International Journal of Machine Learning and Cybernetics (2021) 12:2887–2898
https://doi.org/10.1007/s13042-021-01375-9

ORIGINAL ARTICLE

Hierarchical high-order co-clustering algorithm by maximizing modularity

Jiahui Wei[1] · Huifang Ma[1,2,3] · Yuhang Liu[1] · Zhixin Li[3] · Ning Li[4]

Received: 28 January 2020 / Accepted: 5 July 2021 / Published online: 23 July 2021

Abstract
The star-structured high-order heterogeneous data is ubiquitous, such data represent objects of a certain type, connected to other types of data, or the features, so that the overall data schema forms a star-structure of inter-relationships. In this paper, we study the problem of co-clustering of star-structured high-order heterogeneous data. We present a new solution, a Hierarchical High-order Co-clustering Algorithm by Maximizing Modularity, MHCoC, which iteratively optimizes the objective function based on modularity and finally converges to a unique clustering result. In contrast to the traditional co-clustering methods, MHCoC merges information of multiple feature spaces of high-order heterogeneous data. Moreover, MHCoC takes a top-down strategy to perform a greedy divisive procedure, generating a tree-like hierarchical clustering result that reveal the relationship between clusters. To illustrate the process in more detail, we design a toy example to describe how MHCoC selects the appropriate co-cluster and splits it. Extensive experiments on real-world datasets demonstrate the effectiveness of the proposed method.

Keywords Co-clustering · Modularity · High-order heterogeneous data · Hierarchical structure

1 Introduction

In the era of big data, there is more than ever a need for techniques that simultaneously group objects and features of data, thus making large data sets easier to handle and interpret. Co-clustering techniques just serve this purpose [1]. Several co-clustering methods have been proposed in the past decades and applied to many fields, such as collaborative filtering recommendation [2, 3], biological science [4, 5], multimedia data analysis [6, 7], and natural language processing [1, 8].

With the rapid development of data science, an increasing amount of high-order heterogeneous data containing multiple types of object and involving the relationship between multiple types of object has been generated [9]. These high-order heterogeneous data provide useful information in comparison with homogeneous data [10], properly combining information from multiple types of object data would help improve the clustering performance [11], Hence, it is essential to develop high-order co-clustering models to solve the high-order heterogeneous data co-clustering problems [12]. Researchers have developed many high-order co-clustering methods based on different theories, such as graph partitioning [13], information theory [14], matrix factorization [15] and so on.

Although these methods are different in the technique employed on co-clustering high-order heterogeneous data, they all simultaneously generate the flat partitioning of every types of object data, thus the relationship between the clusters cannot be captured. Motivated by the above observations, this paper proposes a Hierarchical High-order Co-Clustering algorithm by maximizing Modularity (MHCoC). MHCoC chooses the modularity in the co-clustering context as a quality evaluation indicator, and an efficient iterative algorithm is designed to optimize the objective function. In

✉ Huifang Ma
mahuifang@yeah.net

1 College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, Gansu, China

2 Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, Guangxi, China

3 Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, Guangxi, China

4 Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

each iteration, MHCoC uses a top-down strategy for splitting co-cluster, and finally the tree-like clustering result of each types of object data is obtained. Experiments show that the method is effective and superior to other similar methods.

The main contributions of this paper are summarized as follows:

(1). We present a modularity-based co-clustering objective function, which effectively merges information of high-order heterogeneous data.

(2). An alternating iterative method is designed to optimize the objective function in an interplay manner, which simultaneously generates tree-like clustering result of each types of object data.

(3). Extensive experiments have been conducted on several real-world datasets, demonstrating that the proposed algorithm outperforms the existing high-order co-clustering algorithms.

The remainder of this paper is organized as follows: In Sect. 2, we review related work. In Sect. 3, we briefly provide the modularity in co-clustering context and presents co-clustering on star-structured high-order heterogeneous data. The technical details of our approach are provided in Sect. 4. In Sect. 5, we present the results with a comprehensive set of experiments and analyzed them. Finally, Sect. 6 concludes the paper.

2 Related works

Before presenting our proposed method, we review the background of our proposed method, including co-clustering and high-order clustering, followed by hierarchical clustering. Here we mainly discuss these approaches that are most relevant to our work, and then briefly explore other techniques used in our work.

2.1 Co-clustering

Co-clustering, also called bi-clustering or block clustering, aims to cluster both rows and columns of a data matrix simultaneously. Compared with the traditional one-side clustering, co-clustering algorithms show more powerful performance [16] and they have become one of the important methods to solve the problem of high-dimensional sparse data clustering [17]. Co-clustering could be used for an extensive range of applications [18]. Some notable works are summarized as follows: Dhillion [19] poses the clustering problem as a graph partitioning problem and gives a new spectral algorithm based on normalized cut that, uses the second left and right singular vectors of an appropriately scaled word-document matrix to yield good bi-partitioning. Another notable co-clustering algorithm is the block diagonal clustering algorithm described in [20]. This algorithm produces a block diagonal matrix of the given binary document-term matrix by minimizing the squared error between the original data and its approximation. More recently, another attempt is presented to use network-related criteria in the field of block diagonal structure is the spectral co-clustering maximizing a generalization of the modularity [21]. Compared to clustering methods based on other theories like nonnegative matrix factorization or tri-factorization, this algorithm appears to perform better. Despite similar algorithms are trying to optimize criteria initially used in the studies of networks, they do so by using a spectral relaxation of the discrete optimization problem. Ailem et al. [1] develop an algorithm that allows getting even better co-clusters by directly maximizing modularity using an iterative alternating optimization procedure.

Inspired by above work and in order to take full advantage of the modularity co-clustering for high-order heterogeneous data, we develop our approach by extending the modularity to the task of high-order clustering.

2.2 High-order co-clustering

Although many clustering methods have been developed, yet they cannot be directly applied to cluster high-order heterogeneous data. Properly combining information from high-order heterogeneous data would help improve the learning performance, which leads to the emergence of the high-order clustering technique [11]. It has drawn a large amount of attention in different research fields, such as recommender system [22], social network mining [23], natural language processing [24], biomedicine [25] and multimedia analysis [26]. In recent years, high-order clustering has been studied extensively, researchers have designed many high-order co-clustering methods based on different theories. The graph partitioning based methods [27] take the features of samples and different feature spaces as nodes and consider the data values as edges to find the clustering problem of optimal graph partitioning. The information theory-based methods [28] perform clustering by maximizing the mutual information of object cluster and feature cluster random variables. For the authoritative-based clustering algorithms [29], the authoritative ordering of samples and features is obtained and applied to clustering. The correlation measure-based methods [9] uses the predictive ability to estimate the correlation between objects and features, and partitions the objects and features with strong correlation into a cluster. The fuzzy-based methods are used to maximize the degree of aggregation inside the cluster, and the membership degree of the object and feature is solved by iterative method [30]. The matrix factorization-based methods reduce each high dimensional feature space to a low dimensional space [31] for clustering.

Their approach, although using different theories, does not involve network-related measures. Also, they all generate the flat partitioning of every types of object data, while our method is a hierarchical high-order co-clustering method.

2.3 Hierarchical clustering

Top-down hierarchical clustering is a recursive partitioning of a dataset into successively smaller clusters [32]. It does not require a fixed number of clusters and provides richer information at all levels of granularity, simultaneously displayed in an intuitive form, importantly, there are many fast and easy to implement algorithms commonly used in practice to find a tree-shaped cluster structure [33]. Several hierarchical clustering methods have been proposed and applied to many fields, include image and text classification [34], Anomaly detection [35], bioinformatics [36], person re-identification [37] and more. In summary, a taxonomy structure can be more beneficial than a flat partition for many real applications [8].

In order to obtain tree-like hierarchical clustering results and reveal the relationship between clusters, MHCoC adopts a top-down strategy to perform a greedy splitting process.

2.4 Summary

By integrating co-clustering and high-order clustering and hierarchical clustering, this paper proposes a Hierarchical High-order Co-Clustering algorithm by maximizing Modularity (MHCoC). MHCoC can merges information of high-order heterogeneous data and then produce a tree-like partition for each types of object data.

3 Preliminaries

In this section, we first review the modularity in co-clustering context and then present the star-structured high-order heterogeneous data.

3.1 Modularity for co-clustering

Modularity has been widely used in many fields and has shown outstanding effects. Modularity is a commonly used to measure the quality of community detection, and it can also be used as a quality evaluation in graph clustering. By rewriting the formula, the modularity can be used as a quality measure for block diagonal co-clustering, which means that objects and features have the same number of clusters [1]. In this way, while obtaining the clustering results of the objects, the feature clusters that indicate them can be discovered.

Let $\mathbf{A} \in \mathrm{R}^{m\times n}$ be the two dimensional contingency table or co-occurrence matrix, where rows and columns are represented by object set $O=\{o_1,o_2,\dots,o_i,\dots,o_m\}$ and feature set $F=\{f_1,f_2,\dots,f_j,\dots,f_n\}$, respectively. The co-clustering problem is to partition object set O and feature set F into k object clusters $\{r_1,r_2,\dots,r_l,\dots,r_k\}$ and k feature clusters $\{c_1,c_2,\dots,c_l,\dots,c_k\}$ simultaneously ($k \ll min\{m,n\}$ can be greater or equal to 2). For this, we define the object index matrix $\mathbf{U} \in \{0,1\}^{m\times k}$, with each entry as follows: $u_{il}=1$ if the object i belongs to the object cluster l and $u_{il}=0$ otherwise. Feature index matrix $\mathbf{V} \in \{0,1\}^{n\times k}$ is defined similar with $\mathbf{U}$. The relationship matrix is defined as $\mathbf{C}=\mathbf{U}\mathbf{V}^{t}$, with its the entry $c_{il}=1$ when object i and feature j belong to object cluster and feature cluster having same cluster sequence number, otherwise $c_{il}=0$, ie.$c_{ij}=\sum_{l=1}^{k} u_{il}v_{jl}$. Then the modularity can be formulated as follows in co-clustering context:

$$Q(\mathbf{A},\mathbf{C}) = Q(\mathbf{A},\mathbf{U}\mathbf{V}^{\mathrm{T}}) = \frac{1}{a_{..}} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l}^{k} \left(a_{ij} - \frac{a_{i.}a_{.j}}{a_{..}} \right) u_{il} v_{jl} \tag{1}$$

where $a_{..}=\sum_{i,j} a_{ij}$, $a_{i.}=\sum_{j} a_{ij}$ and $a_{.j}=\sum_{i} a_{ij}$. For convenience, we define the matrix $\mathbf{X} \in \mathrm{R}^{m\times n}$ with elements $x_{ij}=\frac{a_{i.}a_{.j}}{a_{..}}$, the modularity can also be expressed in the following form:

$$Q(\mathbf{A},\mathbf{C}) = \frac{1}{a_{..}} Trace\left[(\mathbf{A}-\mathbf{X})^{\mathrm{T}}\mathbf{U}\mathbf{V}^{\mathrm{T}}\right] \tag{2}$$

3.2 Star-structured high-order heterogeneous data

High-order heterogeneous data is a data structure containing multiple types of object data. In this structure, there is a central type of object that connects the other types of objects. We take academic thesis systems as an example, where paper is the central data type while other data types can be seen as feature spaces used to represent paper. In order to identify a certain group of authors who usually write papers of a certain series of topics with a certain list of terms, and submit them to a certain kind of conferences [14].

Figure 1 shows the high-order heterogeneous data $A=\cup\mathbf{A}^{(s)}(s=1,2,\dots,N)$, which is composed of the central object set $O=\{o_1,o_2,\dots,o_m\}$ with its multiple feature spaces $F^{(1)}=\left\{f_1^{(1)},f_2^{(1)},\dots,f_{n^{(1)}}^{(1)}\right\}$, $F^{(2)}=\left\{f_1^{(2)},f_2^{(2)},\dots,f_{n^{(2)}}^{(2)}\right\}$ $,\dots,F^{(N)}=\left\{f_1^{(N)},f_2^{(N)},\dots,f_{n^{(N)}}^{(N)}\right\}$, where $f_j^{(s)}$ is the j-th feature in the s-th feature space $F^{(s)}$, and $a_{ij}^{(s)}$ is the feature value of the object o_i expressed on the feature $f_j^{(s)}$.

3.3 Problem definition

The goal of the MHCoC algorithm is to acquire the hierarchical clustering result $R=\{R_1,R_2,\dots,R_h,\dots R_H\}$ of the central data O and hierarchical clustering results $C^{(s)}=\left\{C_1^{(s)},C_2^{(s)},\dots,C_h^{(s)},\dots C_H^{(s)}\right\}(s=1,2,\dots,N)$ of feature spaces $F^{(s)}(s=1,2,\dots,N)$. R_h is the h-th time split clustering result of O, which is a set with h sample clusters, that is,

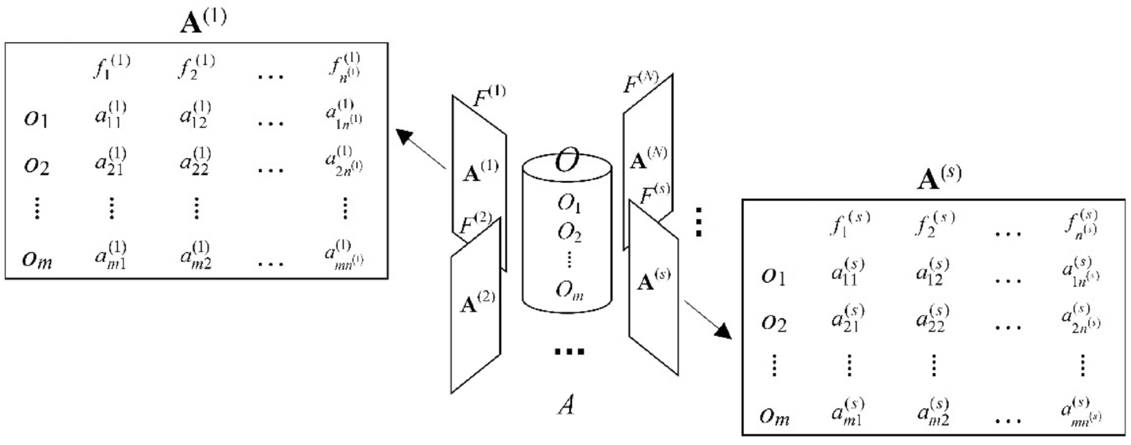


Fig. 1 Star-structure high-order heterogeneous data

$R_h = \{r_{h1}, r_{h2} \ldots, r_{hh}\}$, and $C_h^{(s)} = \left\{c_{h1}^{(s)}, c_{h2}^{(s)}, \ldots, c_{hh}^{(s)}\right\}$ is the h-th time split clustering result of $F^{(s)}$. The co-cluster formed by r_{hl} and $c_{hl}^{(s)}$ is denoted as $rc_{hl}^{(s)}$, and $RC_{hl} = \{rc_{hl}^{(1)}, rc_{hl}^{(2)}, \ldots, rc_{hl}^{(N)}\}$ is the set of $rc_{hl}^{(s)}$ in each feature space.

4 MHCoC algorithm

In this section, we first propose the modularity-based objective function and then develop a greedy divisive algorithm to optimize the objective function.

4.1 Objective function of MHCoC

MHCoC algorithm chooses the modularity used for the co-clustering context as the quality measurement. However, if we conduct co-clustering on every feature space independently, it will have a great probability that the clustering result for central type of data is different in every co-clustering result. In other words, every single clustering result of central type of data do not match in most cases. Actually, we are looking for a series of co-clustering results that each of them is not locally optimal, but their clustering results on the central type are the same, and the overall partitioning is globally optimal under a certain objective function. Therefore, the modularity-based objective function of MHCoC is defined as follows:

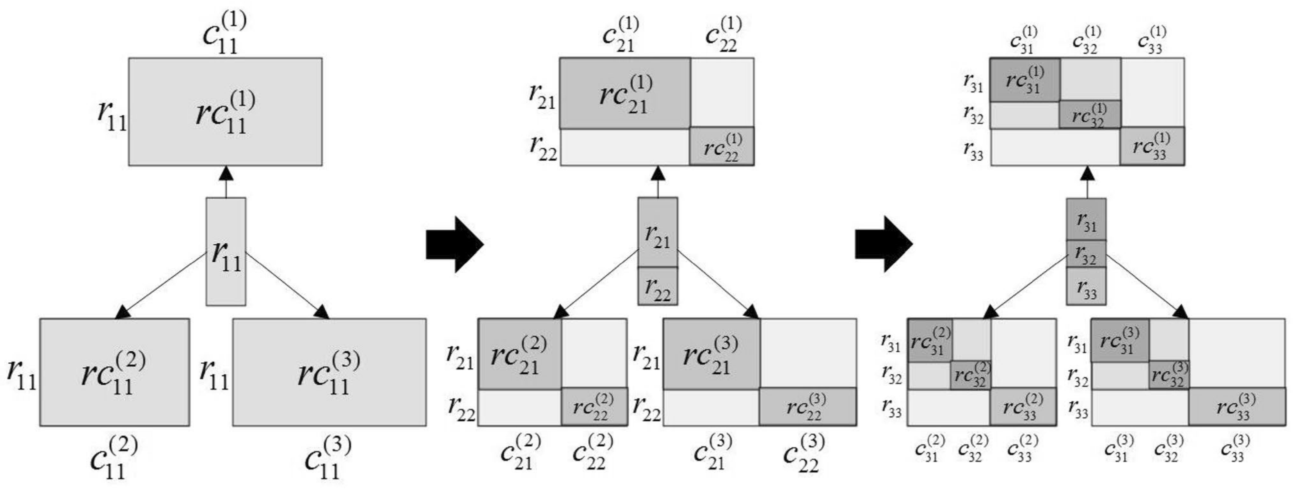
$$Q(A, C) = \sum_s Q(\mathbf{A}^{(s)}, \mathbf{U}\mathbf{V}^{(s)\mathrm{T}})$$
$$= \sum_{s=1}^{N} \frac{1}{a_{\cdot\cdot}^{(s)}} \sum_{i=1}^{m} \sum_{j=1}^{n^{(s)}} \sum_{l=1}^{k} \left(a_{ij}^{(s)} - \frac{a_{i\cdot}^{(s)} a_{\cdot j}^{(s)}}{a_{\cdot\cdot}^{(s)}} \right) u_{il} v_{jl}^{(s)} \tag{3}$$

where $A = \cup \mathbf{A}^{(s)} (s = 1,2,\ldots,N)$ is the star-structured high-order heterogeneous data, and $\mathbf{A}^{(s)}$ is a co-occurrence matrix of O and $F^{(s)}$. $a_{\cdot\cdot}^{(s)} = \sum_{i,j} a_{ij}^{(s)}$ is the sum of the weights of all terms in $\mathbf{A}^{(s)}$, $a_{i\cdot}^{(s)} = \sum_j a_{ij}^{(s)}$ is the sum of the weights of all terms on the i-th row, and $a_{\cdot j}^{(s)} = \sum_i a_{ij}^{(s)}$ is the sum of the weights of all terms on the j-th column. $\mathbf{C}^{(s)} = \mathbf{U}\mathbf{V}^{(s)\mathrm{T}}$ is the relationship matrix, where $\mathbf{U}$ is a sample index matrix and $\mathbf{V}^{(s)}$ is a feature index matrix of the s-th feature space. Then we define a matrix $\mathbf{X}^{(s)} \in\in \mathrm{R}^{m\times n}$, and the term in $\mathbf{X}^{(s)}$ is $x_{ij}^{(s)} = \frac{a_{i\cdot}^{(s)} a_{\cdot j}^{(s)}}{a_{\cdot\cdot}^{(s)}}$. The objective function can also be expressed in the following form:

$$Q(A, C) = \sum_s Q(\mathbf{A}^{(s)}, \mathbf{C}^{(s)})$$
$$= \sum_s \frac{1}{a_{\cdot\cdot}^{(s)}} Trace\left[\left(\mathbf{A}^{(s)} - \mathbf{X}^{(s)}\right)^{\mathrm{T}} \mathbf{U}\mathbf{V}^{(s)\mathrm{T}}\right] \tag{4}$$

4.2 MHCoC algorithm

MHCoC uses a top-down strategy to perform hierarchical co-clustering to optimize the objective function. For easy understanding, we use an example to describe the MHCoC implement process. Figure 2 shows a toy example with contains central data type and three feature spaces. MHCoC first takes the O as the unique cluster r_{11} in the 1-th time sample clustering result R_1, and takes each $F^{(s)}$ as the unique cluster $c_{11}^{(s)}$ in the 1-th time feature clustering result $C_1^{(s)}$. MCC is then used to process $RC_{11} = \left\{ rc_{11}^{(s)} | s = 1, 2, 3 \right\}$, that is, each co-cluster is split into two sub-co-clusters in three feature spaces. This split can maximizing the module increase value $\Delta Q(A_{11}, C_{11})$. Accordingly, the 2-th time sample clustering result $R_2 = \{r_{21}, r_{22}\}$ and the 2-th time feature clustering result $C_2^{(s)} = \left\{c_{21}^{(s)}, c_{22}^{(s)}\right\} (s = 1,2,3)$ are generated. Then for the two co-cluster sets RC_{21} and RC_{22} in the 2-th time feature clustering result, MCC is used to find the optimal partition and compute the modularity increase value. It is assumed here that the modularity increase value of RC_{21} is larger, so



$c_{11}^{(1)}$
r_{11}
$rc_{11}^{(1)}$
$rc_{11}^{(2)}$
$rc_{11}^{(3)}$
$c_{11}^{(2)}$
$c_{11}^{(3)}$
$c_{21}^{(1)}$ $c_{22}^{(1)}$
r_{21} r_{22}
$rc_{21}^{(1)}$ $rc_{22}^{(1)}$
$rc_{21}^{(2)}$ $rc_{22}^{(2)}$ $rc_{21}^{(3)}$ $rc_{22}^{(3)}$
$c_{21}^{(2)}$ $c_{22}^{(2)}$ $c_{21}^{(3)}$ $c_{22}^{(3)}$
$c_{31}^{(1)}$ $c_{32}^{(1)}$ $c_{33}^{(1)}$
r_{31} r_{32} r_{33}
$rc_{31}^{(1)}$ $rc_{32}^{(1)}$ $rc_{33}^{(1)}$
$rc_{31}^{(2)}$ $rc_{32}^{(2)}$ $rc_{33}^{(2)}$ $rc_{31}^{(3)}$ $rc_{32}^{(3)}$ $rc_{33}^{(3)}$
$c_{31}^{(2)}$ $c_{32}^{(2)}$ $c_{33}^{(2)}$ $c_{31}^{(3)}$ $c_{32}^{(3)}$ $c_{33}^{(3)}$

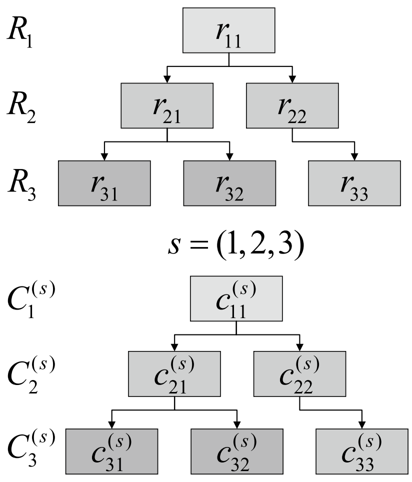
Fig. 2 MHCoC algorithm process example

each co-cluster in RC_{21} is respectively split into two sub-co-clusters, and other co-clusters are retained. Then the 3-th time co-clustering result $R_3=\{r_{31},r_{32},r_{33}\}$ and $C_3^{(s)}=\left\{c_{31}^{(s)},c_{32}^{(s)},c_{33}^{(s)}\right\}(s=1,2,3)$ are generated, and then the layer-by-layer split is continued with a top-down strategy. Finally, the MHCoC algorithm returns $R=\{R_1,R_2,\ldots,R_h,\ldots R_H\}$ and $C^{(s)}=\left\{C_1^{(s)},C_2^{(s)},\ldots,C_h^{(s)},\ldots C_H^{(s)}\right\}(s=1,2,\ldots,N)$ as shown in Fig. 3. The algorithm is described in more details as follows:

```
Algorithm 1 MHCoC
Input: star-structure high-order heterogeneous data $A$, the number of clusters $k$
Output:
$R=\{R_1,R_2,\ldots,R_h,\ldots R_H\}$, $C^{(s)}=\left\{C_1^{(s)},C_2^{(s)},\ldots,C_h^{(s)},\ldots C_H^{(s)}\right\}$ $(s=1,2,\ldots,N)$
Initialization of $H=k$, $h=1$
while $h\le H-1$ do
    for all $RC_{hl}(l=1,2,\ldots,h)$ do
        $\{RC_{hl1},RC_{hl2},\Delta Q(A,C)_{hl}\}=\mathrm{MCC}(RC_{hl})$
            $l=\arg\max_{l^*}\Delta Q(A_{hl^*},C_{hl^*})$
            $R_{h+1}=\{R_h-r_{hl}\}\cup\{\boldsymbol{r_{hl1}},\boldsymbol{r_{hl2}}\}$
            $C_{h+1}^{(s)}=\left\{C_h^{(s)}-c_{hl}\right\}\cup\{\boldsymbol{c_{hl1}},\boldsymbol{c_{hl2}}\}$
    $h=h+1$
end while
Return $R, C^{(s)}$ $(s=1,2,\ldots,N)$
```

4.3 MCC algorithm

For given data matrix **A** and relationship matrix **C**, the following two propositions are hold:



R_1
r_{11}
R_2
r_{21} r_{22}
R_3
r_{31} r_{32} r_{33}
$s=(1,2,3)$
$C_1^{(s)}$
$c_{11}^{(s)}$
$C_2^{(s)}$
$c_{21}^{(s)}$ $c_{22}^{(s)}$
$C_3^{(s)}$
$c_{31}^{(s)}$ $c_{32}^{(s)}$ $c_{33}^{(s)}$

Fig. 3 Hierarchical clustering results

Proposition 1

$$Q(\mathbf{A},\mathbf{C})=\frac{1}{a_{..}}Trace\left[(\mathbf{A}-\mathbf{X})^{\mathrm{T}}\mathbf{U}\mathbf{V}^{\mathrm{T}}\right]$$
$$=\frac{1}{a_{..}}Trace\left[\left(\mathbf{A}^{\mathbf{V}}-\mathbf{X}^{\mathbf{V}}\right)^{\mathrm{T}}\mathbf{U}\right]=Q(\mathbf{A}^{\mathbf{V}},\mathbf{U})$$

The matrices $\mathbf{A}^{\mathbf{V}}$ *and* $\mathbf{X}^{\mathbf{V}}$ *are computed from the matrices* **A** *and* **X** *according to the column index matrix* **V**:

$$\mathbf{A}^{\mathbf{V}}:=\left\{a_{il}^{\mathbf{V}}=\sum_{j=1}^{n}v_{jl}a_{ij};i=1,\ldots,m;l=1,\ldots,k\right\},$$

$$\mathbf{X}^{\mathbf{V}}:=\left\{x_{il}^{\mathbf{V}}=\frac{a_{i.}a_{.l}^{\mathbf{V}}}{a_{..}};i=1,\ldots,m;l=1,\ldots,k\right\},$$

where $a_{.l}^{\mathbf{V}}=\sum_{j=1}^{n}v_{jl}a_{.j}$.

Proof

$$
\begin{aligned}
Q(\mathbf{A},\mathbf{C}) &= \frac{1}{a_{..}} Trace\left[(\mathbf{A}-\mathbf{X})^{\mathrm{T}}\mathbf{U}\mathbf{V}^{\mathrm{T}}\right] \\
&= \frac{1}{a_{..}}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l}^{k}\left(a_{ij}-\frac{a_{i.}a_{.j}}{a_{..}}\right)u_{il}v_{jl} \\
&= \frac{1}{a_{..}}\sum_{i=1}^{m}\sum_{l=1}^{k}u_{il}\sum_{j=1}^{n}v_{jl}\left(a_{ij}-\frac{a_{i.}a_{.j}}{a_{..}}\right) \\
&= \frac{1}{a_{..}}\sum_{i=1}^{m}\sum_{l=1}^{k}u_{il}\left(\sum_{j=1}^{n}v_{jl}a_{ij}-\frac{a_{i.}}{a_{..}}\sum_{j=1}^{n}v_{jl}a_{.j}\right) \\
&= \frac{1}{a_{..}}\sum_{i=1}^{m}\sum_{l=1}^{k}\left(a_{il}^{\mathbf{V}}-\frac{a_{i.}a_{.l}^{\mathbf{V}}}{a_{..}}\right)u_{il} \\
&= \frac{1}{a_{..}} Trace\left[\left(\mathbf{A}^{\mathbf{V}}-\mathbf{X}^{\mathbf{V}}\right)^{\mathrm{T}}\mathbf{U}\right] = Q(\mathbf{A}^{\mathbf{V}},\mathbf{U})
\end{aligned}
$$

Proposition 2

$$
Q(\mathbf{A},\mathbf{C}) = \frac{1}{a_{..}} Trace\left[(\mathbf{A}-\mathbf{X})^{\mathrm{T}}\mathbf{U}\mathbf{V}^{\mathrm{T}}\right] = \frac{1}{a_{..}} Trace\left[\left(\mathbf{A}^{\mathbf{U}}-\mathbf{X}^{\mathbf{U}}\right)^{\mathrm{T}}\mathbf{V}\right] = Q(\mathbf{A}^{\mathbf{U}},\mathbf{V}).
$$

The matrices $\mathbf{A}^{\mathbf{U}}$ *and* $\mathbf{X}^{\mathbf{U}}$ *are computed from the matrices* $\mathbf{A}$ *and* $\mathbf{X}$ *according to the column index matrix* $\mathbf{U}$: $\mathbf{A}^{\mathbf{U}} := \left\{a_{jl}^{\mathbf{U}} = \sum_{i=1}^{m} u_{il}a_{ij}; j=1,\dots,n; l=1,\dots,k\right\}$,

$\mathbf{X}^{\mathbf{U}} := \left\{x_{jl}^{\mathbf{U}} = \frac{a_{j.}a_{.l}^{\mathbf{U}}}{a_{..}}; j=1,\dots,n; l=1,\dots,k\right\}$, *where* $a_{.l}^{\mathbf{U}} = \sum_{i=1}^{m} u_{il}a_{i.}$.

Proof

$$
\begin{aligned}
Q(\mathbf{A},\mathbf{C}) &= \frac{1}{a_{..}} Trace\left[(\mathbf{A}-\mathbf{X})^{\mathrm{T}}\mathbf{U}\mathbf{V}^{\mathrm{T}}\right] \\
&= \frac{1}{a_{..}}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l}^{k}\left(a_{ij}-\frac{a_{i.}a_{.j}}{a_{..}}\right)u_{il}v_{jl} \\
&= \frac{1}{a_{..}}\sum_{j=1}^{n}\sum_{l=1}^{k}v_{jl}\sum_{i=1}^{m}u_{il}\left(a_{ij}-\frac{a_{i.}a_{.j}}{a_{..}}\right) \\
&= \frac{1}{a_{..}}\sum_{j=1}^{n}\sum_{l=1}^{k}v_{jl}\left(\sum_{i=1}^{m}u_{il}a_{ij}-\frac{a_{i.}}{a_{..}}\sum_{i=1}^{m}u_{il}a_{i.}\right) \\
&= \frac{1}{a_{..}}\sum_{j=1}^{n}\sum_{l=1}^{k}\left(a_{jl}^{\mathbf{U}}-\frac{a_{i.}a_{.l}^{\mathbf{U}}}{a_{..}}\right)v_{jl} \\
&= \frac{1}{a_{..}} Trace\left[\left(\mathbf{A}^{\mathbf{U}}-\mathbf{X}^{\mathbf{U}}\right)^{\mathrm{T}}\mathbf{V}\right] \\
&= Q(\mathbf{A}^{\mathbf{U}},\mathbf{V})
\end{aligned}
$$

According to the above two Propositions, we can iteratively update the column index matrixes according to the latest row cluster index matrix or update the row index matrixes according to the latest column index matrixes.

In each iteration, using proposition 1, the row index matrix $\mathbf{U}$ is first updated according to the current column index matrices $V=\{\mathbf{V}^{(s)}| s=1,2,\dots,N\}$ to maximize $Q(\mathbf{A}^{V},\mathbf{U})$:

$$
\begin{aligned}
\mathbf{U} &= \arg\max_{\mathbf{U}*} Q\left(\mathbf{A}^{V},\mathbf{U}\right) \\
&= \arg\max_{\mathbf{U}*}\sum_{s=1}^{N} Trace\left(\mathbf{A}^{\mathbf{V}(s)}-\mathbf{X}^{\mathbf{V}(s)}\right)^{\mathrm{T}}\mathbf{U}*
\end{aligned} \tag{5}
$$

the term in the matrix $\mathbf{A}^{V}$ is $a_{il}^{V} = \sum_{s=1}^{N} a_{il}^{\mathbf{V}(s)}$, and the term in the matrix $\mathbf{X}^{V}$ is $x_{il}^{V} = \sum_{s=1}^{N} x_{il}^{\mathbf{V}(s)}$. Then according to the updated row index matrix $\mathbf{U}$, each column index matrix $\mathbf{V}^{(s)}(s=1,2,\dots,N)$ is updated using proposition 2 to maximize $Q(\mathbf{A}^{\mathbf{U}(s)},\mathbf{V}^{(s)})$:

$$
\begin{aligned}
\mathbf{V}^{(s)} &= \arg\max_{\mathbf{V}}^{(s)*} Q\left(A^{\mathbf{U}(s)},\mathbf{V}^{(s)}\right) \\
&= \arg\max_{\mathbf{V}}^{(s)*} Trace\left(\mathbf{A}^{\mathbf{U}(s)}-\mathbf{X}^{\mathbf{U}(s)}\right)^{\mathrm{T}}\mathbf{V}^{(s)}*
\end{aligned} \tag{6}
$$

the term in the matrix $\mathbf{A}^{\mathbf{U}}$ is $a_{jl}^{\mathbf{U}} = \sum_{s=1}^{N} a_{jl}^{\mathbf{U}}$, and the term in the matrix $\mathbf{X}^{\mathbf{U}}$ is $x_{jl}^{\mathbf{U}} = \sum_{s=1}^{N} x_{jl}^{\mathbf{U}}$.The modularity increase value $\Delta Q(A,C)$ is computed after each iteration. Since the MCC only processes one co-cluster set, the global modularity increase value is only generated in the split of this group of co-clusters:

$$
\begin{aligned}
\Delta Q(A,C) = &\sum_{s=1}^{N}\frac{1}{a_{..}^{(s)}}\sum_{i=1}^{m}\sum_{j=1}^{n^{(s)}}\sum_{l=1}^{k}\left(a_{ij}^{(s)}-\frac{a_{i.}^{(s)}a_{.j}^{(s)}}{a_{..}^{(s)}}\right)u_{il}v_{jl}^{(s)} \\
&-\sum_{s=1}^{N}\frac{1}{a_{..}^{(s)}}\sum_{i=1}^{m}\sum_{j=1}^{n^{(s)}}\left(a_{ij}^{(s)}-\frac{a_{i.}^{(s)}a_{.j}^{(s)}}{a_{..}^{(s)}}\right)
\end{aligned} \tag{7}
$$

Taking RC_{21} in Fig. 2 as an example, the following example is designed to describe the execution process of the MCC:

$$
\mathbf{A}_{21}^{(1)}=\begin{pmatrix}1&0&1&1&0\\0&1&0&0&1\\1&0&1&1&0\\0&1&0&0&1\end{pmatrix}
\mathbf{A}_{21}^{(2)}=\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&1&0\\0&1&0&1\end{pmatrix}
\mathbf{A}_{21}^{(3)}=\begin{pmatrix}1&0&0&0&1\\0&1&1&1&0\\1&0&0&0&1\\0&1&1&1&0\end{pmatrix}
$$

Initialization of $\mathbf{V}_{hl}^{(s)}$

$$\mathbf{V}_{21}^{(1)}=\begin{pmatrix}1&0\\1&0\\0&1\\1&0\\0&1\end{pmatrix}\mathbf{V}_{21}^{(2)}=\begin{pmatrix}1&0\\0&1\\1&0\\0&1\end{pmatrix}\mathbf{V}_{21}^{(3)}=\begin{pmatrix}0&1\\0&1\\1&0\\0&1\\1&0\end{pmatrix}$$

Compute $\mathbf{A}_{hl}^{V}-\mathbf{X}_{hl}^{V}$ according to $\mathbf{V}_{hl}^{(s)}$

$$\mathbf{A}_{21}^{\mathbf{V}(1)}-\mathbf{X}_{21}^{\mathbf{V}(1)}=\begin{pmatrix}0.2&-0.2\\-0.2&0.2\\0.2&-0.2\\-0.2&0.2\end{pmatrix}\mathbf{A}_{21}^{\mathbf{V}(2)}-\mathbf{X}_{21}^{\mathbf{V}(2)}=\begin{pmatrix}1&-1\\-1&1\\1&-1\\-1&1\end{pmatrix}$$

$$\mathbf{A}_{21}^{\mathbf{V}(3)}-\mathbf{X}_{21}^{\mathbf{V}(3)}=\begin{pmatrix}0.2&-0.2\\-0.2&0.2\\0.2&-0.2\\-0.2&0.2\end{pmatrix}\mathbf{A}_{21}^{V}-\mathbf{X}_{21}^{V}=\begin{pmatrix}1.4&-1.4\\-1.4&1.4\\1.4&-1.4\\-1.4&1.4\end{pmatrix}$$

Compute $\mathbf{U}_{hl}$

$$\mathbf{U}_{21}=\begin{pmatrix}1&0\\0&1\\1&0\\0&1\end{pmatrix}$$

Compute $\mathbf{A}_{hl}^{\mathbf{U}(s)}-\mathbf{X}_{hl}^{\mathbf{U}(s)}$ according to $\mathbf{U}_{hl}$

$$\mathbf{A}_{21}^{\mathbf{U}(1)}-\mathbf{X}_{21}^{\mathbf{U}(1)}=\begin{pmatrix}0.8&-0.8\\-1.2&1.2\\0.8&-0.8\\0.8&-0.8\\-1.2&1.2\end{pmatrix}$$

$$\mathbf{A}_{21}^{\mathbf{U}(2)}-\mathbf{X}_{21}^{\mathbf{U}(2)}=\begin{pmatrix}1&-1\\-1&1\\1&-1\\-1&1\end{pmatrix}$$

$$\mathbf{A}_{21}^{\mathbf{U}(3)}-\mathbf{X}_{21}^{\mathbf{U}(3)}=\begin{pmatrix}1.2&-1.2\\-0.8&0.8\\-0.8&0.8\\-0.8&0.8\\1.2&-1.2\end{pmatrix}$$

Compute $\mathbf{V}_{hl}^{(s)}$

$$\mathbf{V}_{21}^{(1)}=\begin{pmatrix}1&0\\0&1\\1&0\\1&0\\0&1\end{pmatrix}\mathbf{V}_{21}^{(2)}=\begin{pmatrix}1&0\\0&1\\1&0\\0&1\end{pmatrix}\mathbf{V}_{21}^{(3)}=\begin{pmatrix}1&0\\0&1\\0&1\\0&1\\1&0\end{pmatrix}$$

Compute $\Delta Q(A_{hl},C_{hl})$

$$\Delta Q(A_{21},C_{21}) = \sum_{s=1}^{3}\frac{1}{a_{..}^{(s)}}\sum_{i=1}^{4}\sum_{j^{(s)}=1}^{n^{(s)}}\left(a_{ij}^{(s)}-\frac{a_{i\cdot}^{(s)}a_{\cdot j}^{(s)}}{a_{..}^{(s)}}\right)u_{il}v_{jl}^{(s)} - \sum_{s=1}^{3}\frac{1}{a_{..}^{(s)}}\sum_{i=1}^{4}\sum_{j^{(s)}=1}^{n^{(s)}}\left(a_{ij}^{(s)}-\frac{a_{i\cdot}^{(s)}a_{\cdot j}^{(s)}}{a_{..}^{(s)}}\right) = 1.46$$

then iterating. This alternated co-clustering is described in more details in algorithm 2.

```
Algorithm 2 MCC
INPUT: $A_{hl}=\left\{\mathbf{A}_{hl}^{(1)},\mathbf{A}_{hl}^{(2)},\ldots,\mathbf{A}_{hl}^{(N)}\right\}$
OUTPUT: $\mathbf{U}_{hl}$, $\mathbf{V}_{hl}^{(s)}$ (s=1,2,…,N)
1. Initialization of $\mathbf{V}_{hl}^{(s)}$ (s=1,2,…,N)
Repeat
2. Compute $\mathbf{A}_{hl}^{V}-\mathbf{X}_{hl}^{V}$
    $\mathbf{A}^{\mathbf{V}(s)} := \left\{a_{il}^{\mathbf{V}(s)}=\sum_{j=1}^{n}v_{jl}a_{ij};\ i=1,\ldots,m, l=1,2\right\}$
    $\mathbf{X}^{\mathbf{V}(s)} := \left\{x_{il}^{\mathbf{V}(s)}=\frac{a_{i\cdot}a_{\cdot l}^{\mathbf{V}(s)}}{a_{..}};\ i=1,\ldots,m, l=1,2\right\}$
    $\mathbf{A}^{\mathbf{V}(s)}-\mathbf{X}^{\mathbf{V}(s)} := \left\{a_{il}^{\mathbf{V}(s)}-x_{il}^{\mathbf{V}(s)};\ i=1,\ldots,m, l=1,2\right\}$
    $\mathbf{A}_{hl}^{V}-\mathbf{X}_{hl}^{V} := \left\{\sum_{s=1}^{N}\left(a_{il}^{\mathbf{V}(s)}-x_{il}^{\mathbf{V}(s)}\right);\ i=1,\ldots,m, l=1,2\right\}$
3. Compute $\mathbf{U}_{hl}$
  $\mathbf{U}_{hl}$=0,i=1
  while i≤m do
      $u_{il^*}=1$
    where $l^*=\arg\max_{l}\left(\mathbf{A}^{V}-\mathbf{X}^{V}\right)_{il}$
    i=i+1
4. Compute $\mathbf{A}_{hl}^{\mathbf{U}}-\mathbf{X}_{hl}^{\mathbf{U}}$
    $\mathbf{A}^{\mathbf{U}} := \left\{a_{jl}^{\mathbf{U}}=\sum_{i=1}^{m}v_{il}a_{ij};\ j=1,\ldots,n, l=1,2\right\}$
    $\mathbf{X}^{\mathbf{U}} := \left\{x_{jl}^{\mathbf{U}}=\frac{a_{j\cdot}a_{\cdot l}^{\mathbf{U}}}{a_{..}};\ j=1,\ldots,n, l=1,2\right\}$
    $\mathbf{A}_{hl}^{\mathbf{U}}-\mathbf{X}_{hl}^{\mathbf{U}} := \left\{z_{jl}=a_{jl}^{\mathbf{U}}-x_{jl}^{\mathbf{U}};\ j=1,\ldots,n, l=1,2\right\}$
5. Compute $\mathbf{V}_{hl}^{(s)}$ (s=1,2,…,N)
  $\mathbf{V}_{hl}^{(s)}$=0, j=1
  while j≤n do
      $v_{jl}^{(s)}=1$
    where $l^*=\arg\max_{l}\left(\mathbf{A}^{\mathbf{U}}-\mathbf{X}^{\mathbf{U}}\right)_{jl}$
    j=j+1
6. Compute Q(A,C)
Until nochange of Q(A,C)
Return $\mathbf{U}_{hl}$, $\mathbf{V}_{hl}^{(s)}$ (s=1,2,…,N)
```

5 Experiments and results analysis

In this section, we perform experiments on two real-world datasets to evaluate our proposed method. We aim to answer the following research questions:

• RQ1: How does MHCoC perform as compared with state-of-the-art homogeneous co-clustering methods?

• RQ2: Compared with existing high-order heterogeneous clustering methods, how does MHCoC perform?

• RQ3: What are the benefits of tree-like clustering results generated from MHCoC?

In the following parts, we will first present the experimental settings and then answer the above research questions one by one.

5.1 Experimental setting

5.1.1 Datasets description

In order to verify comprehensively and objectively, we use four different real-world datasets spanning different domains to verify the effectiveness of our approach.

Reuters is a multilingual dataset consists of 2000 samples with 5 types of languages. We use the subset that is written in English, while the other 4 features are its corresponding translations in 4 different languages.

USPS dataset consists of features of handwritten numerals and there are 2000 examples uniformly distributed in 10 classes, and three types of features are used.

20Newsgroup corresponds to about 20,000 news documents from 20 different newsgroups with each newsgroup containing 1000 documents. Two types of features are used.

Corel5k dataset contains 5000 images, each with text annotation information and image segmentation. We construct high-order heterogeneous dataset including image, word and blobs. We summarize the statistics of the datasets in Table 1.

Table 1 Datasets

Dataset	#Object	#Class	Type	Feature
Reuters	1200	6	text	English, French, German, Spanish, Italian
USPS	2000	10	image	fourier coefficients, profile correlations, zernike moments
20Newsgroup	19,949	20	text	subject, word
Corel5k	5000	5	image	word, blobs

5.1.2 Evaluation protocols

To evaluate the clustering performance, we focus primarily on the Macro-F_1 score and NMI value which measures how well each algorithm finds the ground truth clusters. Specifically, given a set of algorithmic clusters C, and the ground truth clusters S, the Macro-F_1 score is computed by averaging the F_1 score of the best match between each ground truth cluster and algorithmic clusters:

$$\text{Macro} - F_1 = \frac{1}{|S|}\sum_{s\in S}\text{F}_1(s) \tag{8}$$

The F_1 score of a single ground truth cluster s is computed as the harmonic mean of P(Precision) and R(Recall):

$$\text{F}_1(s) = \frac{2\times P\times R}{P+R} \tag{9}$$

where Precision P and Recall R can be calculated as follows:

$$P = \frac{TP}{TP+FP} \tag{10}$$

$$R = \frac{TP}{TP+FN} \tag{11}$$

TP, TN, FP, FN represent True Positive (The fact is the positive samples, which were judged as positive samples), True Negative, False Positive, False Negative, respectively.

NMI value is calculated as:

$$\text{NMI} = \frac{I(S,C)}{(H(S)+H(C))/2} \tag{12}$$

where the molecule is the mutual information of S and C, and the denominator is the average of the entropy of S and C:

$$I(S,C) = \sum_{s_i\in S}\sum_{c_j\in C} p\left(s_i\cap c_j\right)\log\frac{p\left(s_i\cap c_j\right)}{p\left(s_i\right)p\left(c_j\right)} \tag{13}$$

$$H(S) = -\sum_{s_i\in S} p\left(s_i\right)\log p\left(s_i\right) \tag{14}$$

Since the clustering algorithm involved in this experiment is random, each clustering algorithm is executed 30 times on each data set, and the average value of 30 clustering results are used for comparative analysis.

5.2 Performance comparison with homogeneous co-clustering methods (RQ1)

We first compare our proposed methods against other competing homogeneous co-clustering approaches, spanning from the information theory based approach *ITCC* [38]:

a classical co-clustering algorithm maximizes the mutual information between the clustered random variables subject to constraints on the number of row and column clusters, and the modularity-based approach *Coclus* [1]: a block diagonal co-clustering algorithm that implements automatic feature reduction during co-clustering, and matrix factorization based approach *NMTFCoS* [39]: a non-negative matrix tri-factorization model based on co-sparsity regularization, with its objective to simultaneously minimize the loss function for the matrix tri-factorization, and the deep co-clustering algorithm *DCC* [17]: a deep co-clustering model jointly optimizes the parameters of the deep autoencoder and the mixture model in an end-to-end fashion on both the instance and the feature spaces.

For the above four baseline methods, all the features from each feature space are simply combined into one feature space. The input parameters of the comparing methods are set as the authors suggested in their papers or shared code. The clustering performance comparison of each method is shown in Tables 2 and 3, respectively. We have the following observations:

- ITCC achieves poorest performance on all the datasets. This indicates that maximizing the mutual information between the clustered random variables is insufficient to capture the complex relations among different feature spaces, further limiting the performance. NMTFCoS consistently outperforms ITCC across all cases, verifying that incorporating the co-sparsity regularization can improve the clustering results. Compared to ITCC and NMTFCoS, the performance of Coclus demonstrates the effectiveness of modularity for co-clustering. DCC reflects the superior performance of deep learning, but as a method for homogeneous data, it still has limitations when dealing with heterogeneous data.
- As expected, MHCoC consistently yields the best performance on all the datasets. Through dealing with multiple feature space under a unified framework, MHCoC is capable of exploring the high-order heterogonous features in an explicit way, while other baseline methods only simply merge all the features to guide the clustering process. This verifies the importance of properly capturing heterogonous features in the objective function.

Table 3 Macro-F1 comparison of the homogeneous co-clustering methods

	Reuters	USPS	20Ng	Corel5k
ITCC	14.4	19.4	16.5	16.4
Coclus	23.4	27.5	33.6	20.7
NMTFCoS	16.7	30.5	28.4	23.9
DCC	22.4	45.6	48.3	40.3
MHCoC	30.6	68.3	71.7	62.3

In summary, the four excellent baseline homogeneous co-clustering algorithms are based on different theories, their performance degenerate when dealing with heterogeneous data.

5.3 Performance comparison with high-order heterogeneous clustering methods (RQ2)

To answer RQ2, we compare the proposed approach against four high-order heterogeneous clustering methods *CBGC* [13] is a consistent bipartite graph co-partitioning co-clustering method based on semi-definite programming. *CIT* [14] extends the information theoretic co-clustering algorithm to solve the high-order co-clustering problem. *O-NMTF* [40] employs Nonnegative Matrix Tri-Factorization (NMTF) to simultaneously cluster different types of data using the inter-type relationships, and incorporate intra-type information through manifold regularization. *CoStar* [9] optimizes the measure for cross-association in contingency tables that evaluates the strength of the relationship between two categorical variables by a local search approach.

Likewise, the input parameters of the comparing methods are set as the authors suggested in their papers or shared code. For Reuters and USPS datasets, CIT uses the first two features of each dataset. The clustering performance comparisons on the experimental data sets are shown in Tables 4 and 5. From the results, we observed that:

- CBGC and CIT achieve the worse performance than other baselines over all the datasets in all cases. It is reasonable since they both model high-order co-clustering

Table 2 NMI comparison of the homogeneous co-clustering methods

	Reuters	USPS	20Ng	Corel5k
ITCC	15.5	22.7	21.7	16.4
Coclus	23.4	28.5	40.4	36.6
NMTFCoS	18.7	21.5	30.7	23.9
DCC	34.6	48.3	40.8	36.4
MHCoC	40.5	81.7	76.7	69.4

Table 4 NMI comparison of the heterogeneous clustering methods

	Reuters	USPS	20Ng	Corel5k
CBGC	31.2	51.5	60.3	41.3
CIT	28.5	45.2	63.3	51.4
O-NMTF	36.1	60.3	70.5	56.6
CoStar	34.2	61.4	67.3	59.7
MHCoC	40.5	81.7	76.7	69.4

Table 5 Macro-F1 comparison of the heterogeneous clustering methods

	Reuters	USPS	20Ng	Corel5k
CBGC	23.5	51.5	57.3	37.6
CIT	20.5	40.2	55.3	51.4
O-NMTF	26.1	50.6	56.5	54.6
CoStar	27.6	57.4	61.1	53.7
MHCoC	30.6	68.3	71.7	62.3

problem as the consistent fusion of pair-wise co-clustering sub-problems.

- O-NMTF achieves better results in terms of both metrics, showing the importance of the factor's orthogonalities can lead to better results. Costar shows similar results, as it benefits from that the τ functions used to compare co-clustering of different sizes. Further, compared with first two methods, these two algorithms demonstrate that it is a better strategy to dealing with multiple feature space under a unified framework.
- Generally speaking, MHCoC achieves better performance than other baselines over all datasets, sometimes very significantly, which confirms the effectiveness of MHCoC in co-clustering of high-order heterogeneous data. We mainly attribute the improvement of the clustering performance to the advantage of modularity and the objective function we designed which can properly combine the information from the high-order heterogeneous data. Moreover, MHCoC has implicit subspace selection in the cluster partitioning process, thus it shows better performance on high-dimensional sparse data high-order co-clustering.

5.4 Hierarchical co-clustering analysis (RQ3)

In this section, we attempt to demonstrate how the hierarchical co-clustering facilitates the understanding of clustering result. Towards this end, we select five newsgroups with different relationships from the 20NG dataset to form the NG5 dataset for experiments. The five newsgroups we selected are 'talk.politics.mideast', 'sci.space', 'comp.graphics', 'rec.motorcycles' and 'rec.sport.baseball'. It is easy to see that in NG5, 'sci.space' and 'comp.graphics' are similar, and so are 'rec.motorcycles' and 'rec.sport.baseball', but there are obvious differences between these two groups. Besides, 'talk.politics.mideast' shares no obvious relationship with other classes. Therefore, the NG5 dataset can representatively and intuitively show how the hierarchical strategy reflects the relationship between clusters. We initially run MHCoC on NG5 to get 5 clusters, then continue to split into 20 clusters.

The hierarchical object clustering results are shown in Fig. 4. The root consists of the entire sample set. After four splits, five clusters represented by rectangle with a thickened border are obtained, and the circles below indicate the results of continued splitting. The number represents which class the cluster corresponds to. Intuitively, the two clusters 'comp.graphics' and 'sci.space' are from the same cluster from the previous layer, and 'rec.motorcycles' and 'rec.sport.baseball' are from the same cluster in the previous layer. This structure has practical meaning and meets our expectations. Both the 'comp.graphics' and 'sci.space' clusters involve science and technology. The two clusters 'rec.motorcycles' and 'rec.sport.baseball' belong to the entertainment category. However, the difference between

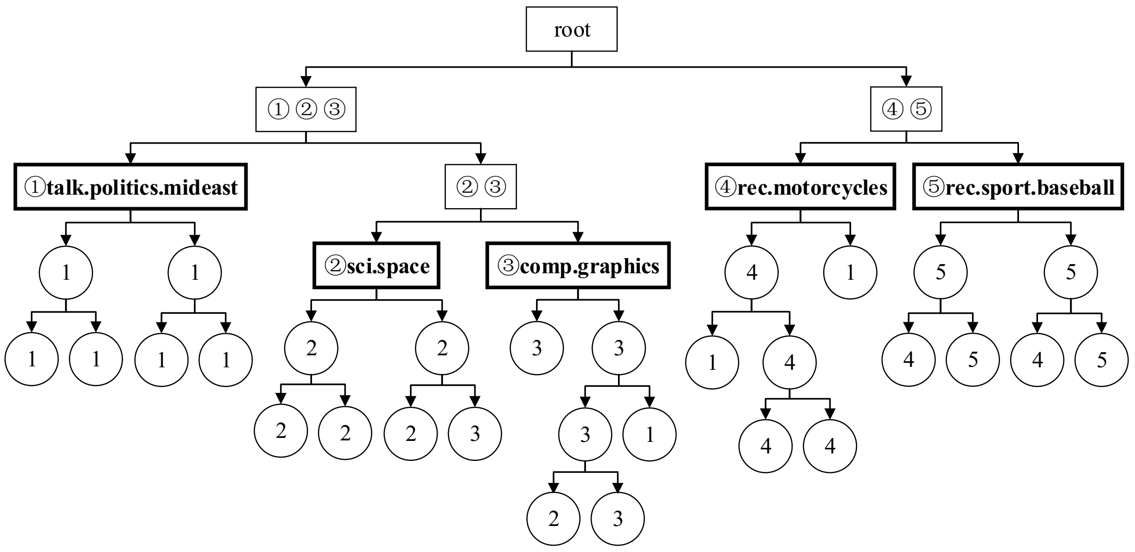


Fig. 4 Result of visual hierarchical cluster structure

Table 6 Keywords in different clusters

①talk.politics.mideast	②sci.space	③comp.graphics	④rec.motorcycles	⑤rec.sport.baseball
Government	University	Image	Clutch	Baseball
War	Spacecraft	Program	Driving	Players
Turkish	Nasa	Information	Engine	Season
Right	Orbit	Data	Lock	Pitcher
Greek	Launch	Systems	Car	Brave

the two clusters of 'comp.graphics and sci.space' and 'rec. motorcycles and rec.sport.baseball' is relatively large. In addition, it can be seen from the Fig. 4 that during the first split, 'talk.politics.mideast', which has no obvious relationship with other classes, is divided into the same cluster as 'sci.space and comp.graphics', and then it is split from them. Although in the subsequent splitting process, we can see that some of the samples in 'talk.politics.mideast' also appeare in the 'rec.motorcycles' that was divided with them at the first splitting, but from the results, there seems to be more in common between 'talk.politics.mideast' and 'comp.graphics and sci.space'.

In summary, we find that compared with the flat cluster structure, the hierarchical cluster structure is more conducive to reveal the implicit relationship between clusters and clusters.

We then take the word feature space as an example. Table 6 shows the first five words in the corresponding word cluster of each sample cluster obtained by sorting according to the method in [41]. It can be seen intuitively that the word cluster has a clear indication of the object cluster. This shows that the block diagonal co-clustering includes the implicit hard subspace selection process, and the dimensionality reduction function brings the algorithm to improve the clustering effect and efficiency.

Conclusion

In this paper, we propose a hierarchical high-order co-clustering algorithm by maximizing modularity. Our approach merges information of multiple types of object data and optimizes modularity-based objective function by performs high-order co-clustering, finally the tree-like hierarchical high-order co-clustering results are obtained. The effectiveness of the proposed method is proved on the real data set. Future work will focus on the study of overlapping hierarchical high-order co-clustering algorithms.

Acknowledgements The work is supported by the National Natural Science Foundation of China (No.61762078, 61363058, 61762079), Guangxi Key Laboratory of Trusted Software (No. kx201910) and Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (MIMS18-08)

Declarations

Conflict of interest No conflicts of interests.

References

1. Ailem M, Role F, Nadif M Co-clustering document-term matrices by direct maximization of graph modularity. In: Proc. 24th ACM Int. Conf. Inf. Knowl. Manage, 2015 pp 1807–1s810.
2. Wang Y, Feng C, Guo C, Chu Y, Hwang J (2019) Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering. In: Proc. 12th ACM Int. Conf. Web Search Data Mining, 717–725.
3. Feng L, Zhao Q, Zhou C (2020) Improving performances of Top-N recommendations with co-clustering method, Expert Syst. Appl., 143.
4. Chen X, Huang JZ, Wu Q, Yang M (2019) Subspace weighting co-clustering of gene expression data. IEEE/ACM Trans Comput Biol Bioinform 16(2):352–364
5. Hussain SF, Iqbal S (2018) CCGA: Co-similarity based Co-clustering using genetic algorithm. Appl Soft Comput 72:30–42
6. Keuper M, Tang S, Andres B, Brox T, Schiele B (2020) Motion segmentation & multiple object tracking by correlation co-clustering. IEEE Trans Pattern Anal Mach Intell 42(1):140–153
7. Meng L, Tan A, Xu D (2014) Semi-supervised heterogeneous fusion for multimedia data co-clustering. IEEE Trans Knowl Data Eng 26(9):2293–2306
8. Cheng W, Zhang X, Pan F, Wang W (2016) HICC: an entropy splitting-based framework for hierarchical co-clustering. Knowl Inf Syst 46(2):343–367
9. Ienco D, Robardet C, Pensa RG, Meo R (2013) Parameter-less co-clustering for star-structured heterogeneous data. Data Min Knowl Discov 26(2):217–254
10. Yin M, Gao J, Xie S, Guo Y (2019) Multiview subspace clustering via tensorial t-product representation. IEEE Trans Neural Netw Learning Syst 30(3):851–864
11. Huang L, Chao H, Wang C (2019) Multi-view intact space clustering. Pattern Recogn 86:344–353
12. Yin M, Gao J, Xie S, Guo Y (2020) Auto-weighted multi-view co-clustering with bipartite graphs. Inf Sci 512:18–30
13. Gao B, Liu T, Zheng X, Cheng Q, Ma W (2005) Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In: Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp 41–50.
14. Gao B, Liu T, Ma W (2006) Star-structured high-order heterogeneous data co-clustering based on consistent information theory, In: Proc. 6th ACM Int. Conf. Data Mining, pp 880–884.
15. Chen Y, Wang L, Dong M (2009) Non-negative matrix factorization for semi-supervised heterogeneous data co-clustering. IEEE Trans Knowl Data Eng 22(10):1459–1474

16. Wang S, Guo W (2017) Robust co-clustering via dual local learning and high-order matrix factorization, Knowl.-Based Syst., 138:176–17.
17. Xu D, Cheng W, Zong B, Ni J, Song D, Yu W, Chen Y, Chen H, Zhang X (2019) Deep Co-Clustering. In: Proc. SIAM Int. Conf. Data Mining, pp 414–422.
18. Papalexakis EE, Sidiropoulos ND, Bro R (2013) From K -means to higher-way co-clustering: multilinear decomposition with sparse latent factors. IEEE Trans. Signal Processing 61(2):493–506
19. Dhillon IS (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp 269–274.
20. Li T (2005) A general model for clustering binary data, In: Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp 188–197.
21. Labiod L, Nadif M (2011) Co-clustering for binary and categorical data with maximum modularity. In: Proc. 11th ACM Int. Conf. Data Mining, pp 1040–1045.
22. Han H, Dong X, Zuo C (2020) A weighted recommendation algorithm based on multiview clustering of user. J Intell Fuzzy Syst 38(1):441–451
23. Li J, Wang C, Li P, Lai J (2018) Discriminative metric learning for multi-view graph partitioning. Pattern Recogn 75:199–213
24. Kim Y, Amini M, Goutte C, Gallinari P (2010) Multi-view clustering of multilingual documents, In: Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., pp 821–822.
25. Zhang M, Yang Y, Shen F, Zhang H, Wang Y (2017) Multi-view feature selection and classification for Alzheimer's disease diagnosis. Multimedia Tools Appl 76(8):10761–10775
26. Zhan K, Chang X, Guan J, Chen L, Ma Z, Yang Y (2019) Adaptive structure discovery for multimedia analysis using multiple features. IEEE Trans Cybernetics 49(5):1826–1834
27. Gao B, Liu T, Feng G, Qin T, Cheng Q, Ma W (2005) Hierarchical taxonomy preparation for text categorization using consistent bipartite spectral graph co-partitioning. IEEE Trans Knowl Data Eng 19(7):1263–1273
28. Greco G, Guzzo A, Pontieri L (2010) Coclustering multiple heterogeneous domains: linear combinations and agreements. IEEE Trans Knowl Data Eng 22(12):1649–1663
29. Sun Y, Yu Y, Han J (2009) Ranking-based clustering of heterogeneous information networks with star network schema, In: Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp 797–806.
30. Xie XL, Beni G (1991) A validity measure for fuzzy clustering. IEEE Trans Pattern Anal Mach Intell 13(8):841–847
31. Chen Y, Wang L, Dong M (2010) Non-negative matrix factorization for semi-supervised heterogeneous data co-clustering. IEEE Trans Knowl Data Eng 22(10):1459–1474
32. Cohen-Addad V, Kanade V, Mallmann-Trenn F, Mathieu C (2019) Hierarchical clustering: objective functions and algorithms, J ACM 66(4):26:1–26:42.
33. Charikar M, Chatziafratis V, Niazadeh R (2019) Hierarchical clustering better than average-linkage. In: Proc. 15th ACM-SIAM Symp. on Dis. Algor., pp 2291–2304.
34. Emmendorfer LR, Canuto AM (2021) A generalized average linkage criterion for hierarchical agglomerative clustering. Appl Soft Comput 100:106990
35. Shi P, Zhao Z, Zhong H, Zhong H, Shen H, Ding L (2021) An improved agglomerative hierarchical clustering anomaly detection method for scientific data. Concurrency and computation, Practice and Experience, 33(6):e6077
36. Diez I, Bonifazi P, Escudero I, Mateos B, Muñoz MA, Stramaglia S, Cortes JM (2015) A novel brain partition highlights the modular skeleton shared by structure and function. Sci Reports 5:10532
37. Hu M, Zeng K, Wang Y et al (2021) Threshold-based hierarchical clustering for person re-identification. Entropy 23(5):522
38. Dhillon IS, Mallela S, Modha DS (2003) Information-theoretic co-clustering, In: Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp 89–98.
39. Tan Q, Yang P, He J (2018) Feature co-shrinking for co-clustering. Pattern Recogn 77:12–19
40. Wang H, Nie F, Huang H, Ding CHQ (2011) Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation. In: *Proc. 11th ACM Int. Conf. Data Mining*, 2011, pp 774–783.
41. Slonim N, Tishby N (2000) Document clustering using word clusters via the information bottleneck method, In: Proc. 23rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., pp 208–215.