to the distinction of row cluster J_p^1 from the other row cluster $J_b^1 (b \neq p \cap 1 \leq b \leq l_1)$, i.e., $w(p, q, j) \geq 0$.

Definition 2 (*Discriminative row cluster*) For a heterogeneous information network $\mathcal{G} = (\mathcal{V}, \mathcal{R})$, given the column cluster $J_q^j \in J^j (2 \leq j \leq k, 1 \leq q \leq l_j)$ and the row cluster $J_p^1 \in J^1 (1 \leq p \leq l_1)$, the cluster J_p^1 is defined as the discriminative row cluster of column cluster J_q^j iff. J_p^1 contributes to the distinction of column cluster J_q^j from the other column cluster $J_d^j (d \neq q \cap 1 \leq d \leq l_j)$, i.e., $w'(q, p, j) \geq 0$.

Consequently, for any column object $c \in J_q^j (2 \leq j \leq k)$ of the star-structured heterogeneous information network, we use Eq. (15) to evaluate whether object c should also be placed into another cluster $J_t^j (1 \leq t \leq l_j \cap t \neq q)$ or not.

$$\begin{aligned} E_{oc}(c, t) &= \sum_{f \in F_1} w(f, t, j) \left(P_1 \left(c_f \right) - P_1 \left(R_{f,t}^{1,j} \right) \right) \\ &= \sum_{f \in F_1} \left(P_1 \left(R_{f,t}^{1,j} \right) - \frac{1}{l_j} \sum_{i=1}^{l_j} P_1 \left(R_{f,i}^{1,j} \right) \right) \left(P_1 \left(c_f \right) - P_1 \left(R_{f,t}^{1,j} \right) \right) \end{aligned} \quad (15)$$

where c_f represents the elements at the crossing of column c and fth row group of $R^{1,1}$, and F_1 is the index set of discriminative row clusters in $R^{1,1}$, i.e., $F_1 = \{f | w(f, t, j) \geq 0, 1 \leq f \leq l_1\}$. If $E_{oc}(c, t) \geq 0$, column objects c will not only be placed into its original column cluster J_q^j, but also placed into the column cluster J_t^j. From Eq. (15), we know that only the discriminative column clusters take part in the computation of row cluster in the overlapping process. Moreover, we notice that it is possible for $(P_1(c_f) - P_1(R_{f,t}^{1,j}))$ to be negative. If in this case, it means the link density between column object c and objects in row cluster J_f^j is lower than the link density between objects in column cluster J_t^j and objects in row cluster J_f^j.

In a star-structured heterogeneous information network, the objects of T^1 type have two kinds of link relations, which are homogeneous link relation among themselves and the heterogeneous link relations between objects of $T^j (2 \leq j \leq k)$ type, respectively. Thus, given the relation matrices $R^{1,1}, \ldots, R^{1,k}$, we have Eq. (16) for evaluating whether the row objects $r \in J_p^1$ need to be placed into row cluster J_s^1.

$$\begin{aligned} E_{or}(r, s) &= \sum_{j=1}^{k} \sum_{f \in F_j} w'(f, s, j) \left(P_1 \left(r_f \right) - P_1 \left(R_{s,f}^{1,j} \right) \right) \\ &= \sum_{j=1}^{k} \sum_{f \in F_j} \left(P_1 \left(R_{s,f}^{1,j} \right) - \frac{1}{l_1} \sum_{i=1}^{l_1} P_1 \left(R_{i,f}^{1,j} \right) \right) \left(P_1 \left(r_f \right) - P_1 \left(R_{s,f}^{1,j} \right) \right) \end{aligned} \quad (16)$$

where r_f is the interest of row r, fth is the column group of $R_{1,j}$, and F_j is the index set of discriminative column clusters in $R^{1,j}$, i.e., $F_j = \{w'(s, f, j) \geq 0, 1 \leq f \leq l_j\}$.

The detail of OSS-H is given in Algorithm 4. The second input parameter, i.e., $\{l_i\}_{i=1}^k$, is optional if we call GSS-H for detecting the non-overlapping schema. Besides, the order of calculating the overlapping row objects and overlapping column objects does not change the final cluster structures, since going through the row clusters and column clusters is solely based on the non-overlapping schema $\{\Psi_i\}_{i=1}^k$.

Input : Relation matrix set $\{R^{1,j}\}_{j=1}^k$;
Cluster number set $\{l_i\}_{i=1}^k$ (optional).
Output : Overlapping clusters set $\{J^i\}_{i=1}^k$

1 Call LSS-H($\{R^{1,j}\}_{j=1}^k$, $\{l_i\}_{i=1}^k$) or GSS-H($\{R^{1,j}\}_{j=1}^k$) for detecting non-overlapping schema $\{\Psi_i\}_{i=1}^k$;
2 **foreach** *cluster set* $J^j (1 \le j \le k)$ **do**
3 **foreach** *cluster* $J_q^j \in J^j$ **do**
4 **foreach** *object* $v \in J_q^j$ **do**
5 **foreach** $t (1 \le t \le l_j \cap t \ne q)$ **do**
6 **if** $j == 1$ **then**
7 $isOverlap \leftarrow E_{or}(v, t)$;
8 **else**
9 $isOverlap \leftarrow E_{oc}(v, t)$;
10 **end**
11 **if** $isOverlap \ge 0$ **then**
12 Copy object v into cluster J_t^j ;
13 **end**
14 **end**
15 **end**
16 **end**
17 **end**

Algorithm 4: Overlapping schema search (OSS-H)

5 Time complexity analysis

The computational complexity of LSS-H is $O(N_1(R^{1,\ldots,k}) \cdot I \cdot L)$, where I is the number of iteration of LSS-H, and L is the total number of clusters, i.e., $L = \sum_{j=1}^k l_j$. The analysis process is as follows. Firstly, at each iteration, adjusting a single-column c to the best column group $\Psi_j^{(t+1)}(c)$ in Eq. (6) is $O(N_1(c) \cdot l_j)$. Then, adjusting all of the columns in $R^{1,j}$, which corresponds to steps $7 \sim 9$, is $O(N_1(R^{1,j}) \cdot l_j)$. Thus, adjusting all of columns in each matrix of the edge types, i.e., $R^{1,j} (2 \le j \le k)$, is $O(\sum_{j=2}^k N_1(R^{1,j}).l_j)$. Secondly, adjusting all of the rows in $R^{1,\ldots,k}$ (steps 14–16) is $O(N_1(R^{1,k}) \cdot l_1)$. In total, each iteration costs $O(\sum_{j=2}^k N_1(R^{1,j}) \cdot l_j + N_1(R^{1,\ldots,k}) \cdot l_1)$, which can be rewritten as $O(N_1(R^{1,\ldots,k}) \cdot \sum_{j=1}^k l_j)$. Therefore, the computational complexity of LSS-H is $O(N_1(R^{1,\ldots,k}) I \sum_{j=1}^k l_j)$.

The computational complexity of GSS-H is $O(k \cdot N_1(R^{1,\ldots,k}) \cdot L^2)$. At each iteration of GSS-H, the time spent on $Switching$ function is $O(N(R^{1,j}))$. Moreover, at tth iteration , the time spent on LSS-H is $O(N_1(R^{1,\ldots,k}) I^{(t)} \sum_{j=1}^k l_j^{(t)})$, where $l_j^{(t)}$ is the number of type T^j clusters at tth iteration, and $I^{(t)}$ is the number of iterations of LSS-H in tth iteration of GSS-H. Usually, $I^{(t)}$ is <30 in practice. If we ignore the number of iterations of LSS-H, each iteration of GSS-H costs $O(N_1(R^{1,\ldots,k}) \sum_{j=1}^k l_j^{(t)})$. Hence, it costs $O(N_1(R^{1,\ldots,k})(\sum_{j=1}^k l_j) \cdot l_j)$ from

step 7 to step 22 for each iteration of GSS-H. Finally, the total time complexity of GSS-H is $O(k \cdot N_1(R^{1k}) \cdot (\sum_{j=1}^{k} l_j)^2)$.

OSS-H calls LSS-H or GSS-H for searching the appropriately optimal schema at different input situations. Moreover, going through each object in the heterogeneous information network and computing its overlapping function, i.e., steps 2–17, cost $O(\sum_{i=1}^{k} m^i)$, which is lower than the computational complexity of LSS-H and GSS-H. Therefore, the computational complexity of OSS-H is the same with LSS-H or GSS-H.

6 Experiments

In this section, we empirically demonstrate the effectiveness and efficiency of our proposed methods. We apply our methods to two important tasks, multi-type clustering for non-overlapping cluster structures and multi-type clustering for overlapping cluster structures in the heterogeneous information network. Moreover, we compare our methods with three state-of-art algorithms, which are NMF [8], SRC [32], and CMRF [6]. The description of the compared algorithms is presented in Sect. 2. For NMF and SRC, we have the same standard inputs. However, for CMRF, there are a list of parameters need to be set. In this paper, we carefully set the parameters according to the instruction of the source code provided by the original author. Specifically, the search strategies used for CMRF are "divisive" for objects of central type and "agglomerative" for objects of edge types, respectively. Each experiment is repeated 10 times, and the average is reported.

6.1 Data set description

6.1.1 Synthetic data sets

We generate the synthetic data set based on the real-world data set Classic3.[3] Classic3 contains three types of documents, which are MEDLINE (medical journals), CISI (information retrieval), and CRANFIELD (aerodynamics). The documents and words form a binary matrix of 3891×5896. In order to get the non-overlapping multi-type objects, we randomly choose 1000 documents from each type of documents and form a *document-word* matrix. Then, we take the "splitting strategy," which is also used in [25], to split the words. Concretely, we randomly permute the columns of the *document-word* matrix. Subsequently, we averagely divide the columns into two groups. Each group of columns is assumed as a type of objects. Consequently, we get two *document-word* matrices, which describe the relations of three types of objects.

In order to further get the overlapping multi-type objects, firstly we randomly pick 1000 documents of each type. Then, we randomly choose two documents, let say d_i and d_j, from two different types of documents T^i and $T^j (i \neq j)$. Next, we merge the two documents together to form a new document d_{ij}, which is annotated with both T^i and T^j. The number of new generated documents is controlled by overlapping percentage parameter, i.e., OV%. Finally, all of the documents are split and generated two *document-word* matrices as we discussed above.

[3] http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets.

6.1.2 Real-world data sets

We used two real-world data sets for non-overlapping multi-type clustering. The first one is *Newsgroup*,[4] which contains 20 groups of documents from 20 news groups. The second data set is from [21],[5] which is a composition of *oh15*, *re0*, and *WAP*. Particularly, *oh15* is from OHSUMED collection, which is a subset of MEDLINE database and contains 23,445 documents indexed by 14,321 non-overlapping categories; *re0* is the Reuters-21578 text categorization collection; *WAP* is from WebACE Project, and each document is a web page listed in one of the hierarchical categories of Yahoo. We create three new data sets for each group of real-world data set. Hence, we get six real-world data sets in total. Since the compared method SRC needs to solve eigenvectors of the relation matrix, which consumes very high memory and CPU resources, we carry out feature selection to choose the top 1000 words by mutual information for Newsgroup. Each data set mentioned above forms a heterogeneous information network described the relations among words, documents, and categories [8,17,30,32].

We also used two real-world data sets for overlapping multi-type clustering. The first one is,*3Sources*[6] which describes 948 news articles covering 416 distinct news stories. Among these stories, 169 are reported in all three sources, 194 in two sources, and 53 in a single news source. All of the stories are annotated with at least one of the six topical labels, e.g., *business*, *entertainment*, *health*, *politics*, *sport*, *technology*. We create three groups of sub-data sets denoted as T1, T2, and T3 from *3Sources* data set. The second real-world data set is the DBLP[7] data set, which contains 28,702 authors who published papers on the specified 20 conferences of computer sciences including KDD, VLDB, AAAI, etc. Each author is described by a set of words which are extracted from the titles and abstracts of these papers published by this author.

Besides, in this data set, the homogeneous relations exist between different authors, and the heterogeneous relations exist among authors, conference, and words. Since we neither know the ground truth of the size of communities (clusters) nor the number of communities (clusters) of this data set, and moreover, only our methods among all of the compared methods can capture homogeneous links, we use DBLP data set for a case study.

Except for the DBLP data set, the detailed description of the real-world data sets used in this paper is given in Table 3. We notice that the standard text preprocessing, such as stop word removal and stemming, has already been applied to all of these data sets.

6.2 Evaluation metrics

In order to evaluate our experimental results in a subjectively and comprehensive way, we use three different metrics to measure the performance of all compared methods. Assume C_1 and C_2 are two different coverings of the given network. The first metric is omega [11], which is a generalization of the well-known adjusted Rand index (ARI) [24] used for partitions of disjoint clusters. Different from ARI, omega can not only be used to measure the similarity of non-overlapping coverings, but also can be used to measure the similarity of overlapping coverings. The definition of omega is as follows.

[4] http://people.csail.mit.edu/jrennie/20Newsgroups/.

[5] http://www-users.cs.umn.edu/han/data/.

[6] http://mlg.ucd.ie/datasets.

[7] http://www.cs.uiuc.edu/homes/sun22/data/.

Table 3 Data set descriptions

Name	Data set	Category structure	♯documents	♯words	Overlap
N1	Newsgroup	{rec.sport.baseball, rec.sport.hockey}	1846	1000	NO
		{comp.os.ms-windows.misc, comp.windows.x}			
		{politics.guns, politics.mideast, politics.misc}			
N2	Newsgroup	{comp.os.ms-windows.misc, comp.windows.x}	3006	1000	NO
		{religion.christian, religion.misc}			
		{sci.med, sci.space}			
		{rec.sport.baseball, rec.sport.hockey}			
N3	Newsgroup	{comp.sys.ibm.pc.hardware, comp.sys.mac.hardware}	2355	1000	NO
		{rec.autos, rec.motorcycles}			
		{sci.crypt, sci.electronics}			
W1	oh15	{Adenosine, Aluminum, Cell-Move}	883	3987	NO
	re0	{cpi, money}			
W2	oh15	{Enzyme-Act, Staph-Inf, Cell-Move}	1432	7917	NO
	re0	{interest, trade, cpi}			
	WAP	{film, television, music}			
W3	oh15	{Staph-Inf, Enzyme-Act}	1175	7239	NO
	re0	{trade, interest}			
	WAP	{television, film}			
T1	3Source(BBC)	{entertainment} {health} {politics}	290	3523	YES
		{sport} {technology}			
T2	3Source(Reuter)	{entertainment} {sport}	122	2474	YES
T3	3Source(Reuter)	{entertainment} {health} {politics}	294	3068	YES
		{sport} {technology} {business}			

$$\begin{aligned}
\text{Omega}(C_1, C_2) &= \frac{\text{Omega}_u(C_1, C_2) - \text{Omega}_e(C_1, C_2)}{1 - \text{Omega}_e(C_1, C_2)} \\
\text{Omega}_u(C_1, C_2) &= \frac{1}{N}\sum_{j=0}^{m_{\max}} |t_j(C_1) \cap t_j(C_2)| \\
\text{Omega}_e(C_1, C_2) &= \frac{1}{N^2}\sum_{j=0}^{m_{\max}} |t_j(C_1)| \cdot |t_j(C_2)|
\end{aligned} \tag{17}$$

where Omega_u is the unadjusted omega index, and Omega_e is the expected omega index. $m_{\max} = \max(m(C_1), m(C_2))$ is the larger number of clusters existing in C_1 or C_2. $N = n(n-1)/2$ is the number of all possible pairs in n nodes, and $t_j(C_i)$ is the set of node pairs which occur j times together in a cluster of covering $C_i (i = 0, 1)$. In the special case of non-overlap covering, omega is degenerated to ARI. Notice that this metric can take negative values. When $\text{Omega}(C_1, C_2) = 1$, we have identical coverings. This metric is also used in [19,22].

The second metric used in this paper is purity. In order to compute purity, each cluster is assigned to the majority class of objects in the cluster. The metric is computed by counting the number of object assigned correctly divided by the max number of objects between coverings C_1 and C_2. In order to compare the purity between the overlapping covering and non-overlapping covering, we extend the definition of purity as follows.

$$\text{Purity}(C_1, C_2) = \frac{1}{n}\sum_i \max_j \left| C_1^i \cap C_2^j \right| \tag{18}$$

where $n = \max\{\sum_i |C_1^i|, \sum_j |C_2^j|\}$. Besides, C_1^i stands for the ith cluster of covering C_1, and C_2^j stands for the jth cluster of covering C_2. If the ground truth of the clusters have overlapping structures, both disjointed covering and incorrect overlapping covering will be penalized by purity.

The third metric is normalized mutual information (NMI), which is proposed by Lancichinetti et al. [27]:

$$\begin{aligned}
\text{NMI}(C_1, C_2) &= 1 - \frac{1}{2}\left(H(C_1|C_2)_{\text{norm}} + H(C_2|C_1)_{\text{norm}}\right) \\
H(C_1|C_2)_{\text{norm}} &= \frac{1}{|C_1|}\sum_k \frac{\min_{l \in \{1,2,\ldots,|C_2|\}} H\left(C_1^k|C_2^l\right)}{H\left(C_1^k\right)} \\
H(C_2|C_1)_{\text{norm}} &= \frac{1}{|C_2|}\sum_k \frac{\min_{l \in \{1,2,\ldots,|C_1|\}} H\left(C_2^k|C_1^l\right)}{H\left(C_2^k\right)}
\end{aligned} \tag{19}$$

where $H(C_1|C_2)$ and $H(C_2|C_1)$ are conditional entropies; $|C_1|$ and $|C_2|$ are the number of clusters in C_1 and C_2, respectively; and C_i^j means the ith cluster in covering C_j. The value of NMI ranges from 0 to 1. The higher value of NMI means the more similar between two coverings C_1 and C_2. We notice that this measurement is also commonly used in many other papers [48].

6.3 Non-overlapping multi-type clustering

We give the results of non-overlapping multi-type clustering for the synthetic data set based on Classic3 in Table 4. The number of clusters for the document is given, i.e., $l_1 = 3$. Since the

number of clusters for each type of words is available, we report all of the three metric scores, i.e., omega, purity, and NMI, under different numbers of word clusters, i.e., $l_2 = 15, 20, 25$. The same number of clusters for the two types of word objects is set, e.g., $l_2 = l_3$. From Table 4, it is clear that both LSS-H and NMF gain very high scores, while CMRF achieves the relative lowest scores on all of the three metrics under different numbers of word clusters. Furthermore, LSS-H outperforms NMF slightly on purity scores and achieves almost average 10 % higher scores on both omega and NMI than NMF. This demonstrates that, compared to the other three methods, LSS-H can detect clusters with much higher quality on Classic3.

The metric scores of different methods on data sets of N1, N2, and N3, which are extracted from Newsgroup, are given in Figs. 5, 6, and 7, respectively. In this group of experiments, three different types of objects, which are words, documents, and categories, form a heterogeneous information network. Similar to Classic3, all of the compared methods are evaluated under

Table 4 Metric scores for compared methods on Classic3 data set

Metrics	$l_2(l_3)$	NMF	SRC	CMRF	LSS-H
Omega	15	0.852	0.650	0.571	0.906
	20	0.840	0.761	0.592	0.891
	25	0.808	0.772	0.581	0.914
Purity	15	0.935	0.895	0.695	0.947
	20	0.923	0.904	0.689	0.952
	25	0.924	0.913	0.704	0.934
NMI	15	0.783	0.579	0.519	0.844
	20	0.752	0.663	0.491	0.859
	25	0.751	0.680	0.501	0.863

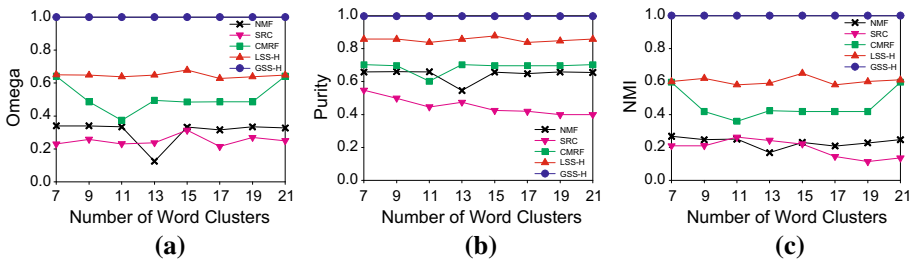


Fig. 5 Metric score comparisons between different methods on data set $N1$. **a** Omega, **b** purity, **c** NMI

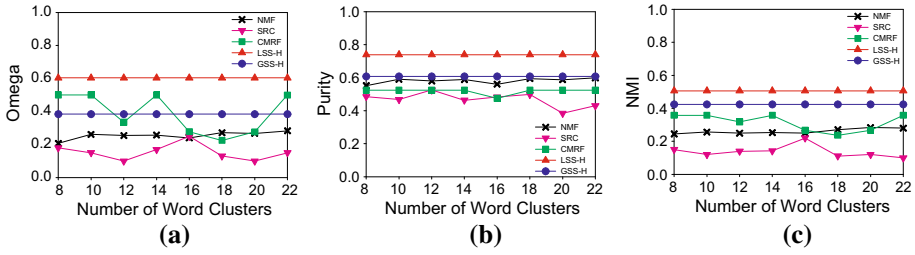


Fig. 6 Metric score comparisons between different methods on data set $N2$. **a** Omega, **b** purity, **c** NMI

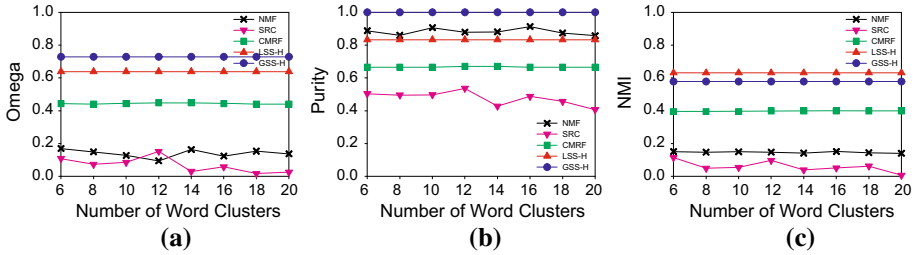


Fig. 7 Metric score comparisons between different methods on data set *N*3. **a** Omega, **b** purity, **c** NMI

different numbers of word clusters. We notice that both LSS-H and GSS-H perform very well on all three data sets generated from Newsgroup. As GSS-H is parameter-free multi-type clustering method, it is not surprising to find that the plot of GSS-H is a horizontal line in all cases. Despite without given any knowledge of the number of clusters for each type of objects, it is worth to point out that GSS-H achieves high cluster quality on data set N1. Furthermore, we notice that GSS-H gets the highest omega and purity scores on N3 data set. This indicates that GSS-H has excellent performance on searching the most appropriate number of clusters for each type of objects in the heterogeneous information network.

According to Table 3, we know that the ground truth of the number of document clusters in N1, N2, and N3 is 3, 4, and 3, respectively. In this group of experiments, GSS-H discovers 3, 5, 3 document clusters in N1, N2, and N3, respectively, which are very close to the ground truth. We carefully examine the content of clusters discovered by GSS-H in N2, and we find that GSS-H divides the documents of the third category into two clusters.

Besides, LSS-H outperforms all of the compared methods over all of the three metric scores on data set N2. From the further analysis of this group of experiments, we know that CMRF outperforms NMF on data sets of N1 and N2. SRC gains the relatively lowest scores on all of three data sets. An interesting phenomenon is that NMF achieves very high purity scores but relatively low scores on omega and NMI on data set N3. We carefully analyze the content of the clusters discovered by NMF on data set N3. We find that the detected clusters are very unbalanced. In other words, the number of documents in one cluster is much more than the rest of clusters, which greatly deviates from the ground truth of this data set. This demonstrates that, compared to most existing works [8,32] which only use one single metric for evaluating the experimental multi-type clustering results, our results are more comprehensive, since we analyze the experimental results more subjectively by using three different metrics at the same time.

The experimental results of the compared methods on data sets W1, W2, and W3 are given in Figs. 8, 9, and 10, respectively. Once again, both GSS-H and LSS-H perform very effectively. More concretely, compared to the other methods, GSS-H achieves the highest scores on all of three metrics on data set W1, while both GSS-H and LSS-H gain the average higher scores on data sets W2 and W3. We also observe that CMRF achieves a very comparative scores, which is even higher than LSS-H but lower than GSS-H, over all of three metrics on data set W1. However, the clustering quality of CMRF decreases greatly on data sets W2 and W3. Another observation is that the performance of NMF and SRC is not as stable as LSS-H, GSS-H, and CMRF. This point is especially obvious on data set W1. Besides, compared to the first group of data sets, SRC performs much better on this group of data sets, especially on data set W3.

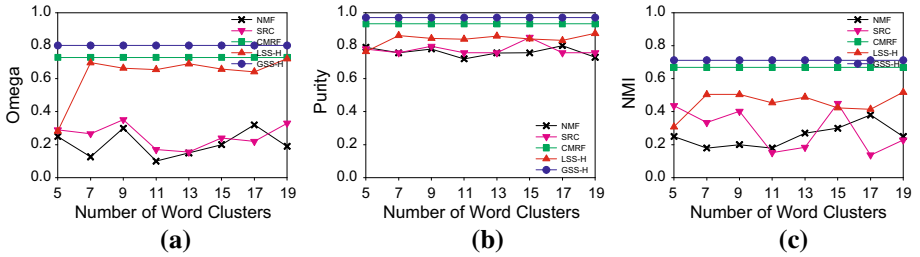


Fig. 8 Metric score comparisons between different methods on data set $W1$. **a** Omega, **b** purity, **c** NMI NMI

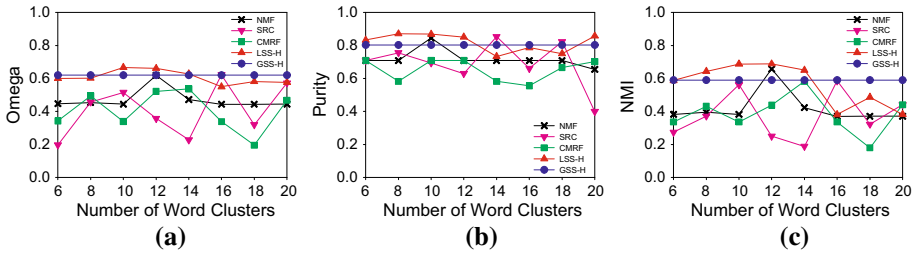


Fig. 9 Metric score comparisons between different methods on data set $W2$. **a** Omega, **b** purity, **c** NMI

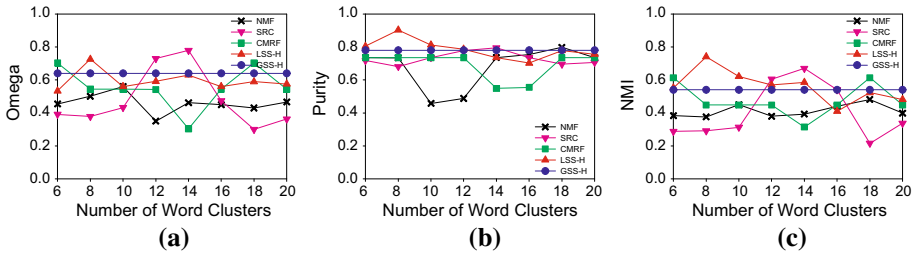


Fig. 10 Metric score comparisons between different methods on data set $W3$. **a** Omega, **b** purity, **c** NMI

6.4 Overlapping multi-type clustering

The experimental results of the compared methods on Classic3-based overlapping data sets are given in Tables 5, 6, and 7, respectively. In this synthetic data set, we test the performance of the compared methods under different numbers of word clusters and different overlapping percentages, which are controlled by parameter OV%. It is clear that OSS-H, which can discover the overlapping structure in heterogeneous information network, outperforms the other compared methods under different overlapping percentages. That is because OSS-H is able to discover the overlapping structure, but the rest of compared methods, i.e., NMF, SRC, and CMRF, is not able to detect the overlapping structure of the clusters in heterogeneous information networks. We notice that, in order to fairly compare the accuracy with each other, OSS-H calls LSS-H since the number of clusters is provided.

Another interesting observation on Classic3-based overlapping data set is that all of these three metric scores, i.e., omega, purity, and NMI, of the compared methods decrease with the increase in overlapping percentage of documents. One possible reason for this phenomenon is the increasing overlapping cluster structure makes the cluster structure become more

Table 5 Omega scores for compared methods on Classic3 with different overlap percentages

OV%	$l_2(l_3)$	NMF	SRC	CMRF	OSS-H
5	15	0.806	0.630	0.537	0.896
	20	0.810	0.731	0.532	0.895
	25	0.788	0.733	0.532	0.903
10	15	0.739	0.620	0.507	0.854
	20	0.737	0.595	0.500	0.850
	25	0.731	0.652	0.508	0.854
15	15	0.686	0.605	0.465	0.752
	20	0.687	0.499	0.467	0.763
	25	0.674	0.530	0.469	0.775
20	15	0.619	0.392	0.464	0.726
	20	0.615	0.561	0.469	0.731
	25	0.613	0.562	0.473	0.740

Table 6 Purity scores for compared methods on Classic3 with different overlap percentages

OV%	$l_2(l_3)$	NMF	SRC	CMRF	OSS-H
5	15	0.915	0.845	0.645	0.946
	20	0.913	0.884	0.647	0.943
	25	0.900	0.884	0.644	0.944
10	15	0.867	0.827	0.625	0.932
	20	0.868	0.815	0.627	0.930
	25	0.869	0.833	0.625	0.933
15	15	0.832	0.808	0.604	0.860
	20	0.836	0.760	0.605	0.866
	25	0.835	0.771	0.612	0.874
20	15	0.798	0.720	0.536	0.881
	20	0.796	0.688	0.534	0.875
	25	0.796	0.701	0.535	0.886

complicated and harder to be detected for all of the compared methods. The other possible reason is the number of hybrid document, which originally belongs to two different types of documents, is increasing with the overlapping percentage getting higher. In other words, the ground truth for the number of clusters in Classic3 is changing from three to six. However, even in the more complicated situations, e.g., OV% $= 20$, OSS-H still achieves the highest scores over all three metrics.

The three different types of metric scores of the compared methods on data sets T1, T2, and T3, which are extracted from on 3Source, are reported in Figs. 11, 12, and 13, respectively. In this group of data sets, the high word dimension and overlapping cluster structures make the task of detecting the correct cluster very challenging for all of the compared methods. However, even in this case, OSS-H still achieves comparative performance. Concretely, OSS-H achieves the average highest scores over all three metrics on data sets T1 and T2. We notice that, when the number of word cluster is small, the performance of OSS-H is not as good as when the number of word cluster is larger. Moreover, the performance changing point for OSS-H is 15 for data set T1 and six for data set T2. The reason for this phenomenon is that

Table 7 NMI scores for compared methods on Classic3 with different overlap percentages

OV%	$l_2(l_3)$	NMF	SRC	CMRF	OSS-H
5	15	0.725	0.555	0.479	0.834
	20	0.732	0.653	0.481	0.839
	25	0.718	0.660	0.488	0.859
10	15	0.660	0.557	0.434	0.787
	20	0.661	0.541	0.425	0.783
	25	0.663	0.585	0.433	0.798
15	15	0.626	0.533	0.426	0.704
	20	0.623	0.443	0.415	0.713
	25	0.614	0.483	0.400	0.724
20	15	0.563	0.543	0.392	0.684
	20	0.569	0.513	0.384	0.670
	25	0.573	0.516	0.375	0.657

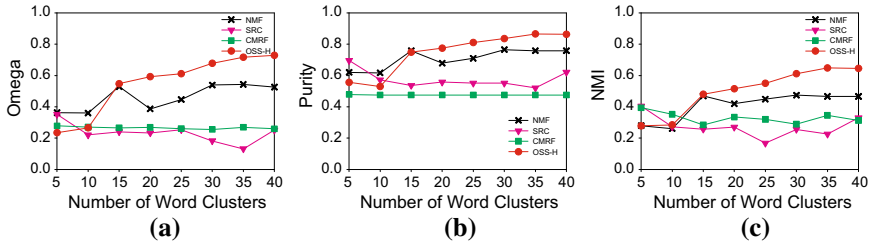


Fig. 11 Metric score comparisons between different methods on data set $T1$. **a** Omega, **b** purity, **c** NMI

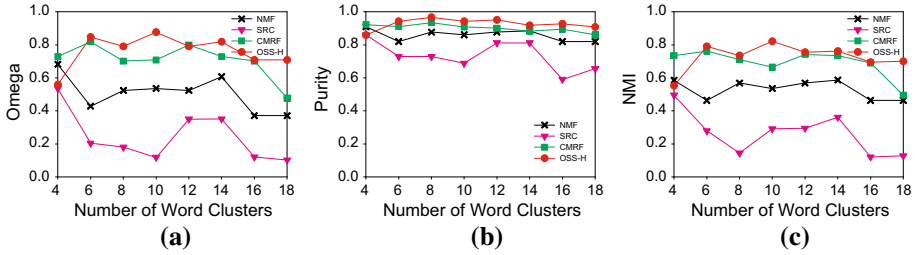


Fig. 12 Metric score comparisons between different methods on data set $T2$. **a** Omega, **b** purity, **c** NMI

our overlapping strategy for OSS-H has a close relation with the number of clusters, and the appropriate number of clusters provides higher accuracy information for detecting the overlapping cluster structures. We also notice that NMF does poorly on T2, while CRFM performs badly on T1.

Since all of the six types of documents in Reuter data source are covered in T3, whose overlapping cluster structure is even harder to detect than T1 and T2, GSS-H is called in OSS-H. We notice that, in data set T3, neither the number of document clusters nor the number of word clusters is provided, OSS-H still outperforms the rest of other compared methods. This further demonstrates the effectiveness of OSS-H for discovering the overlapping cluster structures.

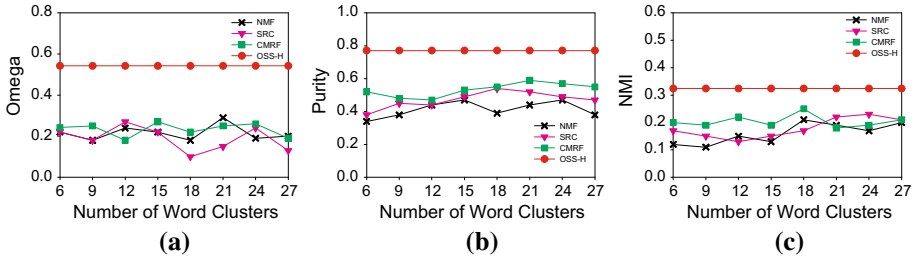


Fig. 13 Metric score comparisons between different methods on data set $T3$. **a** Omega, **b** purity, **c** NMI

Table 8 Most overlapping authors appear in 17 different author clusters

Alon Halevy, Amr El Abbadi, Beng Chin Ooi, Baruch Awerbuch, Christos Faloutsos, C. Lee Giles, David J. DeWitt, Divesh Srivastava, Divyakant Agrawalm, D. R. Karger, Dimitris Papadias, Gerhard Weikum, George Varghese, Hari Balakrishnan, Hans-Peter Kriegel, Hector Garcia-Molina, Heikki Mannila, H. V. Jagadish, Ion Stoica, J. D. Ullman, Jennifer Widom, Jeffrey F. Naughton, Jiawei Han, Jian Pei, Joseph M. Hellerstein, Jon Kleinberg, Johannes Gehrke, Krithi Ramamritham, Minos N. Garofalakis, Michael J. Carey, Michael J. Franklin, Nick Koudas, Phillip B. Gibbons, Philip Yu, Piotr Indyk, Prabhakar Raghavan, Rajeev Motwani, Rakesh Agrawal, Ravi Kumar, Rajeev Rastogi, Raghu Ramakrishnan, Scott Shenker, Serge Abiteboul, S. Muthukrishnan, Surajit Chaudhuri, Vipin Kumar, W. Bruce Croft, Wei-Ying Ma, Wei Wang, Yossi Azar, . . . (total: 154)

7 Case studies

However, only OSS-H can automatically discover the overlapping clustering structures in the heterogeneous information networks with homogeneous links and heterogeneous information links. We conduct case studies on DBLP data set, where the homogeneous relations exist among authors and the heterogeneous relations exist among authors, words, and conferences.

In this data set, OSS-H discovers four clusters in conference field, 12 clusters in words field, and 17 clusters in author field, respectively. We notice that the number of clusters in the conference field detected by OSS-H is exactly the same with the number of research fields where the 20 conferences come from. This demonstrates that OSS-H performs very well on clustering conference field. We also carefully analyze the contents of the author clusters and word clusters. We assign the IDs ranges from one to 17 to the different author clusters. We note that the authors in clusters one, two, and three are mainly interested in database field, authors in clusters 4, 7, and 11 are mainly focusing on artificial intelligence field, and authors in clusters 5, 6, 8, and 10 are mainly from machine learning field, while the rest of author clusters are mainly active in information retrieval. We observe that many authors appear in more than one author clusters. This illustrates that many authors published papers in more than one research area. Usually, the more clusters the author appears in, the wider research fields and collaborating fields the author has. According to the H-index value,[8] we list the top 50 among all of the 154 most overlapped authors in Table 8. We know that H-index is always used to present the achievement of a researcher. Table 8 reflects the phenomenon that the active and fruitful authors always have relatively extensive cooperation network and wide research interest, which fits for the fact of the real world.

[8] http://arnetminer.org/.

Table 9 Shared words and exclusive words in column cluster 2 and column cluster 11

Media, structure, models, graph, detection, topic, predict, hash, cluster, matches, attribute, spaces, experimentally, online, supervised, pruning, system, feature, label Gaussian, probabilistic, statistical, semi-supervised, . . .	Web, information, semantic, XML, retrieval, engine, search, language, sentence, query, extract, feedback, bootstrap, negative, rank, name, entity, keyword, . . .
	Community, mining, social, pattern, streams, outliers, uncertainty, classify, path, flow, influence, evolution, workflow, association, itemset, networks, . . .

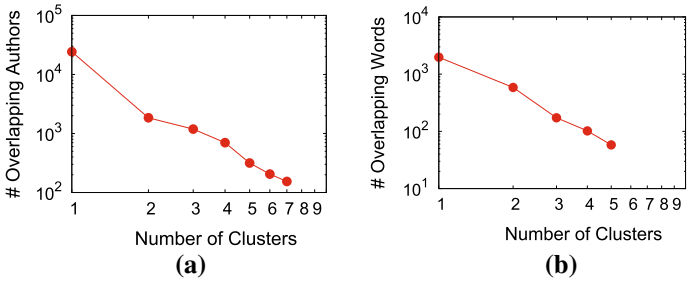


Fig. 14 Distribution of overlapping authors/words on DBLP data set. **a** Overlapping authors distribution, **b** overlapping words distribution

In order to further verify the quality of the word clusters, in Table 9, we give the words only appear in word cluster 2 in the first row of right column and the words only appear in cluster 11 in the second row of right column, while the words appear in both word clusters 2 and 11 are placed in the left column of Table 9. We find the words contained in word cluster 2 are more frequently used by authors from information retrieval field, while the words in cluster 11 are often used by authors from data mining field. Moreover, the set of words shared by both word cluster 2 and word cluster 11 are more frequently used by authors from both information retrieval and data mining areas.

In Fig. 14, the distribution of the number of overlapping authors is given. For a specified point (x, y) in Fig. 14a, it means y authors appear in x different author clusters. An interesting phenomenon is observed that the distribution of the number of overlapping authors and the number of clusters obey *power-law distribution*. As shown in Fig. 14b, a similar phenomenon is also observed in the distribution of overlapping words. This further illustrates that most of the authors focus on one research field, and most of the words have very strong context.

8 Running time comparison

We conduct running time comparison on DBLP data set for its large size. Since NMF, SRC, and CMRF cannot capture the homogeneous relations in DBLP data set, we only extract the heterogeneous links, which are the links among authors, conferences, and words, from DBLP data set for running time comparison. Moreover, for NMF, SRC, CMRF, and LSS-H, we feed the number of clusters of each type discovered by GSS-H directly.

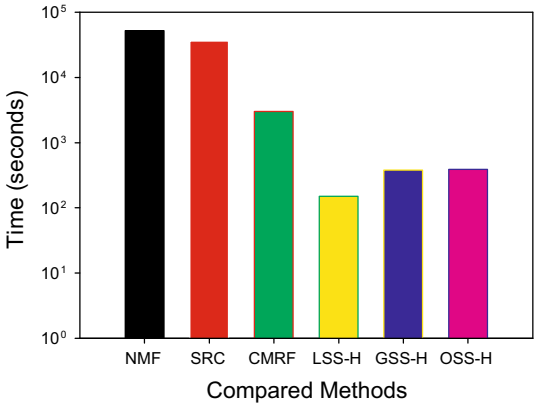


Fig. 15 Running time comparison between different methods on DBLP data set

The running time for different methods is shown in Fig. 15. Not surprisingly, NMF, SRC, and CMRF cost much more time than our proposed methods, which are LSS-H, GSS-H, and OSS-H. That is because, on the one hand, the time complexity of NMF is $O(ITkN_cN_p)$, SRC is $O(IT(\max(N_c, N_p)^3) + kN_cN_p)$, and CMRF is $O(IT(\max(N_c^3, N_p^3)))$, where I is the number of iterations, N is the number of data types, $k = \max(k_c, k_p)$ is the maximum number of clusters in all data types, and N_c is the number of features in the central data type, and N_p is the maximum feature dimension among all different types of objects in the heterogeneous information network [6,8,32]. On the other hand, as we analyzed in Sect. 5, the time complexity of LSS-H, GSS-H, and OSS-H is linear to the number of links in the heterogeneous information network. Hence, our methods are more appropriate for large data sets. Even though GSS-H is based on LSS-H and need to invoke LSS-H multiple times for automatically discovering the optimal number of clusters of each type, the running time of GSS-H is still much less than that of NMF, SRC, and CMRF. We also notice that SRC needs to solve the eigenvectors of the relational matrices which is very memory intensive, so it is hard to be used for the large-scale data sets. From these results, we know that our proposed methods provide a much more efficient way for clustering heterogeneous information networks than existing state-of-the-art methods.

9 Conclusion

Multi-type clustering on heterogeneous information networks is critical and significant for discovering the cluster structures of the heterogeneous information networks. In this paper, we proposed a general model for the heterogeneous information network that can be parameter-free and support overlapping multi-type clustering. In this model, both the homogeneous relations and heterogeneous relations are considered simultaneously. By condensing the group of relation matrices, which are used to describe the relations of different types of objects, we transfered the multi-type clustering problem into the information compressing problem. Then, two greedy search algorithms, i.e., LSS-H and GSS-H, were devised. LSS-H can discover different types of clusters with the knowledge of the number of clusters in each type of objects in the heterogeneous information network, while GSS-H invokes LSS-H many times and can discover the number of clusters automatically in each type of objects when detecting the cluster structures of the heterogeneous information network. By distinguishing the discriminative clusters of the heterogeneous information network,

a novel overlapping multi-type clustering algorithm OSS-H was proposed. Experimental results demonstrated our proposed methods outperform state-of-the-art methods for multi-type clustering on heterogeneous information networks.

Acknowledgments Wangqun Lin and Bo Deng are supported by National Natural Science Foundation of China through Grant 61271252. Philip S. Yu and Yuchen Zhao are supported by NSF through Grant CNS-1115234, Google Research Award, and the Pinnacle Lab at Singapore Management University.

References

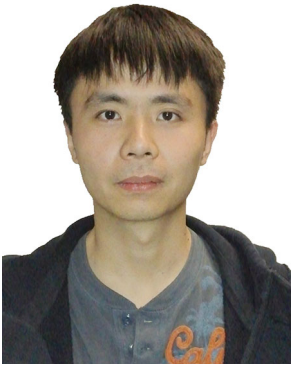
1. Ahn YY, Bagrow JP, Lehmann S (2010) Link communities reveal multiscale complexity in networks. Nature 466:761–764
2. Banerjee A, Basu S, Merugu S (2007) Multi-way clustering on relation graphs. Proceedings of the 7th SIAM international conference on data mining. SIAM, Minneapolis, MN, USA, pp 145–156
3. Banerjee A, Dhillon I, Ghosh J, Merugu S, Modha DS (2007) A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. J Mach Learn Res 8:1919–1986
4. Barron A, Rissanen J, Yu B (1998) The minimum description length principle in coding and modeling. IEEE Trans Inf Theory 44(6):2743–2760
5. Bekkerman R, Mccallum A (2005) Multi-way distributional clustering via pairwise interactions. Proceedings of the 22nd international conference on machine learning. ACM, Bonn, pp 41–48
6. Bekkerman R, Jeon J (2007) Multi-modal clustering for multimedia collections. Computer society conference on computer vision and pattern recognition. IEEE Computer Society, Minneapolis, MN, USA, pp 1–8
7. Chakrabarti D, Papadimitriou S, Modha DS, Faloutsos C (2004) Fully automatic cross-associations. Proceedings of the 10th international conference on knowledge discovery and data mining. ACM, Seattle, Washington, DC, USA, pp 79–88
8. Chen Y, Wang L, Dong M (2010) Non-negative matrix factorization for semisupervised heterogeneous data coclustering. IEEE Trans Knowl Data Eng 22(10):1459–1474
9. Cheng YZ, Church GM (2000) Biclustering of expression data. International conference on intelligent systems for molecular biology 8:93–103
10. Cho H, Dhillon IS, Guan YQ, Sra S (2004) Minimum sum-squared residue co-clustering of gene expression data. Proceedings of the 4th international conference on data mining. SIAM, Lake Buena Vista, FL, USA, pp 114–125
11. Collins LM, Dent CM (1998) Omega: a general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. Multivar Behav Res 23(2):231–242
12. Cook DJ, Holder LB (1994) Substructure discovery using minimum description length and background knowledge. J Artif Intell Res 1:231–255
13. Dhillon IS (2001) Co-clustering documents and words using bipartite spectral graph partitioning. Proceedings of the 7th international conference on knowledge discovery and data mining. ACM, San Francisco, CA, USA, pp 269–274
14. Dhillon IS, Guan YQ (2003) Information theoretic clustering of sparse co-occurrence data. Proceedings of the 9th international conference on knowledge discovery and data mining. IEEE Computer Society, Melbourne, FL, USA, pp 517–528
15. Dhillon IS, Mallela S, Modha DS (2003) Information theoretic co-clustering. Proceedings of the 9th international conference on knowledge discovery and data mining. ACM, Washington DC, pp 89–98
16. Gao B, Liu TY, Zheng X, Cheng QS, Ma WY (2005) Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. Proceedings of the 11th international conference on knowledge discovery and data mining. ACM, New York, NY, USA, pp 41–50
17. Gao B, Liu TY, Ma WY (2006) Star-structured high-order heterogeneous data co-clustering based on consistent information theory. 6th international conference on data mining. IEEE Computer Society, Hong Kong, pp 880–884
18. Gossen T, Kotzyba M, Nürnberger A (2014) Graph clusterings with overlaps: adapted quality indices and a generation model. Neurocomputing 123:13–22
19. Gregory S (2009) Finding overlapping communities using disjoint community detection algorithms. In: Results of the 2009 international workshop on complex networks, Catania, pp 47–61
20. Guimerá R, Amaral LAN (2005) Functional cartography of complex metabolic networks. Nature 433(7028):895–900

21. Han EH, Karypis G (2000) Centroid-based document classification: analysis and experimental results. Proceedings of the 4th European conference on principles of data mining and knowledge discovery. Springer, Lyon, pp 424–431
22. Havemann F, Heinz M, Struck A, Gläser J (2011) Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. J Stat Mech Theory Exp 01:P01023
23. He JR, Tong H, Papadimitriou S, Rad TE, Faloutsos C, Carbonell J (2009) Pack: scalable parameter-free clustering on k-partite graphs. In: SDM workshop on link analysis. SIAM, John Ascuagas Nugget
24. Hubert L, Arabie P (1985) Comparing partitions. J Classif 1:193–218
25. Ienco D, Robardet C, Pensa R, Meo R (2013) Parameter-less co-clustering for star-structured heterogeneous data. Data Min Knowl Discov 26(2):217–254
26. Koutra D, Kang U, Vreeken J, Faloutsos C (2014) VOG: summarizing and understanding large graphs. Proceedings of the 2014 international conference on data mining. SIAM, Philadelphia, PA, USA, pp 91–99
27. Lancichinetti A, Fortunato S, Kertesz J (2009) Detecting the overlapping and hierarchical community structure in complex networks. New J Phys 11(3):033015
28. Lazzeroni L, Owen A (2000) Plaid models for gene expression data. Stat Sin 12:61–86
29. Lin WQ, Zhao YC, Yu PS, Deng B (2014) An effective approach on overlapping structures discovery for co-clustering. 16th Asia-Pacific web conference in web technologies and applications. Springer, Changsha, pp 56–67
30. Long B, Zhang ZF, Yu PS (2010) A general framework for relation graph clustering. Knowl Inf Syst 24:393–413
31. Long B, Wu YX, Zhang ZF, Yu PS (2006) Unsupervised learning on k-partite graphs. Proceedings of the 12th international conference on knowledge discovery and data mining. ACM, New York, NY, USA, pp 317–326
32. Long B, Zhang ZF, Wu XY, Yu PS (2006) Spectral clustering for multi-type relational data. Proceedings of the 23rd international conference on machine learning. ACM, Apia, pp 585–592
33. Long B, Zhang ZF, Yu PS (2005) Co-clustering by block value decomposition. Proceedings of the 11th international conference on knowledge discovery and data mining. IEEE Computer Society, Binghamton, pp 635–640
34. Meo PD, Ferrara E, Fiumara G, Provetti A (2014) Mixing local and global information for community detection in large networks. J Comput Syst Sci 80(1):72–87
35. Newman M, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69:026113
36. Palla G, Derenyi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435:814
37. Papadimitriou S, Gionis A, Tsaparas P, Vaisanen RA, Mannila H, Faloutsos C (2005) Parameter-free spatial data mining using MDL. Proceedings of the 5th international conference on data mining. IEEE Computer Society, Houston, TX, USA, pp 346–353
38. Papadimitriou S, Sun J, Faloutsos C, Yu PS (2008) Hierarchical, parameter-free community discovery. European conference in machine learning and knowledge discovery in databases. Springer, Antwerp, Belgium, pp 170–187
39. Rosvall M, Bergstrom CT (2007) An information-theoretic framework for resolving community structure in complex networks. Proc Natl Acad Sci USA 104:7327–7331
40. Sales MP, Guimerà R, Moreira A, Amaral L (2007) Extracting the hierarchical organization of complex systems. Proc Natl Acad Sci 104(39):15224–15229
41. Shiga M, Takigawa I, Mamitsuka H (2007) A spectral clustering approach to optimally combining numerical vectors with a modular network. Proceedings of the 13th international conference on knowledge discovery and data mining. ACM, San Jose, CA, USA, pp 647–656
42. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
43. Sun YS, Yu YT, Han HW (2009) Ranking-based clustering of heterogeneous information networks with star network schema. Proceedings of the 15th international conference on knowledge discovery and data mining. ACM, New York, NY, USA, pp 797–806
44. Tian Y, Hankins R, Patel J (2008) Efficient aggregation for graph summarization. Proceedings of the international conference on management of data (SIGMOD 2008). ACM, Vancouver, pp 567–580
45. Tsai C, Chiu C (2008) Developing a feature weight self-adjustment mechanism for a k-means clustering algorithm. Comput Stat Data Anal 52:4658–4672
46. Wakita K, Tsurumi T (2007) Finding community structure in mega-scale social networks. Proceedings of the 16th international conference on world wide web. ACM, Banff, AB, Canada, pp 1275–1276

47. Wang JD, Zeng HJ, Chen Z, Lu HJ, Tao L, Ma WY (2003) Recom:reinforcement clustering of multi-type interrelated data objects. Proceedings of the 26th annual international conference on research and development in information retrieval. ACM, New York, NY, USA, pp 274–281
48. Wang XF, Tang L, Gao HJ, Liu H (2010) Discovering overlapping groups in social media. 10th international conference on data mining. IEEE Computer Society, Sydney, pp 569–578
49. Xu X, Yuruk N, Feng Z, Schweiger TAJ (2007) Scan: A structural clustering algorithm for networks. Proceedings of the 13th international conference on knowledge discovery and data mining. ACM, San Jose, CA, USA, pp 824–833

Wangqun Lin is currently an Assistant Professor in Beijing Institute of System Engineering. He received his M.Sc. and Ph.D. from National University of Defense Technology in 2008 and 2012, respectively. From 2010 to 2012, he was a visiting scholar in Information Technology at the Department of Computer Science, University of Illinois at Chicago. His research interests include data mining, database, and software engineering.

Philip S. Yu is a Distinguished Professor and the Wexler Chair in Information Technology at the Department of Computer Science, University of Illinois at Chicago. Before joining UIC, he was at the IBM Watson Research Center, where he built a world-renowned data mining and database department. He is a Fellow of ACM and IEEE. Dr. Yu is the recipient of IEEE Computer Society's 2013 Technical Achievement Award for "pioneering and fundamentally innovative contributions to the scalable indexing, querying, searching, mining and anonymization of big data". With more than 870 publications and 300 patents, cited more than 62,000 times with an H-index of 116, Dr. Yu is a leader in the data mining and data management community. Dr. Yu is the Editor-in-Chiefof ACM Transactions on Knowledge Discovery from Data. He is on the steering committee of the IEEE Conference on Data Mining and ACM Conference on Information and Knowledge Management and was a member of the IEEE Data Engineering steering committee. He was the Editor-in-Chief of IEEE Transactions on Knowledge and Data Engineering (2001–2004). He received a Research Contributions Award from IEEE Intl. Conference on Data Mining (ICDM) in 2003, the ICDM 2013 10-year Highest-Impact Paper Award, and the EDBT Test of Time Award (2014). Dr. Yu received his Ph.D. from Stanford University.



Yuchen Zhao received his Ph.D. degree from the University of Illinois at Chicago in data mining and machine learning. He has been actively conducting data science-related research both in academia and industry. He has published over 10 referred papers in conferences and journals, and a number of patents have been filed based on his research findings. He is also serving as program committee member on top data science conferences including ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), International Joint Conferences on Artificial Intelligence (IJCAI), and ACM International Conference on Information and Knowledge Management (CIKM).

Bo Deng is currently a Professor in Beijing Institute of System and Engineering. He received his Ph.D. from National University of Defense Technology in 2000. His research interests include computer architecture, software engineering, and data mining.