Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Cross Multi-Type Objects Clustering in Attributed Heterogeneous Information Network[☆]

Sheng Zhou [a,b,c], Jiajun Bu [a,b,c,*], Zhen Zhang [a,b,c], Can Wang [a,b,c], Lingzhou Ma [a,b], Jianfeng Zhang [a,b]

[a] *College of Computer Science, Zhejiang University, Hangzhou 310027, China*
[b] *Zhejiang Provincial Key Laboratory of Service Robot, Hangzhou 310027, China*
[c] *Alibaba Group, Hangzhou 310027, China*

ARTICLE INFO

Article history:
Received 25 March 2019
Received in revised form 23 December 2019
Accepted 28 December 2019
Available online 7 January 2020

Keywords:
Heterogeneous information network
Clustering
Attributed network

ABSTRACT

Real-world networks usually consist of a large number of interacting, multi-typed components which are usually referred as heterogeneous information networks (HIN). HIN that associated with various attributes on nodes is defined as attributed HIN (or AHIN). Clustering is a fundamental task for HIN and AHIN. However, most of the current existing methods focus on single type nodes and there is very limited existing work that groups objects of different types into the same cluster. This is largely due to the reasons that object similarities can either be attribute-based or link-based between same type of nodes and it is challenging to incorporate both in a unified framework. To bridge this gap, in this paper, we propose a framework, namely **C**ross **M**ulti-Type **O**bjects **C**lustering in **A**ttributed **H**eterogeneous **I**nformation **N**etwork, or *CMOC-AHIN*, to integrate both the attribute information and multi-type node clustering in a principled way. We empirically show superior performances of *CMOC-AHIN* on three large scale challenging data sets and also summarize insights on the performances compared to other state-of-the-arts methodologies.

1. Introduction

In the past decades, homogeneous information network has been attracting much attention, and numerous data mining tasks such as ranking, clustering and classification have been explored. Most of contemporary information networks analyses have a basic assumption that the type of objects or links is unique [1–3]. However, real systems usually consist of a large number of interacting, multi-typed components, such as social interactions, biological networks, and communication networks, etc. Such interconnected networks are usually referred to as heterogeneous information networks (HIN) [1]. Compared to the widely studied homogeneous network, an HIN contains richer structure and semantic information that provides plenty of research opportunities as well as challenges [4–7]. Further more, in some real-world HIN, objects are often associated with various attributes. For example, in bibliographic network, an author may be associated with attributes like country, organization, address etc. A conference may be associated with attributes like year, topic, place etc. An HIN with object attributes is called an attributed HIN or AHIN for short [6,8].

Clustering is a fundamental task in data mining. It aims at partitioning a set of data objects (or observations) into a set of clusters, such that objects in the same cluster are similar to each other, yet dissimilar to objects in other clusters. Clustering in HIN attracts much attention recently since it gives insight of the structure of the network and may benefit other data mining tasks such as link prediction and ranking [9]. For example, in bibliographic heterogeneous information network such as DBLP [10], clustering authors shows the research field or latent co-author relationship among authors. In social network such as Facebook, clustering users reveals the social community or the latent interests of users. To facilitate clustering in large complex networks, it has been suggested that the user provide some supplementary information about the data (e.g. pairwise relationships between few data points), which when incorporated in the clustering process, could lead to a better data partition [11]. The side-information usually supplies by providing a constraint to the solution space [12–14] or learning a better distance metric in the network [15]. Such

[☆] One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment, either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.105458.

[*] Corresponding author.
E-mail addresses: zhousheng_zju@zju.edu.cn (S. Zhou), bjj@zju.edu.cn (J. Bu), zhen_zhang@zju.edu.cn (Z. Zhang), wcan@zju.edu.cn (C. Wang), mlz@zju.edu.cn (L. Ma), zjf2308@alibaba-inc.com (J. Zhang).

https://doi.org/10.1016/j.knosys.2019.105458
0950-7051/© 2020 Elsevier B.V. All rights reserved.

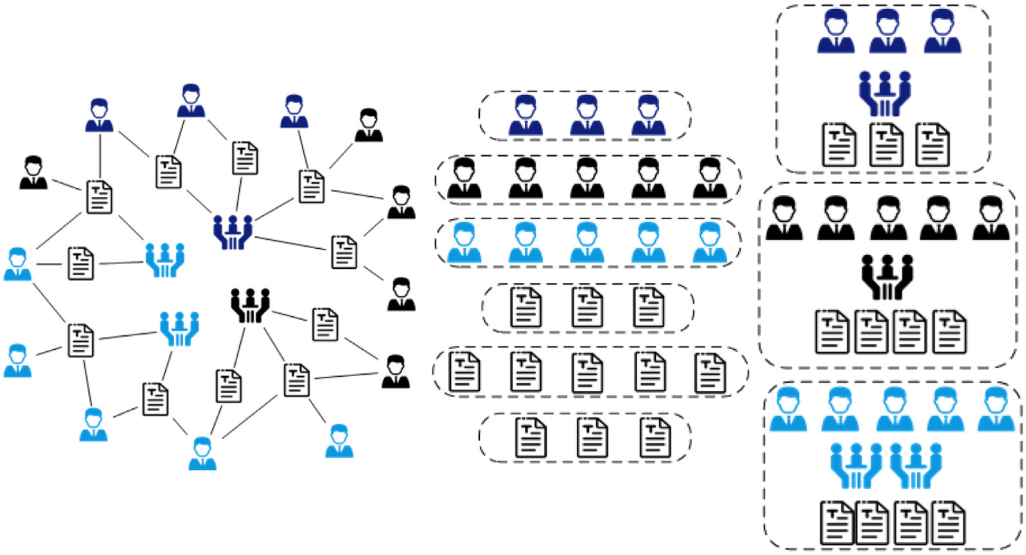

Fig. 1. An illustration of cross multi-type clustering in AHIN (attributes not shown). It contains three types of nodes including author, paper and conference. Traditional clustering methods focus on single type clustering (the middle subfigure) while our proposed method focus on multi type clustering (the right subfigure).

clustering are called semi-supervised clustering which has been widely studied in real-world data set.

However, most of the existing clustering methods in HIN or AHIN targets at one of the node-types, namely, the target type node and other node types are only used to help cluster the target node type, which means no clustering will be performed on these node types. Analyzing the outcomes of these clustering methods could only see the relationship among the target types of nodes while ignoring the whole picture of the HIN. In real-world HIN, different types of nodes may belong to one particular cluster, in other words, there may exist different types of nodes in one cluster. For example, in the bibliographic network, the authors, papers, conferences may belong to one cluster that represents one research topic. In social network, different types of social roles may belong to one user, such as jobs, bank accounts and social accounts from different social platforms. Finding such clusters could give us more insight of the relationships between different types of nodes and the latent representations of the clusters. An example of cross multi-type clustering in bibliographic attributed heterogeneous information network is illustrated in Fig. 1, where three types of nodes are contained: author, paper and conference. Traditional clustering methods focus on single type clustering (the middle subfigure) while our proposed method focus on multi type clustering(the right subfigure). Studying the relationship between all kinds of nodes could also iteratively improve the quality of clustering. For example, compared with clustering authors in DBLP data set, the cross multi-type clustering in DBLP data set could show how the conferences and papers are related to the research topic. This motivates the cross multi-type clustering in heterogeneous information network an interesting task in HIN or AHIN.

Although clustering different types of nodes are important, very few methods have been proposed for this purpose. The main challenges of cross multi-type clustering in HIN or AHIN network are as follows:

1. All the node types in HIN or AHIN need to be studied together in the same framework and enhance each other in clustering so that the whole HIN can be partitioned into clusters with all types of nodes.
2. The similarity measure for clustering should combine both the attribute information and network structure information. Since the cluster may contain different types of nodes, the measure should also be able to handle both same type and different types of nodes.
3. Given the side-information by users, label constraints would be constructed and clustering result should agree to the label constraints.

To address these challenges, only a few methods have been proposed to overcome partial challenges. For example, Aggarwal and Sun at al. [8,16,17] proposed to integrate the attribute information into clustering analysis on HIN. Deng et al. Deng et al. [18] proposed a joint probabilistic topic model for simultaneously modeling the contents of multi-typed objects of a HIN. However, to the best of our knowledge, there is not much previous work that explicitly investigates both in a unified framework. To bridge this gap, we propose a generic inference framework to integrate both the attribute information and multi-type data clustering in a principled way.

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

1. We propose a novel framework to cluster different types of nodes into clusters in heterogeneous information network. Similarity based on node attributes and network topology between nodes are learned in a unified framework.
2. An efficient EM-style updating algorithm is proposed to learn cluster assignment as well as parameters with respect to similarity. We provide time complexity analysis of the proposed method and existing methods.
3. We conduct extensive experiments on three real-world datasets to evaluate the effectiveness of the proposed method. We also summarize insights on the performances compared to other state-of-the-arts methodologies.

The rest of the paper is organized as follows. In Section 2, we briefly review the related work of clustering in heterogeneous information networks. In Section 3, we introduce the problem definition and the proposed *CMOC-AHIN* framework. In Section 4, we conduct experiment on two bibliographic networks and a very challenging and sparse real user behavior data set provided by a world leading E-commerce company. Finally, we conclude the paper in Section 5.

2. Related work

Most real systems usually consist of a large number of interacting, multi-typed components [19], such as human social activities, communications and computer systems, and biological networks. In such systems, the interacting components constitute interconnected networks, or information networks. The information network analysis, especially clustering analysis, has gained extremely wide attentions from academia as well as industry.

Traditional clustering methods, such as K-Means [20], Kmeoids [21] and Spectral Clustering [22] are based on features of related objects. Although they have been widely studied in the past decades, they fail to capture the relation-type data like edges in networks. Clustering based on network data (a.k.a community detection [23]) is generally an NP-hard problem and many methods have been proposed to model the data as a homogeneous network and cluster with some defined measures, e.g., normalized cut [24] and modularity [25], to divide the network into a series of subgraphs. However, the manual designed metrics highly depend on the network structure and cannot fit different networks. Some researchers also proposed to simultaneously model objects' link structure and attribute information [26,27] whose basic idea is to transfer the network structure into the feature representation of each node based on the connectivity or high-order proximities, then apply the traditional feature based clustering method. Other directions include spectral method [28], greedy method [29] and sampling technique [30–32].

Recently, with the development of deep learning framework, graph clustering based on deep neural networks has been attracting much attention. Among them, the network embedding methods [33] jointly learn node embedding as well as community embedding so that they can benefit each other. In VGAECD [34], they propose a deep generative model to take clustering into the prior of the generation process and utilize variational auto-encoder to learn the node embedding and clustering in a unified framework. In CommunityGAN [35], instead of studying the observed links, they try to generate and discriminate motifs (clique). The generator aims to generate (or select) subsets of vertices most likely to be real motifs and discriminator aims to estimate the probability that a vertex subset is a real motif.

In practice, partial true cluster labels may also be known and semi-supervised clustering methods can be adopted, such as COPKMEANS [20], PCKmeans [12] etc. Semi-supervised clustering methods in attributed network are also proposed to combine the network information and attribute information [36,37]. Although the methods mentioned above solve the clustering problems, they are designed only for homogeneous information networks which contains single type nodes and edges, it is hard to apply on heterogeneous information network with different types of nodes.

Compared with homogeneous networks, HIN and AHIN integrate multi-typed objects and attribute information which brings both challenges and opportunities. As a result, more and more researchers have noticed the importance of HIN and AHIN clustering and many novel data mining tasks have been exploited in such networks [38,38–40]. In RankClus [41], they proposed a method that utilizes links across multi-typed objects to generate high-quality net-clusters. An iterative enhancement algorithm is developed for effective ranking-based clustering. However, this method only works on a special HIN with star network schema without considering the node attributes as well. In HeteSim [42], a link based similarity measure (or HeteSim) is proposed for the similarity among different types of nodes. Li et al. [6] proposed a semi-supervised clustering method on the AHIN, in this framework, the target type nodes are studied by combining the attribute information and network structure information. However, only symmetric meta-path in the network are studied and this limit the extension to different types of nodes, also, only the target type of nodes is studied and the cluster only contains single node type. A few existing works focus on the multi-type nodes clustering in HIN. In CFRM [43], they proposed a general collective factorization on related matrices for multi-type relational data clustering. They do the simultaneous clustering for each type of objects respectively without including attributes of nodes, which is not multi-type clustering in essence. Zhou et al. [44] proposed a social influence based clustering framework SI-Cluster to analyze HIN based on social connections and activities. Alqadah, Zhou et al. [45] proposed a novel game theoretic framework for defining and mining clusters in HIN, the clustering problem is modeled as a game in which players attempt to maximize their reward, clusters are defined as the Nash equilibrium solution concepts. Other popular methods include RankClus [9], PathSelClus [46], GenClus [47], and they all cluster single type of nodes and do not consider their attributes.

To conclude, although clustering in HIN and AHIN has been studied in the literature, clustering different types of nodes in attributed heterogeneous information network has not been well studied, which motivates our proposed method.

3. The clustering model

In this section, we first provide some formal definition of the multi-type objects clustering in heterogeneous information network. Then we introduce our proposed *CMOC-AHIN* model by combining the attributes and meta path based node similarity, to learn the parameters of similarity as well as clustering results, we further propose an efficient EM-style update algorithm.

3.1. Definitions

Definition 1 (*Heterogeneous Information Network [48]*)**.** An information network is defined as a directed graph $G = (V, E)$ with an object type mapping function $\phi : V \rightarrow \mathcal{T}$ and a link type mapping function $\psi : E \rightarrow \mathcal{R}$. Each object $v \in V$ belongs to one particular object type in the object type set $\mathcal{T}$, and each link $e \in E$ belongs to a particular relation type in the relation type set $\mathcal{R}$. If two links belong to the same relation type, the two links share the same starting object type as well as the ending object type. The information network is called heterogeneous information network if the types of objects $|\mathcal{T}| > 1$ or the types of relations $|\mathcal{R}| > 1$; otherwise, it is a homogeneous information network.

Definition 2 (*Attributed Heterogeneous Information Network*)**.** [6] Some prior works have considered either the network structure/object linkage or the node attributes. The attributed heterogeneous information network (AHIN) is a graph defined as $G = (V, E, A)$. The definition of V and E are same as the heterogeneous information network and A is set of attributes of each node in the network. Note that different types of nodes have different types of attributes, and there are some overlapping attributes for different types of nodes. To make the attribute comparable among both same and different type nodes, we propose to unity multi-modal attributes in a consistent way in Section 3.2.1.

Definition 3 (*Network Schema [48]*)**.** The network schema $T_G = (\mathcal{T}, \mathcal{R})$ of an AHIN $G = (V, E, A)$ is the meta template of the network representing the relation between different types of nodes in the network. Given the mapping functions of HIN: the object type mapping $\phi : V \rightarrow \mathcal{T}$ that maps the node in V to its type $\mathcal{T}$ and the link mapping function $\psi : E \rightarrow \mathcal{R}$ that maps a link-relation in E into a relation in R, $T_G = (\mathcal{T}, \mathcal{R})$ can be represented by a schematic graph with $\mathcal{T}$ being the nodes and $\mathcal{R}$ being the edges.

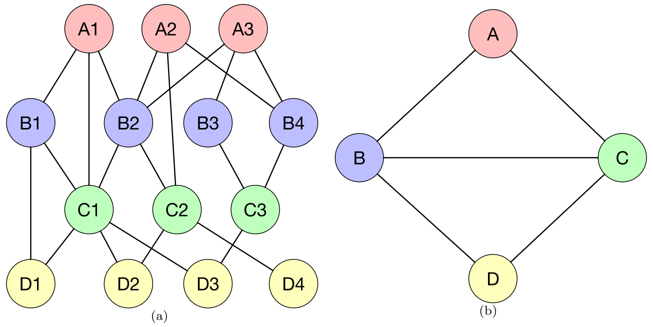

Fig. 2. An example of an attributed heterogeneous information network (attributes not shown) with four node types (a) and its network schema (b).

Network schema represents types of objects and linkage among nodes. Notice that there is an edge between type $\mathcal{T}_i$ and $\mathcal{T}_j$ if and only if there exist links that connect objects of these two types. Fig. 2 is an illustration of AHIN with four types of nodes $\mathcal{T} = \{A, B, C, D\}$ and its corresponding network schema.

Definition 4 (*Meta Path [48]*)**.** A meta-path $\mathcal{P}$ is a path defined on a network schema $\mathcal{T}_G = (\mathcal{T}, \mathcal{R})$ in the form of $T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} T_{l+1}$, which defines a composite relation $R = R_1 \circ R_2 \circ \dots R_l$ between type T_1 and T_{l+1}, where $\circ$ denotes the composition operator on relations. We say a path P is symmetric if the defined relation R is symmetric.

The set of meta-paths are selected by users from all the possible paths in HIN. Different from the widely studied symmetric meta-path, the meta-path $\mathcal{P}$ in our problem may be asymmetric especially when studying the relation between different types of nodes in the network.

Definition 5 (*Supervision Constraint*)**.** Partial true cluster labels may be known and we take this prior knowledge as the supervision constraint of our model. It is defined as $(\mathcal{M}, \mathcal{C})$, where $\mathcal{M}$ is the set of must-links and $\mathcal{C}$ is the set of cannot-links respectively. Links in $\mathcal{M}$ denote that related pairs belong to the same cluster and links in $\mathcal{C}$ denote that associated pairs cannot belong to the same cluster. In the cross multi-type clustering problem, $(\mathcal{M}, \mathcal{C})$ may include all node types instead of only single type nodes.

3.2. Cross **M**ulti-**T**ype **O**bjects Clustering in **A**ttributed **H**eterogeneous **I**nformation **N**etwork (CMOC-AHIN)

Cross Multi-type Objects Clustering in AHIN (or *CMOC-AHIN*) Given an AHIN $G = (V, E, A)$ and its network schema $T_G = (\mathcal{T}, \mathcal{R})$, a supervision constraint $(\mathcal{M}, \mathcal{C})$, a set of meta-path $P = \{P_1, P_2 \dots P_M\}$ and the number of clusters k, *CMOC-AHIN* aims to: (1) find an optimal similarity measure $S(v_i, v_j)$ between all kinds of nodes in the network based on both node attributes and meta-path. (2) Group all objects in the network into K clusters $\{C_k\}_{k=1}^K$ based on the similarity measure with the agreement of the constraint $(\mathcal{M}, \mathcal{C})$. We first define an *overall* similarity measure in Section 3.2.1 that combines *attributed* based and *meta-path* based similarities, both of which is a linear combination of different features. *CMOC-AHIN* finds the optimal clusters with the defined measure under the supervision constraints $(\mathcal{M}, \mathcal{C})$, where the latter is included in the penalty function in Section 3.2.4. An efficient optimization algorithm is proposed in Section 3.2.5 The notations used in the proposed *CMOC-AHIN*model is summarized in Table 1.

Table 1
Notations used in *CMOC-AHIN*.

Notation and Description	
α	Parameter for weighting the attribute based similarity and link based similarity
$\{C_k\}_{k=1}^K$	K clusters of the clustering results
v_i	The identifier of node i
$\boldsymbol{\lambda}$	Weights vector of meta path similarity
$\boldsymbol{\omega}$	Weights vector of attribute based similarity
S_f	Attribute based similarity measure
S_p	Path based similarity measure
S	Overall similarity of two nodes
$\boldsymbol{Z}$	Clustering result matrix denoting whether node belongs to one cluster
N	Number of nodes in AHIN
K	Number of clusters
μ_k	Center node of cluster k

3.2.1. Similarity measure

1. **Attribute based similarity measure:** Given two nodes v_i and v_j in the network, their associated attribute vectors are defined as x_i and x_j, both of which are vectors with length $|A|$. Note that we have mixed the attribute set of different types of nodes so that the attribute vectors are comparable between different node types. To capture the importance of each feature dimension, we define a weight vector $\boldsymbol{\omega} \in \mathbb{R}^{|A| \times 1}$, in which w_j measure the importance of the jth feature dimension. The attribute based similarity $S_a(v_i, v_j)$ of two nodes v_i and v_j is defined as:

$$S_a(v_i, v_j) = \sum_{k=1}^{|A|} \boldsymbol{\omega}_k \cdot \boldsymbol{S}_f(v_{ik}, v_{jk}), \tag{1}$$

where $\boldsymbol{S}_f(x_i, x_j)$ denotes a similarity measure between the kth attribute of node v_i and v_j. To make the similarity comparable, for numerical attributes, we normalize each attribute to range $[0, 1]$ and define $S_f(x_{ik}, x_{jk}) = 1 - |x_{ik} - x_{jk}|$; for categorical attributes, we let $S_f(x_{ik}, x_{jk}) = 1$ if $x_{ik} = x_{jk}$, and 0 otherwise.

2. **Path based similarity measure:** Given a relevant path $\mathcal{P} = T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} T_{l+1}$, define source node $v_s \in T_1$ and target node $v_t \in T_{l+1}$, we use HeteSim [42] to measure the similarity between v_s and v_t:

$$HeteSim(v_s, v_t | R_1 \circ R_2 \circ \dots \circ R_l) = \frac{1}{|O(v_s|R_1)||I(v_t|R_l)|} \sum_{i=1}^{O(v_s|R_1)} \sum_{j=1}^{I(v_t|R_l)} HeteSim(O_i(v_s|R_1), I_j(v_t|R_l)|R_2 \circ \dots \circ R_{l-1}), \tag{2}$$

where $O(v_s|R_1)$ is the set of out-neighbors of v_s based on relation R_1, and $I(v_s|R_l)$ is the set of in-neighbors of v_t based on relation R_l. When node v_s has no out-neighbor following the path $\mathcal{P}$ (i.e. $|O(v_s|R_1)| = 0$) or node v_t has no in-neighbor following the path $\mathcal{P}$ (i.e. $|O(v_t|R_l)| = 0$), $HeteSim(v_s, v_t | R_1 \circ R_2 \circ \dots \circ R_l)$ will be 0. Notice that HeteSim is a nested structure and can be evaluated iteratively until v_s and v_t meet in the middle of meta-path. The atomic relation R that in the middle of the meta path is defined as:

$$HeteSim(v_s, v_t | R) = HeteSim(v_s, v_t | R_o \circ R_I) = \frac{1}{|O(v_s|R_o)||I(v_t|R_I)|} \sum_{i=1}^{|O(v_s|R_o)|} \sum_{j=1}^{|I(v_t|R_I)|} \delta(O_i(v_s|R_o), I_j(v_t|R_I)), \tag{3}$$

where $\delta(v_s, v_t) = 1$ if v_s and v_t are the same node, else $\delta = 0$. To deal the asymmetric meta-path, the middle type nodes are added in the meta path. Given P meta-paths, we first define a weight vector $\boldsymbol{\lambda} = \{\lambda_m\}_{m=1}^{|P|} \in \mathbb{R}^{1\times|P|}$, where λ_m measures the importance of the mth meta-path. The path based similarity between any node pair ν_i and ν_j in the network is defined as:

$$S_p(\nu_i, \nu_j) = \sum_{m=1}^{|P|} \lambda_m HeteSim(\nu_i, \nu_j), \tag{4}$$

where λ_k is the weight of the kth corresponding meta-path.

3. **Overall similarity** The overall similarity of any node pairs in the network is defined as:

$$S(\nu_i, \nu_j) = \alpha S_a(\nu_i, \nu_j) + (1-\alpha) S_p(\nu_i, \nu_j) \tag{5}$$

where α is a hyper parameter as the weighting factor measuring the relative importance of attribute-based similarity and link-based similarity. It can be learned from the knowledge of partly labeled data.

3.2.2. Clustering learning

Given the comprehensive similarity measure between different types of nodes in heterogeneous information network, we are able to cluster different types of nodes into clusters. It is worth noting that many existing similarity/distance based clustering methods can be applied on our problem setting and here we discuss some representative methods for measuring the quality of clustering results.

1. **Centroid based clustering** This group of clustering methods measure the quality of clustering by the similarity/distance between nodes and corresponding cluster centers. The more similar between nodes and corresponding centers, the higher quality is the clustering results with. Given the similarity measurement S and cluster assignment Z, the centroid based clustering can be measured as:

$$\mathcal{J}_c = -\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} S(v_n, v_{\mu_k}) \tag{6}$$

where $\mathcal{J}_c$ is the loss function of centroid based clustering, $z_{nk} = 1$ is node v_n belongs to cluster k and $z_{nk} = 0$ otherwise. S is the comprehensive similarity measure, v_{μ_k} is the centroid of cluster k. The main advantage of the centroid based clustering is that it is easy to optimize with heuristic algorithm.

2. **Internal clustering methods** This group of clustering methods measure the quality of clustering by the similarity/distance between nodes from same cluster or different clusters. The more similar between nodes from same cluster or dissimilar between nodes from different clusters, the higher quality is the clustering results with. Given the similarity measurement S and cluster assignment Z, the internal clustering can be measured as:

$$\mathcal{J}_c = -\sum_{m=1}^{N}\sum_{n=1}^{N} \delta(m,n) S(v_m, v_n) = -\sum_{m=1}^{N}\sum_{n=1}^{N} z_m^T z_n S(v_m, v_n) \tag{7}$$

where $\mathcal{J}_c$ is the loss function of centroid based clustering, $\delta(m, n)$ denotes whether node v_m and v_n belong to same cluster, z_m, z_n is rigorous cluster assignment of node v_m, v_n which is a vector with $v_{mk} = 1$ and 0 for other dimensions. S is the comprehensive similarity measure. However, compared with centroid based clustering, optimizing the internal clustering methods has been proved [43] to be a NP-hard problem. According to the Ky-Fan theorem [49], the optimization has closed-form solution and we leave it in our future work.

3.2.3. Supervision constraints

Given some labeled data of the AHIN, the must-link set $\mathcal{M}$ is defined as the set of node pair that has the same label while the cannot-link set $\mathcal{C}$ is defined as the set of node pair that has different labels. To evaluate the similarity measure, we believe that two nodes in the must-link set $\mathcal{M}$ should be similar to each other while two nodes in the cannot-link set $\mathcal{C}$ should be dissimilar to each other. Formally, we define the similarities of nodes in cluster k as the sum of pairwise node similarities between its center μ_k. For the node pairs in the must-link set $\mathcal{M}$, they are expected to be linked to each other, which means the sum of similarity of these node pairs should be maximized; while for the node pairs in the cannot-link set $\mathcal{C}$, they are expected to be disentangled to each other, which means the similarity of these node pairs should be minimized.

3.2.4. Loss function

Taking this supervision constraint into consideration, the final loss function is defined as:

$$\begin{aligned}\mathcal{J} = &-\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} S(\nu_n, \nu_{\mu_k}) - \sum_{k=1}^{K}\sum_{(i,j)\in\mathcal{M}} S(\nu_i, \nu_j) \\ &+ \sum_{k=1}^{K}\sum_{(i,j)\in\mathcal{C}} S(\nu_i, \nu_j) + \gamma(\|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\omega}\|^2),\end{aligned} \tag{8}$$

where N is the number of nodes in AHIN, K is the total number of clusters, z_{nk} is the indicator of whether node ν_n belongs to cluster k, μ_k denotes the centroid object index of cluster k, $\mathcal{M}$ is the set of must-link and $\mathcal{C}$ is the set of cannot-link extracted from the labeled data, $\boldsymbol{\omega}$ is a M-dimensional vector referring to the weight importance in attribute based similarity, $\boldsymbol{\lambda}$ is a P-dimensional vector referring to the importance weights in meta-path based similarity. γ is the regularization term of λ and ω.

3.2.5. Model optimization

We aim to find the optimal clustering $\{C_k\}_{k=1}^{K}$ or the matrix $\boldsymbol{Z} \in \mathbb{R}^{N\times K}$ that minimizes the loss function $\mathcal{J}$. Notice that the loss function $\mathcal{J}$ is a function of $\boldsymbol{\lambda}, \boldsymbol{\omega}$ with respect to the weights of attribute based and meta-path based similarities. We propose the following EM-style iterative updating steps: in each iteration, given $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$, we first find the optimal clustering $\boldsymbol{Z}$. Second, given the clustering result $\boldsymbol{Z}$, we update $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$. The iteration will continue until the differences of the loss function between two iterations is convergent and below the predefined threshold ϵ.

1. **Updating $\boldsymbol{Z}$ given $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$** Given $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$, the cluster center μ_k from the last iteration, this step aims at finding the optimal clustering result under the constraint of $\mathcal{M}$ and $\mathcal{C}$. In this step, since $\boldsymbol{\lambda}$, $\boldsymbol{\omega}$, $\mathcal{M}$ and $\mathcal{C}$ are given, the second, third and last term in Eq. (8) are fixed values, then the loss function that needs to be minimized in this step is simplified to be:

$$\mathcal{J} = -\sum_{n=1}^{N}\sum_{c=1}^{K} z_{nc} S(\nu_n, \nu_{\mu_c}). \tag{9}$$

Notice that the optimal $\boldsymbol{Z}$ should also satisfy the constraint of $\mathcal{M}$ and $\mathcal{C}$, which means the node pairs in $\mathcal{M}$ should have same labels in $\boldsymbol{Z}$ and node pairs in $\mathcal{C}$ should have different

Algorithm 1 Finding the optimal $\boldsymbol{Z}$

```
Input: $G = \{V, E, A\}, K, \boldsymbol{\omega}, \boldsymbol{\lambda}, \boldsymbol{\mu}$
Output: $C = \{C_1, C_2 ... C_K\}, \boldsymbol{Z}$
  Compute attribute and meta-path similarity elements
  $t = 0, \triangle L = \infty$
  Init $\boldsymbol{\lambda}, \boldsymbol{\omega}, \boldsymbol{\mu}$
  while t < max_iter and $\triangle \mathcal{J} < \epsilon$ do
    STEP 1: Update the node labels $\boldsymbol{Z}$
    for $v_n \in V$ do
      for $k \in K$ do
        Measure overall similarity between $v_n$ and cluster center $\boldsymbol{\mu}_k$ by Eq. (5)
        Update $z_{nk}$ by Eq. (10)
    STEP 2: Update center of cluster
    Update the center of cluster $\mu_k$ by Eq. (11)
    STEP 3: t++
```

labels in $\boldsymbol{Z}$. We use a two-step iterative updating rules to find the optimal $\boldsymbol{Z}$.

The first step is to update the node label under the constraint of $\mathcal{M}$ and $\mathcal{C}$. Given the center of the cluster $\{\mu_1, \mu_2 ... \mu_K\}$, the similarity between each node and every cluster center is computed by Eq. (5) and the node adopts the label of its nearest center. However, if the node and a cluster center do exist in the must link set $\mathcal{M}$, the node should always adopt this cluster center label. If the node and a cluster center exist in the cannot link set $\mathcal{C}$, the similarity between them will be set to 0.

$$z_{nk} = \begin{cases} 1 & if\ k = \underset{k}{argmax}\ S(v_n, v_{\mu_k})\ or\ (v_n, v_{\mu_k})\ in\ \mathcal{M} \\ 0 & otherwise \end{cases} \tag{10}$$

The second step is to update the cluster center. Since the proposed measure is mixed of attributes and path, similarly to K-medoids [21], we define the center of cluster that has the largest sum of similarity to all the other nodes in the cluster as:

$$\mu_k = \underset{i}{argmax} \sum_{j=1}^{|C_k|} S(v_i, v_j). \tag{11}$$

The two step updating algorithm is summarized in Algorithm 1.

2. **Finding the optimal λ and ωLg given $\boldsymbol{Z}$** iven the clustering result z_{nc} and center of cluster μ_k, the loss function $\mathcal{J}$ is a function of $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$:

$$\mathcal{J} = -\sum_{n=1}^{N}\sum_{c=1}^{K} z_{nc} S(v_n, v_{\mu_c}) - \sum_{c=1}^{K}\sum_{(i,j)\in\mathcal{M}} S(v_i, v_j) + \sum_{c=1}^{K}\sum_{(i,j)\in\mathcal{C}} S(v_i, v_j) + \gamma(\|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\omega}\|^2) \tag{12}$$

where

$$S(v_i, v_j) = \alpha[\sum_{k=1}^{|A|} \boldsymbol{\omega}_k \cdot \boldsymbol{S}_f(v_{ik}, v_{jk})] + (1-\alpha)[\sum_{m=1}^{|P|} \lambda_m HeteSim(v_i, v_j)] \tag{13}$$

Note that the first term calculates the similarity between each node and their assigned cluster center, the second term calculates the similarity of nodes in the must-link set, the third term calculate the similarity of nodes in the cannot-link set. All of the above terms select parts of node pairs from the network, and we define three selection matrix $\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3 \in R^{N\times N}$: in each matrix $\boldsymbol{T}$, $\boldsymbol{T}_{i,j} = 1$ if the node pair (i, j) is selected, otherwise $\boldsymbol{T}_{i,j} = 0$. The similarity matrix between any node pair in the network is defined as matrix $\boldsymbol{S}$, where $\boldsymbol{S}_{i,j}$ is the overall similarity between node v_i and v_j. Then the loss function can be rewritten as:

$$\begin{aligned} \mathcal{J}(\boldsymbol{\lambda}, \boldsymbol{\omega}) = & -\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{T}_{1i,j}\boldsymbol{S}_{i,j} - \sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{T}_{2i,j}\boldsymbol{S}_{i,j} + \sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{T}_{3i,j}\boldsymbol{S}_{i,j} \\ & + \gamma(\|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\omega}\|^2) \\ = & \sum_{i=1}^{N}\sum_{j=1}^{N}(-\boldsymbol{T}_1 - \boldsymbol{T}_2 + \boldsymbol{T}_3)_{i,j}\boldsymbol{S}_{i,j} + \gamma(\|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\omega}\|^2) \\ = & \sum_{i=1}^{N}\sum_{j=1}^{N}(-\boldsymbol{T}_1 - \boldsymbol{T}_2 + \boldsymbol{T}_3)_{i,j}[\alpha \sum_{k=1}^{M} \omega_k \cdot S_f x_{ik}, x_{jk} \\ & + (1-\alpha)\sum_{k=1}^{P} \lambda_k S_{Hk}(v_i, v_j)] + \gamma(\|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\omega}\|^2) \\ = & \sum_{i=1}^{N}\sum_{j=1}^{N}(-\boldsymbol{T}_1 - \boldsymbol{T}_2 + \boldsymbol{T}_3)_{i,j}[\alpha\boldsymbol{\omega}\boldsymbol{S}_{fij}{}^{T} + (1-\alpha)\boldsymbol{\lambda}\boldsymbol{S}_{Hij}{}^{T}] \\ & + \gamma(\|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\omega}\|^2) \end{aligned} \tag{14}$$

Note that the $\boldsymbol{\omega}$ and $\boldsymbol{\lambda}$ are uncoupled and solving such a optimization problem is a classic quadratic programming problem with an analytical solution:

$$\begin{gathered} \frac{\mathrm{d}\mathcal{J}}{\boldsymbol{\omega}} = 0 \\ \sum_{i=1}^{N}\sum_{j=1}^{N}(-\boldsymbol{T}_1 - \boldsymbol{T}_2 + \boldsymbol{T}_3)_{i,j}\alpha\boldsymbol{S}_{fij} + 2\gamma\omega = 0 \\ \omega = -\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(-\boldsymbol{T}_1 - \boldsymbol{T}_2 + \boldsymbol{T}_3)_{i,j}\alpha\boldsymbol{S}_{fij}}{2\gamma} \end{gathered} \tag{15}$$

$$\begin{gathered} \frac{\mathrm{d}\mathcal{J}}{\boldsymbol{\lambda}} = 0 \\ \sum_{i=1}^{N}\sum_{j=1}^{N}(-\boldsymbol{T}_1 - \boldsymbol{T}_2 + \boldsymbol{T}_3)_{i,j}(1-\alpha)\boldsymbol{S}_{Hij} + 2\gamma\lambda = 0 \\ \lambda = -\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(-\boldsymbol{T}_1 - \boldsymbol{T}_2 + \boldsymbol{T}_3)_{i,j}(1-\alpha)\boldsymbol{S}_{Hij}}{2\gamma} \end{gathered} \tag{16}$$

The algorithm is summarized in Algorithm 2.

3.2.6. *Time complexity analysis*

The iteration process of *CMOC-AHIN* contains two steps, the first step is updating $\boldsymbol{Z}$ given $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$, the second step is updating $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ based on the optimal $\boldsymbol{Z}$ from last step. Notice that, HeteSim of all meta-paths are not changed during each iteration, thus it can be pre-calculated and serve as input. For the second step, we inferred an analytical solution and the time complexity is at most $\boldsymbol{O}(\boldsymbol{N})$, where N is the number of nodes. As we discussed in Algorithm 1, STEP 1 is also an iterative updating process. In each iteration of step 1, *CMOC-AHIN* first finds the optimal $\boldsymbol{Z}$ given $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$, in this step, the similarity between each node and the center of cluster is computed with complexity of $\boldsymbol{O}(\boldsymbol{NK})$. Next, the

Algorithm 2 *CMOC-AHIN*

```
Input: $G = \{V, E, A\}, V_l, K$
Output: $C = \{C_1, C_2 ... C_K\}$
  Compute attribute and meta-path similarity elements
  $t = 0, \triangle L = \infty$
  Init $\lambda, \omega$
  while t < max_iter and $\triangle \mathcal{J} < \epsilon$ do
      STEP 1: Optimize $\{z_r\}_{r=1}^{k}$ given $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$
      repeat
          for $v_n \in V$ do
              for $k \in K$ do
                  Measure overall similarity between $v_n$ and cluster center $\boldsymbol{\mu}_k$ by Eq. (5)
              Update $z_{nk}$ by Eq. (10)
          for $\mu_k$ in $\boldsymbol{\mu}$ do
              Update the center of cluster $\mu_k$ by Eq. (11)
      until convergence
      STEP 2: Optimize $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ given $\{z_r\}_{r=1}^{K}$
      Solve equation (12) and (13) obtain the optimal $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$
      STEP 3: t++
```

similarity between each node in the same clusters are computed for all the clusters with complexity of $\boldsymbol{O}(\boldsymbol{K}(\boldsymbol{N}-\boldsymbol{K})^{\boldsymbol{2}})$, thus the computational complexity for each iteration is $\boldsymbol{O}(\boldsymbol{NK}+\boldsymbol{K}(\boldsymbol{N}-\boldsymbol{K})^2)$. The total time complexity for *CMOC-AHIN* is $\boldsymbol{O}(\boldsymbol{NK}+\boldsymbol{K}(\boldsymbol{N}-\boldsymbol{K})^2)$. We omit the number of iterations I_1 and I_2 in the final time complexity since they are usually smaller than 10 and far smaller compared to the number of nodes N or the clusters K.

For the comparison methods used in our experimental section, the time complexity for K-medoid based methods is same as our proposed method $\boldsymbol{O}(\boldsymbol{NK}+\boldsymbol{K}(\boldsymbol{N}-\boldsymbol{K})^2)$ which also contains the similarity measurement and clustering update. For the spectral clustering based method, the time complexity is $\boldsymbol{O}(\boldsymbol{N}^3)$ which is affected by calculating the Laplacian eigenvalues. The time complexity for FocusCO algorithm is $O(\boldsymbol{M}(log\boldsymbol{K}+|\boldsymbol{S}|))$ where $|\boldsymbol{S}|$ is the size of the focused cluster, $\boldsymbol{M}$ is the number of edges and $\boldsymbol{K}$ is the number of edges to find the core set. Compare with the baseline methods, although our proposed method is not most sufficient, the main advantage over the existing methods is that our method is designed for heterogeneous information network with different types of nodes.

4. Experimental evaluation

In this section, we empirically show superior performances of *CMOC-AHIN* on three challenging data sets compared with other state-of-the-arts methodologies. We also test *CMOC-AHIN* with two other variations: attribute-based alone and link-based alone similarities and empirically show that *overall* similarity as proposed in Eq. (5) works. To the best of our knowledge, most of the current existing methods focus on single type nodes and there is very limited existing work that groups objects of different types into the same cluster.

4.1. Data sets

We mainly focus on three datasets unified Internet ID (UID), Aminer and DBLP.

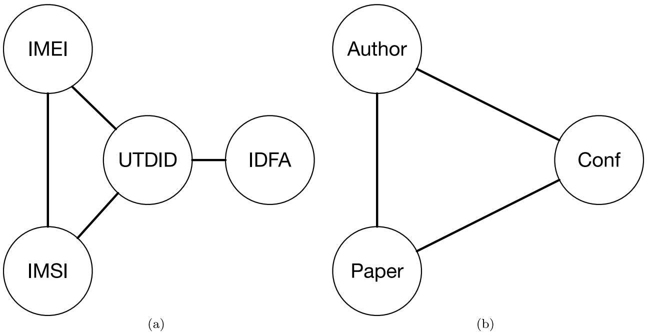

Fig. 3. Network Schema of the dataset used in our experiment. Subfigure (a) denotes the UID dataset, subfigure (b) denotes the Aminer and DBLP dataset.

1. **UID**: We first run experiments on a real-world user behavior network data set UID. In UID, there are four types of device IDs, including IMEI (International Mobile Equipment Identity), IMSI (International Mobile Subscriber Identity), UTDID (or app identifiers) and IDFA (Identifier for advertisers, Apple's alternative to HTTP cookies on iOS), which are different identifiers for physical devices respectively. For each device, one or several co-occurrence records of these device IDs can be collected, e.g., there are at most 3 types of device IDs for Android phones and usually 2 types of device IDs for IOS devices. The network schema is shown in Fig. 3(a). The purpose of cross multi-type clustering in UID dataset is to identify different types of nodes that each physical device contains. Also, the label of some of the identifiers are already known from the supervision information. We run experiments on an extracted data set which includes 5000 physical devices with 6334 IMEI nodes, 9463 IMSI nodes, 5551 UTDID nodes and 53 IDFA nodes. The links are collected from the user behavior records, and each record may contain one or more device ids and the link types include IMEI-IMSI, IMEI-UTDID, IMSI-UTDID and UTDID-IDFA. From these links we define meta-paths as: IMEI-IMSI-IMEI, IMSI-IMEI-IMSI, IMSI-UTDID-IDFA. The attributes of these nodes are extracted from the user behavior records and include IP address, mac address, resolution of physical device, device model and locations, which are concatenated as a feature vector for each node.
2. **Aminer DataSet**: Aminer [50] is a real-world bibliographic data set that contains five research domains, including data mining, medical informatics, theory, visualization and database. Each research domain contains conferences, paper and authors. We extract its conference, paper, author and abstract by randomly sampling authors who publish more than two papers with their published paper and corresponding conferences. In total, the extracted dataset contains 22 conferences (C), 1709 authors (A) and 1000 papers (P). Then we build an AHIN of the bibliographic data set for authors, papers and conferences. The network schema is shown in Fig. 3(b). For each node in the network, we use TF-IDF [51] to extract the top-*k* words as its attributes. For example, the attributes of an author is the top-*k* words of his/her publications and the attribute of a conference is the top-*k* words of the papers that have been published in this specific conference. The links in the network are set up with the meta-relationship between them, e.g., the authorship of a paper, and co-author relationship between authors, etc. Thus we define several meta-paths with semantic meanings: for example A-P-A represents the co-author relationship between two authors, P-A-P shows that one author writes two papers, A-P-A-P shows the relationship between one author and one paper connected by the author's own paper and his/her co-authorship.

3. **DBLP dataset**: DBLP is another bibliographic data set that contains conference, paper and authors. We use a subset dataset of DBLP names 'four area dataset' [41] which contains 20 major conferences and all the related papers, authors and terms in data mining (DM), database (DB), information retrieval (IR), and machine learning (ML) fields, according to the research areas of the conferences. We build an AHIN of the dataset where nodes include 20 conferences (C), 4827 authors (A) and 2000 papers (P). We use the terms of paper as attributes, for authors and conference, we use the terms of connected paper as the attributes. The label of each node is the research area it belongs to. We also define several meta-paths with semantic meanings: for example A-P-A represents the co-authorship between two authors and A-P-C-P-A denotes the relationship between authors that publish paper on same conference. The network schema is same as Aminer dataset and illustrated in Fig. 3(b).

4.2. Baselines

We compared our algorithm with several other state-of-the-arts on the task of cross multi-type nodes clustering in AHIN, which can be categorized into the following 4 semi-supervised groups:

1. **Attribute-only:** This group of clustering algorithms are traditional methods that only take the node attributes into consideration, while ignoring the network structure of an AHIN. Seeded-KMedoids [52] is a semi-supervised variants of KMedoids that uses labeled data to generate initial seed clusters, as well as always keeping them in the initial clusters during the updating procedure. Constrained Spectral Clustering (abbreviated as Con-SC) [53] integrates must-link and cannot-link constraints into spectral clustering framework. Both of these methods do not learn the weights of each attribute, and we assign equal weights to all the attributes when constructing the similarity matrix.
2. **Link-only:**This group of clustering algorithms only utilize the link information in an AHIN, discarding the attribute values. Link-based similarity measures on AHIN take not only multi-typed objects but also meta path connecting these objects into consideration. We consider two clustering algorithms based on meta path similarity measure: HeteSim-KMedoids and HeteSim-Spectral Clustering (abbreviated as HeteSim-SC), both of which take the meta path similarity matrix as inputs. The similarity matrix is constructed as follows: we use partially ground truth and train a logistic regression model to learn the weights of each pre-defined meta path. We then assign the other meta path weights with the learnt model. Notice that, if two nodes cannot reach each other through the pre-defined meta path, we define their similarity as 0.
3. **Attribute and link:** This group of clustering algorithms consider both attribute and link information. There exists some previous working on combining the attribute information and network structure information in HIN [6], but the limit of symmetric paths makes it hard to be compared here. FocusCO [36] is a novel user-oriented method for mining attributed graphs in homogeneous information network. This approach first infers user preference by a set of user-provided exemplar nodes through metric learning. Then the algorithm modifies the weights of each link according to the weighted similarity of its node attributes. Next, core sets are extracted as clusters of interest. However, it can only be applied to homogeneous networks. In this situation, we treat our AHIN as an attributed graph, neglecting the type of objects and links. Another compared algorithm in this group is based on COP-Kmeans and Auto-encoder, autoencoder has been a widely used approach to learn meaningful low-dimensional representation. We first learn low-dimensional representation with autoencoder and then fed it into the COP-Kmeans to learn the clustering results.
4. ***CMOC-AHIN* variants:** We also test *CMOC-AHIN* with two other variations: attribute-based alone (denoted as A-*CMOC-AHIN*) and path-based alone similarities (denoted as P-*CMOC-AHIN*) to show the advantage of combining the attribute information and network structure information. The parameter setting of both *CMOC-AHIN* variants are same as *CMOC-AHIN* to make the comparison fair enough.

4.3. Evaluation metric

The evaluation metrics in our experiment are two popular metrics in clustering: Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). Both of which are external measures adapted from information retrieval, which compare the data partition obtained from the clustering algorithm with the true class labels. Normalized Mutual Information (NMI) [54] can be defined as:

$$NMI(X,Y) = \frac{\sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}}{H(X)+H(Y)} \tag{17}$$

where X is the clustering result of evaluated algorithm and Y is the clustering result of ground truth, $p(x,y)$ is the joint probability distribution function of X and Y, $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively, $H(X)$ and $H(Y)$ are the marginal entropies. NMI is between 0 and 1, and the higher the NMI, the better the quality of the clustering. If $NMI = 1$, the clustering result perfectly agrees with the ground truth. ARI [55] is defined as:

$$RI = \frac{N_A}{N_A + N_D} \quad ARI = \frac{RI - \overline{RI}}{RI_{max} - \overline{RI}} \tag{18}$$

where N_A is the number of agreements between two clustering results and N_D is the number of disagreements between two clustering results. $\overline{RI}$ denotes the expected rand index and RI_{max} denotes the max rand index. The agreement or disagreement denotes whether two nodes has the same relationship, e.g., in the same cluster or in different clusters, for two clustering results. And again, the higher of ARI, the better of the clustering results.

4.4. Results and analyses

This section summarizes the clustering results of the 4 differed groups of the semi-supervised clustering methods. Given the input data, we first construct the supervision constraint $(\mathcal{M}, \mathcal{C})$ by randomly picking a certain percentage of nodes from the ground truth, for each pair of nodes (v_i, v_j), if the label of two nodes are the same, we add the node pair to the must-link set $\mathcal{M}$, else, we add it to the $\mathcal{C}$. Each of the two evaluation metric is the average of 20 runs and in each run, the supervision constraints are randomly picked up from the ground truth.

4.4.1. Clustering results

The clustering result on all three experimental sets are shown in Tables 2–7.

The first row of each table represents the percentage of nodes we used to generate the supervision constraint $\boldsymbol{M}$ and $\boldsymbol{C}$. In each row, the best result of the evaluation metric is highlighted with blue color. We get the result of NMI and ARI in the table

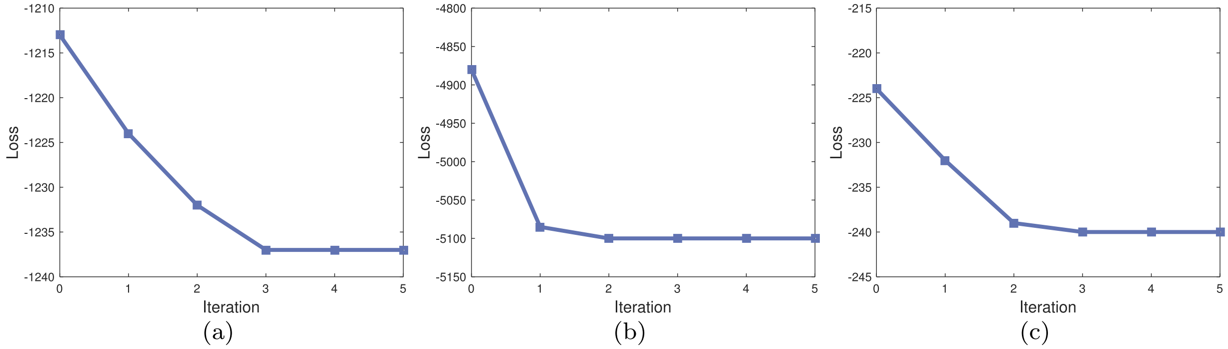

Fig. 4. Convergence analysis on the experimental data sets. Subfigure (a) denotes the UID data set, subfigure (b) denotes the Aminer data set and subfigure (c) denotes the DBLP dataset.

Table 2
Comparison of NMI on Aminer dataset. We use **blue** to highlight wins and **red** to highlight lose.

Model/seeds	5%	10%	15%	20%	25%
Seeded-Kmedoids	0.432	0.608	0.634	0.653	0.679
Constrained-SC	0.247	0.238	0.263	0.363	0.433
HeteSim-Kmedoids	0.642	0.686	0.758	0.788	0.819
HeteSim-SC	0.593	0.65	0.708	0.757	0.772
FocusCO	0.108	0.108	**0.108**	**0.108**	**0.108**
AE-COPKmeans	0.636	0.671	0.703	0.727	0.759
A-*CMOC-AHIN*	**0.080**	**0.102**	0.115	0.133	0.124
P-*CMOC-AHIN*	0.759	0.825	0.761	0.822	0.861
CMOC-AHIN	**0.846***	**0.875***	**0.874***	**0.882***	**0.906***

*Indicates that the improvement is significant with t-test at $p < 0.05$.

Table 3
Comparison of ARI on Aminer dataset. We use **blue** to highlight wins and **red** to highlight lose.

Model/seeds	5%	10%	15%	20%	25%
Seeded-Kmedoids	0.415	0.617	0.644	0.666	0.640
Constrained-SC	0.110	0.116	0.147	0.259	0.290
HeteSim-Kmedoids	0.414	0.549	0.598	0.667	0.747
HeteSim-SC	0.383	0.565	0.624	0.764	0.721
FocusCO	0.028	0.028	0.028	0.028	0.028
AE-COPKmeans	0.477	0.491	0.519	0.539	0.572
A-*CMOC-AHIN*	**0.002**	**0.005**	**0.008**	**0.012**	**0.012**
P-*CMOC-AHIN*	0.625	0.733	0.643	0.724	0.783
CMOC-AHIN	**0.762***	**0.802***	**0.799***	**0.813***	**0.854***

*Indicates that the improvement is significant with t-test at $p < 0.05$.

Table 4
Comparison of NMI on UID dataset. We use **blue** to highlight wins and **red** to highlight lose.

Model/seeds	5%	10%	15%	20%	25%
Seeded-Kmedoids	0.751	0.756	0.767	0.78	0.791
Constrained-SC	0.621	0.688	0.725	0.702	0.79
HeteSim-Kmedoids	0.752	0.741	0.739	0.755	0.753
HeteSim-SC	0.878	0.879	0.876	0.873	0.877
FocusCO	**0.277**	**0.278**	**0.28**	**0.277**	**0.276**
AE-COPKmeans	0.819	0.830	0.841	0.855	0.869
A-*CMOC-AHIN*	0.883	0.884	0.885	0.887	0.888
P-*CMOC-AHIN*	0.751	0.751	0.753	0.754	0.756
CMOC-AHIN	**0.917***	**0.918***	**0.918***	**0.920***	**0.922***

*Indicates that the improvement is significant with t-test at $p < 0.05$.

Table 5
Comparison of ARI on UID dataset. We use **blue** to highlight wins and **red** to highlight lose.

Model/seeds	5%	10%	15%	20%	25%
Seeded-Kmedoids	0.039	0.040	0.050	0.061	0.066
Constrained-SC	0.005	0.020	0.030	0.013	0.063
HeteSim-Kmedoids	0.005	0.004	0.004	0.005	0.005
HeteSim-SC	0.024	0.025	0.023	0.022	0.023
FocusCO	**0.0005**	**0.0009**	**0.0009**	**0.0009**	**0.0009**
AE-COPKmeans	0.021	0.043	0.058	0.070	0.086
A-*CMOC-AHIN*	0.365	0.370	0.374	0.382	0.388
P-*CMOC-AHIN*	0.004	0.004	0.005	0.005	0.006
CMOC-AHIN	**0.469***	**0.471***	**0.473***	**0.480***	**0.485***

*Indicates that the improvement is significant with t-test at $p < 0.05$.

Table 6
Comparison of NMI on DBLP datset. We use **blue** to highlight wins and **red** to highlight lose.

Model/seeds	5%	10%	15%	20%	25%
Seeded-Kmedoids	0.045	0.069	0.081	0.103	0.138
Constrained-SC	0.033	0.047	0.063	0.081	0.101
HeteSim-Kmedoids	0.367	0.382	0.399	0.411	0.436
HeteSim-SC	0.291	0.304	0.314	0.337	0.359
FocusCO	0.371	0.373	0.372	0.370	0.373
AE-COPKmeans	0.386	0.401	0.428	0.448	0.469
A-*CMOC-AHIN*	**0.062**	**0.098**	**0.112**	**0.169**	**0.224**
P-*CMOC-AHIN*	0.538	0.549	0.589	0.598	0.632
CMOC-AHIN	**0.559***	**0.596***	**0.612***	**0.632***	**0.652***

*Indicates that the improvement is significant with t-test at $p < 0.05$.

Table 7
Comparison of ARI on DBLP dataset. We use **blue** to highlight wins and **red** to highlight lose.

Model/seeds	5%	10%	15%	20%	25%
Seeded-Kmedoids	0.059	0.078	0.094	0.117	0.148
Constrained-SC	0.041	0.049	0.067	0.090	0.106
HeteSim-Kmedoids	0.398	0.410	0.431	0.455	0.480
HeteSim-SC	0.301	0.317	0.329	0.344	0.363
FocusCO	0.395	0.395	0.397	0.399	0.399
AE-COPKmeans	0.423	0.441	0.460	0.489	0.501
A-*CMOC-AHIN*	**0.059**	**0.103**	**0.112**	**0.179**	**0.245**
P-*CMOC-AHIN*	0.5861	0.6089	0.6497	0.6581	0.6884
CMOC-AHIN	**0.625***	**0.662***	**0.683***	**0.698***	**0.713***

*Indicates that the improvement is significant with t-test at $p < 0.05$.

by randomly perform 20 independent runs on the data set and output the mean of different runs as our final result. We make the following observation based on the tables of results:

1. Overall clustering quality comparison. From the tables of clustering performance, we can see that our algorithm *CMOC-AHIN* outperforms the compared algorithm of attribute-only and link-only methods on both Aminer,

DBLP data set and the UID data set. Also, compared with the *CMOC-AHIN* variants methods, the quality of clustering is also improved, all of these shows the advantage of our algorithm using an iterative learning process to combine the attribute information and link information.

2. Supervision Constraint Analysis. From each line of the table, we can see that the quality of clustering gets better with the growing number of seed nodes. This shows that taking the supervision into consideration could help improve the quality of clustering. Notice that some method such as FocusCO is not sensitive with the seeded nodes, since it take the supervision information by learning the weights from these nodes and the growing number of nodes cannot guarantee the improvement of the clustering quality.
3. Link VS Attribute in different datasets. According to the comparison between two *CMOC-AHIN* variants in different data sets, we can see that in different data sets, the network structure information and the attribute information play different roles. In the Aminer dataset, the path based method P-*CMOC-AHIN* outperforms the attribute based method A-*CMOC-AHIN*, this could be explained that in Aminer dataset, the attributes are the term frequency extracted from the papers which has been described in Section 4.1. the difference of attributes between node types may be misty. In the UID data set, the result is opposite, the attribute based method A-*CMOC-AHIN* outperforms the path based method P-*CMOC-AHIN*. This is reasonable since in this data set, the attributes of nodes are more diverse, the attribute based method could get superiority in such data sets, also the path in this data set are from the user behavior records which makes it hard to use meta-paths to find out the nodes in one cluster while does not co-occur in the records.
4. Homogeneous VS Heterogeneous. Notice that FocusCO performs poorly on both Aminer data set and UID data set, although FocusCO is a local graph partitioning method that combines attribute information and link information, it is a homogeneous network method and cannot work well on the heterogeneous information network. When applying FocusCO on heterogeneous information network, we have to simplify the network into a homogeneous network which means the difference between node type and the meta-path information are all missed. We believe that this is the weakness of applying the homogeneous network method on the heterogeneous information network.
5. Notice that in Table 5, P-*CMOC-AHIN* obtained rather poor ARI on experimenting with UID. One possible reason is that the UID data set is rather sparse with too many clusters but very limited paths.

4.4.2. Meta path selection

In Table 8, we show the top-5 meta-path that with largest λ in three experimental data set. By analyzing the semantic meaning of the meta-path, we have a better understanding of how *CMOC-AHIN* picked up these meta-paths.

In the Aminer data set, the first meta-path $A - C - A$ shows the relationship between two authors that publish paper in the same conference, since 'Author' node has the largest number in the network, the good clustering result on the authors could lead a better performance on clustering. The second and fourth meta-path shows the relativeness between author and paper, according to 'Conference' node, one author may have close relation to another paper in the conference where the author published paper. The third and fifth meta-path shows the similarity between two paper, the results show that two paper published in the same conference may have a higher similarity. Similarity results can be observed in the DBLP data set which is also a bibliographic network. The above observations show that *CMOC-AHIN* can select the meaningful meta-path from the heterogeneous information. In the UID data set, as we have discussed in the last section, the meta-path suffers from the fact that if two nodes did not co occur in one user behavior record, it would be hard to use meta-path to assign them into same cluster. In this data set, the attributes are playing a more important role.

4.4.3. Convergence analysis

Since *CMOC-AHIN* is an iterative learning process, we show the convergence of loss $\mathcal{J}$ in both Aminer data set and UID data set in Fig. 4. We test 20 random runs on different data sets and show the mean of evaluation metrics. The result shows that in all three datasets we used, our algorithm *CMOC-AHIN* could converge quickly.

4.5. Running time comparison

In this subsection, we compare the running time of our proposed method *CMOC-AHIN* and baseline methods on three HIN datasets. Fig. 5 illustrates the running time results. According to the results, we observe that Spectral Clustering has the worst time complexity since it calculates the eigenvector which is time consuming. Compared with Kmedoids method, the proposed *CMOC-AHIN* method spends little more time as *CMOC-AHIN* combines both attribute information and network topology. Among the compared methods, FocusCO is most efficient method while the performance is not as good as other baseline methods. This is explainable since it ignores the heterogeneous network structure.

5. Conclusions and future work

In this paper, we introduce a novel and practical model to study the problem of cross multi-type clustering in heterogeneous information network, namely *CMOC-AHIN*. Given the attributed network information and some semi-supervised constraints, *CMOC-AHIN* combines node attributes and meta-path information in a constrained way. With an iterative learning process, *CMOC-AHIN* learns the optimal parameters as well as clustering results. To empirically show the superiority of the mixed edge information and learning process, we conduct several experiments on the real-world data. The experiment results on the real-world data show that our algorithm outperforms the existing algorithms that use attribute or meta-path information for clustering in heterogeneous information network. We also test the variants of *CMOC-AHIN* and the results reveal that the consideration of both the path and attribute information is meaningful.

This paper also suggests some potential research directions. First, heterogeneous information network with more complex types of attributes should be discussed especially those contains both binary and continuous attributes. Second, the knowledge graph can be treated as a more complex heterogeneous information which contains more types of entities and relations. Last, recent advances of deep clustering which utilizes deep learning methods could also be explored on heterogeneous information network to cluster different types of nodes.

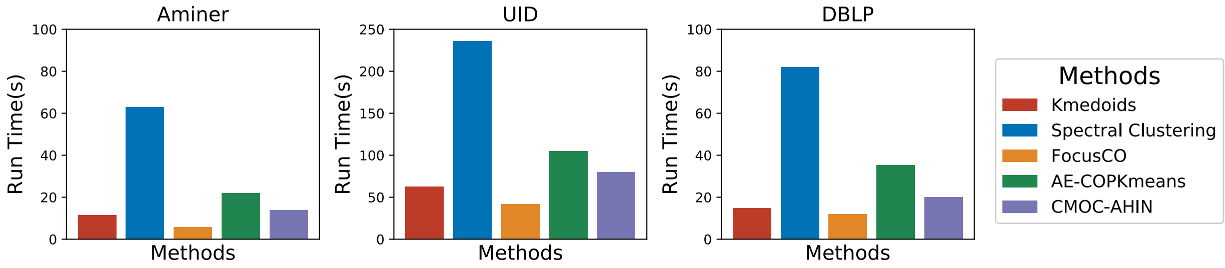

Fig. 5. Running time comparison on real-world datasets.

Table 8
Top-5 meta-path in three real-world datasets.

Rank	DBLP	Aminer	UID
1	Conf-Paper-Conf	Author-Conf-Author	IMEI-UTDID-IMSI
2	Author-Paper	Author-Paper-Conf	IMSI-UTDID-IMEI
3	Author-Conf-Author	Paper-Conf-Paper	IMEI-IMSI-UTDID
4	Author-Paper-Conf	Paper-Author-Conf-Author	UTDID-IMSI-IMEI
5	Author-Paper-Conf-Paper-Author	Paper-Author-Conf-Paper	IMEI-IMSI-IMEI-IMSI

CRediT authorship contribution statement

Sheng Zhou: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft. **Jiajun Bu:** Investigation, Writing - review & editing, Supervision, Funding acquisition, Project administration. **Zhen Zhang:** Software, Validation. **Can Wang:** Investigation, Data curation, Writing - review & editing, Supervision, Project administration. **Lingzhou Ma:** Project administration, Funding acquisition. **Jianfeng Zhang:** Project administration, Funding acquisition.

Acknowledgment

This work is supported by Alibaba-Zhejiang University Joint Institute of Frontier Technologies, National Natural Science Foundation of China (Grant No: U1866602), National Key Research and Development Project (Grant No: 2018AAA0101503, 2019YFB1600 700), the National Key R&D Program of China (No. 2018YFC2002 603, 2018YFB1403202), Zejiang Provincial Natural Science Foundation of China (No. LZ13F020001), the National Natural Science Foundation of China (No. 61972349, 61173185, 61173186) and the National Key Technology R&D Program of China (No. 2012BAI34B01, 2014BAK15B02). National Natural Science Foundation of China (Grant No: U1866602) and National Key Research and Development Project (Grant No: 2018AAA0101503).

References

[1] Y. Sun, J. Han, Mining heterogeneous information networks: A structural analysis approach, SIGKDD Explor. Newsl. 14 (2) (2013) 20–28.
[2] R.N. Lichtenwalter, J.T. Lussier, N.V. Chawla, New perspectives and methods in link prediction, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, ACM, New York, NY, USA, 2010, pp. 243–252.
[3] V. Leroy, B.B. Cambazoglu, F. Bonchi, Cold start link prediction, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, ACM, New York, NY, USA, 2010, pp. 393–402.
[4] C. Shi, X. Kong, P.S. Yu, S. Xie, B. Wu, Relevance search in heterogeneous networks, in: Proceedings of the 15th International Conference on Extending Database Technology, EDBT '12, ACM, New York, NY, USA, 2012, pp. 180–191.
[5] Y. Sun, J. Han, X. Yan, P.S. Yu, T. Wu, PathSim: Meta path-based top-k similarity search in heterogeneous information networks, Proc. VLDB 4 (11) (2011) 992–1003.
[6] X. Li, Y. Wu, M. Ester, B. Kao, X. Wang, Y. Zheng, Semi-supervised clustering in attributed heterogeneous information networks, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2017, pp. 1621–1629.
[7] C. Wan, X. Li, B. Kao, X. Yu, Q. Gu, D. Cheung, J. Han, Classification with active learning and meta-paths in heterogeneous information networks, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, ACM, New York, NY, USA, 2015, pp. 443–452.
[8] Y. Sun, C.C. Aggarwal, J. Han, Relation strength-aware clustering of heterogeneous information networks with incomplete attributes, Proc. VLDB Endow. 5 (5) (2012) 394–405.
[9] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, T. Wu, Rankclus: integrating clustering with ranking for heterogeneous information network analysis, in: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, ACM, 2009, pp. 565–576.
[10] S. Zhou, H. Yang, X. Wang, J. Bu, M. Ester, P. Yu, J. Zhang, C. Wang, Prre: Personalized relation ranking embedding for attributed networks, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, 2018, pp. 823–832.
[11] C. Hennig, M. Meila, F. Murtagh, R. Rocci, Handbook of Cluster Analysis, CRC Press, 2015.
[12] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering, in: Proceedings of the 2004 SIAM International Conference on Data Mining, SIAM, 2004, pp. 333–344.
[13] R. Bekkerman, M. Sahami, Semi-supervised clustering using combinatorial MRFs, in: ICML-06 Workshop on Learning in Structured Output Spaces, 2006.
[14] T. Lange, M.H. Law, A.K. Jain, J.M. Buhmann, Learning with constrained and unlabelled data, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 731–738.
[15] S. Basu, I. Davidson, K. Wagstaff, Constrained Clustering: Advances in Algorithms, Theory, and Applications, CRC Press, 2008.
[16] C. Aggarwal, Y. Xie, P. Yu, Towards community detection in locally heterogeneous, in: Proceedings of the 2011 SIAM International Conference on Data Mining, SDM '11, SIAM, 2011, pp. 391–402.
[17] G.-J. Qi, C.C. Aggarwal, T.S. Huang, On clustering heterogeneous social media objects with outlier links, in: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12, ACM, New York, NY, USA, 2012, pp. 553–562.
[18] H. Deng, B. Zhao, J. Han, Collective topic modeling for heterogeneous networks, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, ACM, New York, NY, USA, 2011, pp. 1109–1110.
[19] J. Han, Mining heterogeneous information networks by exploring the power of links, Discov. Sci. (2009) 13–30.
[20] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., Constrained k-means clustering with background knowledge, in: ICML, Vol. 1, 2001, pp. 577–584.
[21] H.-S. Park, C.-H. Jun, A simple and fast algorithm for K-medoids clustering, Expert Syst. Appl. 36 (2) (2009) 3336–3341.
[22] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2002, pp. 849–856.
[23] M.M. Keikha, M. Rahgozar, M. Asadpour, Community aware random walk for network embedding, Knowl.-Based Syst. 148 (2018) 47–54.
[24] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.

[25] M. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004).

[26] Y. Zhou, H. Cheng, J. Xu, Graph clustering based on structural and attribute similarities, in: VLDB, 2009, pp. 718–729.

[27] T. Yang, R. Jin, Y. Chi, S. Zhu, Combining link and content for community detection: a discriminative approach, in: KDD, 2009, pp. 927–936.

[28] U. Luxburg, A tutorial on spectral clustering, Statist. Comput. 17 (2007) 395–416.

[29] K. Wakita, T. Tsurumi, Finding community structure in mega-scale social networks, in: WWW, 2007, pp. 1275–1276.

[30] M. Sales-Pardo, R. Grimera, A. Moreira, L. Amaral, Extracting the hierarchical organization of complex systems, in: Proceedings of the National Academy of Sciences, Vol. 104, 2007, pp. 15224–15229.

[31] H. Wang, Y. Yang, B. Liu, H. Fujita, A study of graph-based system for multi-view clustering, Knowl.-Based Syst. 163 (2019) 1009–1019.

[32] J. Chen, Y. Feng, M. Ester, S. Zhou, C. Chen, C. Wang, Modeling users' exposure with social knowledge influence and consumption influence for recommendation, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, ACM, New York, NY, USA, 2018, pp. 953–962, http://dx.doi.org/10.1145/3269206.3271742.

[33] L. Yang, X. Cao, D. He, C. Wang, X. Wang, W. Zhang, Modularity based community detection with deep learning, in: IJCAI, Vol. 16, 2016, pp. 2252–2258.

[34] J.J. Choong, X. Liu, T. Murata, Learning community structure with variational autoencoder, in: 2018 IEEE International Conference on Data Mining, ICDM, IEEE, 2018, pp. 69–78.

[35] Y. Jia, Q. Zhang, W. Zhang, X. Wang, Communitygan: Community detection with generative adversarial nets, in: The World Wide Web Conference, ACM, 2019, pp. 784–794.

[36] B. Perozzi, L. Akoglu, P. Iglesias Sánchez, E. Müller, Focused clustering and outlier detection in large attributed graphs, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 1346–1355.

[37] J. Cao, S. Wang, F. Qiao, H. Wang, F. Wang, S.Y. Philip, User-guided large attributed graph clustering with multiple sparse annotations, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2016, pp. 127–138.

[38] L. Gui, Y. Zhou, R. Xu, Y. He, Q. Lu, Learning representations from heterogeneous network for sentiment classification of product reviews, Knowl.-Based Syst. 124 (2017) 34–45.

[39] Y. Zhou, J. Huang, H. Li, H. Sun, Y. Peng, Y. Xu, A semantic-rich similarity measure in heterogeneous information networks, Knowl.-Based Syst. 154 (2018) 32–42.

[40] P. Goyal, E. Ferrara, Graph embedding techniques, applications, and performance: A survey, Knowl.-Based Syst. 151 (2018) 78–94.

[41] Y. Sun, Y. Yu, J. Han, Ranking-based clustering of heterogeneous information networks with star network schema, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, New York, NY, USA, 2009, pp. 797–806.

[42] C. Shi, X. Kong, Y. Huang, S.Y. Philip, B. Wu, Hetesim: A general framework for relevance measure in heterogeneous networks, IEEE Trans. Knowl. Data Eng. 26 (10) (2014) 2479–2492.

[43] B. Long, Z.M. Zhang, X. Wu, P.S. Yu, Spectral clustering for multi-type relational data, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 585–592.

[44] Y. Zhou, L. Liu, Social influence based clustering of heterogeneous information networks, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, ACM, New York, NY, USA, 2013, pp. 338–346.

[45] F. Alqadah, R. Bhatnagar, A game theoretic framework for heterogenous information network clustering, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, ACM, New York, NY, USA, 2011, pp. 795–804.

[46] Y. Sun, B. Norick, J. Han, X. Yan, P.S. Yu, X. Yu, Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks, ACM Trans. Knowl. Discov. Data 7 (3) (2013) 11.

[47] Y. Sun, C.C. Aggarwal, J. Han, Relation strength-aware clustering of heterogeneous information networks with incomplete attributes, Proc. VLDB Endow. 5 (5) (2012) 394–405.

[48] Y. Sun, J. Han, Mining heterogeneous information networks: principles and methodologies, Synth. Lect. Data Min. Knowl. Discov. 3 (2) (2012) 1–159.

[49] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, 2012.

[50] C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, J. Guo, Mining advisor-advisee relationships from research publication networks, in: Proceedings of the Sixteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 203–212.

[51] J. Ramos, et al., Using tf-idf to determine word relevance in document queries, in: Proceedings of the First Instructional Conference on Machine Learning, Vol. 242, Piscataway, NJ, 2003, pp. 133–142.

[52] S. Basu, A. Banerjee, R. Mooney, Semi-supervised clustering by seeding, in: Proceedings of 19th International Conference on Machine Learning, ICML-2002, Citeseer, 2002.

[53] S.S. Rangapuram, M. Hein, Constrained 1-spectral clustering, in: AISTATS, Vol. 30, 2012, p. 90.

[54] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, G. Marchal, Automated multi-modality image registration based on information theory, in: Information Processing in Medical Imaging, Vol. 3, 1995, pp. 263–274.

[55] Y. Sun, J. Han, Mining Heterogeneous Information Networks: Principles and Methodologies, Morgan and Claypool Publishers, 2012.