Coupled Semi-supervised Clustering : Exploring Attribute Correlations in Heterogeneous Information Networks

Jianan Zhao[1], Ding Xiao[1], Linmei Hu[1], and Chuan Shi[1]*

Beijing University of Posts and Telecommunications, Beijing, China
{zhaojianan, dxiao, hulinmei, shichuan}@bupt.edu.cn

Abstract. Heterogeneous Information Network (HIN) has been widely adopted in various tasks due to its excellence in modeling complex network data. To handle the additional attributes of nodes in HIN, the Attributed Heterogeneous Information Network (AHIN) was brought forward. Recently, clustering on HIN becomes a hot topic, since it is useful in many applications. Although existing semi-supervised clustering methods in HIN have achieved performance improvements to some extent, these models seldom consider the correlations among attributes which typically exist in real applications. To tackle this issue, we propose a novel model SCAN for semi-supervised clustering in AHIN. Our model captures the coupling relations between mixed types of node attributes and therefore obtains better attribute similarity. Moreover, we propose a flexible constraint method to leverage supervised information and network information for flexible adaption of different datasets and clustering objectives. Extensive experiments have shown that our model outperforms state-of-the-art algorithms.

Keywords: Attributed heterogeneous information network· Semi-supervised clustering· Coupled attributes.

1 Introduction

Heterogeneous Information Network (HIN) [15], as a new network modeling method, has drawn much attention due to its ability to model complex objects and their rich relations. Moreover, in many real HINs, objects are often associated with various attributes. For example, in Yelp dataset, where businesses and users can be regarded as nodes. Business objects are associated with attributes like locations, ratings and business types; besides, users own attributes including age and gender. Researchers have brought forward the concept of Attributed Heterogeneous Information Network [9] (AHIN) to address HINs with node attributes.

Clustering is a fundamental task in data analysis. Given a set of objects, the goal is to partition them into clusters such that objects in the same clusters are similar to each other, while objects in different clusters are dissimilar.

Semi-supervised clustering incorporates supervision about clusters into the algorithm in order to improve the clustering results. Till now, many semi-supervised clustering algorithms for information networks have been proposed. Some semi-clustering methods [1,6,8] are proposed for semi-supervised clustering in homogeneous information networks, where links are assumed to be of the same type. As for heterogeneous information networks, GNetMine [4], PathSelClus [16] and SemiRPClus [10] are proposed. All of the algorithms above do not concern the attributes of nodes. Recently, SCHAIN [9] studies the problem of semi-supervised clustering in AHIN and achieves state-of-the-art clustering performance.

Although these methods have achieved satisfactory performances to some extent, they have some obvious shortcomings. First of all, they do not consider the coupling relationships among object attributes. In the real world, attributes are associated with each other and have complex relationships addressed as coupling relationships [10,16]. To illustrate, in a movie dataset, the "budget" and "gross" attribute of a movie is largely dependent on the "country" attribute. Ignoring the dependency between attributes will inevitably lead to inferior clustering results. In addition, previous works cannot flexibly consider the importance of supervision with datasets while many real applications may need to take into account the different importance of supervision for different tasks.

To handle these issues, we put forward an innovative model **S**emi-supervised Clustering with **C**oupled Attributes in **A**ttributed Heterogeneous Information **N**etworks (SCAN). Inspired by the newly emerging non-IID learning [3], we take one step further to mine the coupled similarity between node attributes, thus capturing the inter-dependent relationships between attributes. Moreover, we propose a novel constraint method to flexibly leverage the supervision information through which users can adjust the importance of supervised information for different clustering objectives.

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

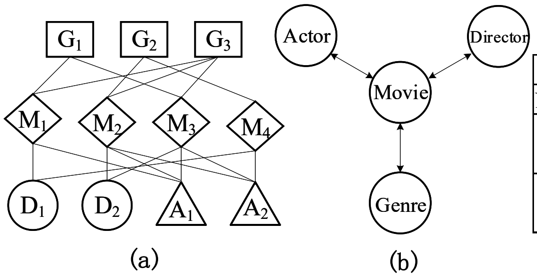
- To our best knowledge, we are the first to mine the coupling relationships between node attributes in AHIN. We propose a coupled node similarity measure to better analyze the inter-dependent relationships between mixed data types of node attributes.
- We propose a novel approach to use supervision information, which is able to leverage node similarity and supervision constraint flexibly.
- We conduct extensive experiments on two real-world datasets. The proposed method is proved to be effective over the state-of-the-art methods.

2 Definitions and Model Overview

2.1 Problem Definition

In this paper, we consider the problem of semi-supervised clustering in AHINs. The terms are defined as follows.

Definition 1. Attributed Heterogeneous Information Network (AHIN) [9] Let $\mathcal{T} = \{T_1, ..., T_m\}$ be a set of m object types. For each type T_i, let $\mathcal{X}_i$



Node Attributes		
Node Type	Categorical	Numerical
Movie	country, language	budget, gross, IMDB score
Actor, Director	gender, country	Facebook fans, average gross

(c)

Fig. 1: An AHIN (a), its network schema (b), and its node attributes (c)

be the set of objects of type T_i and A_i be the set of attributes defined for objects of type T_i. An object x_j of type T_i is associated with an attribute vector $\boldsymbol{f}_j = (f_{j_1}, f_{j_2}, \ldots, f_{j_{|A_i|}})$. An AHIN is a graph $G = (V, E, \mathcal{A})$, where $V = \bigcup_{i=1}^{m} \mathcal{X}_i$ is a set of nodes, E is a set of links, each representing a binary relation between two objects in V, and $\mathcal{A} = \bigcup_{i=1}^{m} A_i$. If $m = 1$ (i.e., there is only one object type), G reduces to a homogeneous information network.

Fig.1a shows an example of movie AHIN consisting of four types of objects: T={movies (M), actors (A), genres (G), directors (D)}. The network schema [15] is shown in Fig.1b. In this AHIN, the M, A and D node types are associated with both numerical and categorical attributes shown in Fig.1c.

A **meta-path** [15] $\mathcal{P} : T_1 \xrightarrow{R_1} \cdots \xrightarrow{R_l} T_{l+1}$ defines a composite relation $R = R_1 \circ \cdots \circ R_l$ that relates objects of type T_1 to objects of type T_{l+1}. We say $\mathcal{P}$ is symmetric if the defined relation R is symmetric. Giving a meta-path $\mathcal{P}$, the multiple specific paths under the meta-path are called path instances denoted by P. To illustrate, in the AHIN shown in Fig.1a, the meta-path MDM means two movies are directed by the same director, $M_1D_1M_4$ is a path instance of the MDM meta-path. Different meta-paths preserve different semantic meanings in AHIN.

Definition 2. Semi-supervised clustering in an AHIN [9] Given an AHIN $G = (V, E, \mathcal{A})$, a supervision constraint $(\mathcal{M}, \mathcal{C})$ of must-link set $\mathcal{M}$ and the cannot-link set $\mathcal{C}$, a target object type T_i, the number of clusters k, and a set of meta-paths $\mathcal{PS}$, the problem of semi-supervised clustering of type T_i objects in G is to (1) discover an object similarity measure S that is based on object attributes and meta-paths, and (2) partition the objects in X_i into k disjoint clusters $\mathbf{C} = C_1, \ldots, C_k$ based on the similarity measure S such that the clustering results best agree with the constraint $(\mathcal{M}, \mathcal{C})$.

2.2 Model Overview

In this section, we will take a brief look at the proposed model SCAN. As introduced in Definition 2.1, the task can be decomposed into two steps, namely node similarity calculation and clustering. In the following, we will detail the two steps.

To begin with, our model calculates the similarity of every node pair based on their attribute similarity and the nodes network connectedness: The former is obtained by an attribute similarity measure considering coupling relationship amongst attributes, while the latter is derived based on the meta-paths connecting the object pair. Moreover, SCAN assigns a weight to each object attribute and meta-path to compute the overall node similarity S, thus enabling the model to learn different weights for different clustering objectives. As for the supervision constraint, SCAN proposes a flexible penalty function to leverage clustering quality and supervision. Finally, SCAN employs the iterative, 2-step learning process of SCHAIN [9] to determine the optimal weights and cluster assignments as output.

The rest of the paper is organized as follows: The calculation of coupled similarity between nodes is explained in Section 3, clustering and weight learning process are explained in Section 4, extensive experiments along with parameter discussion are illustrated in Section 5. Finally, we conclude the whole paper in Section 6.

3 Node Similarity with Coupled Attributes

In this section, we introduce how to calculate the similarity between two nodes in an AHIN. Given a node pair $< x_a, x_b >$ of the same node type T_i in an AHIN, the node similarity between a node pair in an AHIN is twofold: On the one hand, similarity can be obtained from attribute similarity; on the other hand, similarity can be measured by the connectedness of the given node pair, namely link similarity. The overall node similarity matrix S can be obtained by aggregating the coupled attribute similarity matrix S_A and the link similarity matrix S_L. To mine the complex coupling relationships between node attributes, we take one step further to calculate the coupled similarity between node attributes. We will detail this concept in the following section.

3.1 Coupled Node Attribute Similarity

In this section we explain how to capture the coupling relations between node attributes. Given a node pair $< x_a, x_b >$ of type T_i, let $\boldsymbol{f}_a$ and $\boldsymbol{f}_b$ be the attribute vectors of x_a and x_b, respectively (see Definition 1). Recall that A_i is the set of attributes associated with type T_i objects. We define an attribute weight vector $\boldsymbol{\omega}$, whose j-th component, ω_j, captures the importance of the j-th attribute in A_i for the clustering task. The entries of S_A are calculated by:

$$S_A(x_a, x_b) = \sum_{j=1}^{|A_i|} \omega_j \cdot valSim(f_{aj}, f_{bj}). \tag{1}$$

The $valSim()$ function calculates the similarity between the attribute value f_{aj} and f_{bj}. Since in real world, many nodes are assigned with mixed type of attributes. We define A_{ij} as the j-th attribute type of node A_i and calculate the value similarity of different attribute types separately.

For numerical attributes, we firstly normalize numerical attribute to $[0, 1]$, and use Equation (2) to convert distance to similarity. Euclidean distance is adopted as distance metric in this paper.

$$valSim(f_{aj}, f_{bj}) = 1 - d(f_{aj}, f_{bj}), \;\; if \; A_{ij} \, is \, numerical. \tag{2}$$

However, categorical attributes are associated with each other in terms of certain coupling relationships, ignoring attribute value similarity will lead to coarse similarity [17]. This motivates us to design a coupled similarity measure of categorical attributes. Thus, we go one step further to mine the coupled value similarity between categorical attribute values:

$$\begin{aligned} valSim(f_{aj}, f_{bj}) =& (1 - \eta) S_p(f_{aj}, f_{bj}) \\ & + \eta S_c(f_{aj}, f_{bj}), \;\; if \; A_{ij} \, is \, categorical. \end{aligned} \tag{3}$$

We capture two levels of categorical value similarity: The plain similarity S_p to measure the value difference of the categorical value, and the coupled similarity S_c to measure the value similarity in terms of couplings. The overall categorical attribute similarity is obtained by a weighted fusion of S_p and S_c using coupled coefficient η.

For plain similarity, we use SMS [7], which uses 0s and 1s to distinguish the similarity between distinct and identical categorical values, to calculate the value difference. As for the coupled similarity, following IeASV [17], we measure the co-occurrence of the measured attribute to other attributes. To elaborate, we bring forward the inter-coupled attribute value similarity.

Definition 3. Inter-coupled attribute value similarity (IAVS) Given two attribute values f_{aj} and f_{bj} from attribute A_{ij} in an AHIN. The inter-coupled attribute value similarity between attribute values f_{aj} and f_{bj} in terms of attribute A_{ik} is denoted by $\delta_{j|k}(f_{aj}, f_{bj})$.

Although the definition of IAVS is similar to IeASV, IeASV only handles categorical data. Since we are dealing with mixed types of data, we generalize the inter-coupled relative similarity candidate A_{ik} to numerical data. Therefore, the coupled similarity of two attribute values $S_c(f_{aj}, f_{bj})$ can be obtained by an aggregated similarity of other attributes:

$$S_c(f_{aj}, f_{bj}) = \frac{\sum_{k=1, k \neq j}^{|A_i|-1} \delta_{j|k}(f_{aj}, f_{bj})}{|A_i| - 1}. \tag{4}$$

Let us explain the intuition behind the equation by a toy example. In the movie AHIN shown in Fig 1, suppose we are calculating the coupled similarity $S_c(f_{aj}, f_{bj})$ between attribute value $f_{aj} =$ "USA" and $f_{bj} =$ "UK" of attribute $A_{ij} =$ "country". The coupled similarity of "USA" and "UK" will be the aggregated similarity of USA movies and UK movies in terms of other attributes

A_{ik} ∈{"language", "budget", "gross", "IMDB score"} calculated by Equation (4).

For categorical attribute similarity, we adopt the inter-coupled relative similarity based on intersection set (IRSI) [17] to capture the co-occurrence relationship between categorical attributes. However, IRSI is designed to measure the dependencies between categorical variables, while the coupling relationships between categorical and numerical attributes remain undiscussed.

To capture the dependency between a categorical attribute and a numerical attribute, we define $Val(f_{aj}, A_{ik})$ as the values sets of attribute A_{ik} generated by the objects with value f_{aj}. In our example, if $A_{ik} =$ "*gross*", then $\delta_{j|k}(f_{aj}, f_{bj})$ stands for the similarity between USA and UK movies in terms of gross respectively. Thus, $Val(f_{aj}, A_{ik})$ and $Val(f_{bj}, A_{ik})$ stands for the gross values of USA and UK movies. The task is to judge the similarity between the two distributions of $Val(f_{aj}, A_{ik})$ and $Val(f_{bj}, A_{ik})$, the closer the two distributions, the more similar the two categorical values are. This is a problem of measuring two finite and continuous distributions.

Inspired by the histogram, we simplify the problem by discretizing the two distributions to finite discrete distributions. The discretization method can be any of the methods described in [5]. We define $\Psi(V_{ajk}, P_{ajk})$ as the discretized distribution of $Val(f_{aj}, A_{ik})$, where V_{ajk} and P_{ajk} are the discretized value sets (known as bins in the histogram) of $Val(f_{aj}, A_{ik})$ and its corresponding value occurrence frequency. For simplicity, we use uniform quantization as our discretization method in the experiments.

Till now, the dependency between a categorical attribute and a numerical attribute has been simplified to the similarity between two discrete distributions. We first calculate the distance between two distributions and then we convert the distance to similarity. Thus, $\delta_{j|k}(f_{aj}, f_{bj})$ can be calculated by the following equation:

$$\delta_{j|k}(f_{aj}, f_{bj}) = \begin{cases} IRSI(f_{aj}, f_{bj}, A_{ik}), & if\, A_{ik}\, is\, categorical. \\ 1 - d(\Psi(V_{ajk}, P_{ajk}), \Psi(V_{bjk}, P_{bjk})), & if\, A_{ik}\, is\, numerical. \end{cases} \tag{5}$$

Since A_{ik} is numerical, the difference between different discretized sets in V_{ajk} and V_{bjk} can be measured by the distance between the corresponding discretized numerical attribute values (bins in the histogram). To capture this cross-bin relationship, we adopt the earth mover distance (EMD) [12] to measure the distance between the two value distributions.

3.2 Link Similarity and Similarity Aggregation

In this section, we calculate the similarity between a node pair in terms of network connectedness. We adopt the widely used meta-path to calculate link similarity. Given a meta-path set $\mathcal{PS}$, we define a weight vector $\boldsymbol{\lambda}$. Each meta-path $\mathcal{P}_j \in \mathcal{PS}$ is assigned to a weight λ_j, the overall link similarity S_L is obtained by a weighting scheme:

$$S_L(x_a, x_b) = \sum_{j=1}^{|\mathcal{PS}|} \lambda_j \cdot S_{\mathcal{P}_j}(x_a, x_b), \tag{6}$$

SCHAIN [9] uses PathSim [15] to measure the meta-path based similarity $S_{\mathcal{P}_j}$ for a given path $\mathcal{P}_j$. However, PathSim is designed for computing the similarity between symmetric meta-paths. In real applications, asymmetric meta-paths are also useful and cannot be ignored. To illustrate, in movie networks, MAMDM and MDMAM are asymmetric yet useful meta-paths which preserve the cooperating relationship between actors and directors. Thus, we use HeteSim [13] to calculate $S_{\mathcal{P}_j}$, so that SCAN is capable of handling both symmetric and asymmetric meta-paths.

$$S_{\mathcal{P}_j}(x_a, x_b) = HeteSim(x_a, x_b), \tag{7}$$

The overall node similarity matrix S represents the aggregated similarity of link similarity and coupled attribute similarity, which can be calculated by a weighted sum of S_A and S_L:

$$S = \alpha S_A + (1 - \alpha) S_L, \tag{8}$$

where α is a weighting factor that controls the relative importance of the two similarity matrices.

4 Clustering and Weight Learning

4.1 Leveraging the Supervision

The penalty function is introduced in this part. Recall that our objective is to get $\mathbf{C} = C_1, \ldots, C_k$ based on the similarity matrix S such that the clustering results best agree with the constraint $(\mathcal{M}, \mathcal{C})$. After generating the similarity matrix S, we adopt the clustering and optimization framework of SCHAIN. Different from SCHAIN, we propose a more flexible penalty function to better leverage the supervised information.

We use a semi-supervised normalized cut [14] to measure the similarity between clusters. The penalty function of clustering is:

$$\begin{aligned}\mathcal{J}(\boldsymbol{\lambda}, \boldsymbol{\omega}, \{C_r\}_{r=1}^k) = &\sum_{r=1}^{k} \frac{links(C_r, \mathcal{X}_i \backslash C_r)}{links(C_r, \mathcal{X}_i)} \\ &- \beta \sum_{r=1}^{k} \sum_{\substack{(x_a, x_b) \in \mathcal{M} \\ L(x_a) = L(x_b) = r}} \frac{S(x_a, x_b)}{links(C_r, \mathcal{X}_i)} \\ &+ \rho \sum_{r=1}^{k} \sum_{\substack{(x_a, x_b) \in \mathcal{C} \\ L(x_a) = L(x_b) = r}} \frac{S(x_a, x_b)}{links(C_r, \mathcal{X}_i)}.\end{aligned} \tag{9}$$

This penalty function is composed of two parts:

1. Clustering quality based on the similarity matrix S: Normalized cut [14] is used to define the similarity between clusters defined as:$NC = \sum_{r=1}^{k} \frac{links(C_r, \mathcal{X}_i \backslash C_r)}{links(C_r, \mathcal{X}_i)}$, where $links(C_p, C_q) = \sum_{x_a \in C_p, x_b \in C_q} S(x_a, x_b)$. The larger the similarity between different clusters, the worse the clustering quality.
2. Supervision constraints: For an object pair $< x_a, x_b >$ belongs to the same cluster C_r, the presence of $< x_a, x_b >$ in $\mathcal{M}$ indicates good clustering quality; the presence of $< x_a, x_b >$ in $\mathcal{C}$ indicates bad clustering quality.

For flexibility, we define two supervision importance parameters β and ρ to balance the influence of supervised information set. The penalty function of SCHAIN for clustering is equal to Equation (9) with β and ρ assigned to 1. Thus, SCHAIN regards clustering objective and supervision of $\mathcal{M}$ and $\mathcal{C}$ as equal importance. However, while handling different datasets and different clustering objectives, it is flexible to cluster with different β and ρ. In this paper, we set ρ to a fixed value 1, and use β as a hyper-parameter. The larger β indicates more penalty considered for the object pairs in must-link set $\mathcal{M}$.

By defining $\{\boldsymbol{z}_r\}_{r=1}^{k}$, where each $\boldsymbol{z}_r$ is a binary indicator vector of length $n = |\mathcal{X}_i|$, $\boldsymbol{z}_r(a) = 1$ represents x_a is assigned to cluster C_r and 0 otherwise, we obtain the overall penalty function for clustering:

$$\mathcal{J}(\boldsymbol{\lambda}, \boldsymbol{\omega}, \{\boldsymbol{z}_r\}_{r=1}^{k}) = \sum_{r=1}^{k} \frac{\boldsymbol{z}_r^T (D - S - W \circ S) \boldsymbol{z}_r}{\boldsymbol{z}_r^T D \boldsymbol{z}_r}. \tag{10}$$

$\circ$ is the Hadamard product for two matrices. W is a constraint matrix of $W \in \mathbb{R}^{n \times n}$, where $W(x_a, x_b) = \beta$ for $< x_a, x_b > \in \mathcal{M}$, $W(x_a, x_b) = -1$ for $< x_a, x_b > \in \mathcal{C}$ and 0 otherwise. Furthermore, we add a regularization term to Equation (10) and get the overall penalty function:

$$\mathcal{J}(\boldsymbol{\lambda}, \boldsymbol{\omega}, \{\boldsymbol{z}_r\}_{r=1}^{k}) = \sum_{r=1}^{k} \frac{\boldsymbol{z}_r^T (D - S - W \circ S) \boldsymbol{z}_r}{\boldsymbol{z}_r^T D \boldsymbol{z}_r} + \gamma(\|\lambda\|^2 + \|\omega\|^2). \tag{11}$$

Finally, to find the best clustering, we minimize the penalty function subject to the following constraints: $\sum_{r=1}^{k} \boldsymbol{z}_r(a) = 1$;$\boldsymbol{z}_r(a) \in \{0, 1\}$;$\sum_{l=1}^{|A_i|} \omega_l = 1$; $\omega_l > 0$ and $\lambda_j > 0$. Note that α, β, η and γ are hyper-parameters in the function.

4.2 Model Optimization

There are two objectives to be optimized in the learning process: the clustering results $\{\boldsymbol{z}_r\}_{r=1}^{k}$ and the weighting vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$. In this section, we introduce the optimization of our model. Following SCHAIN [9], we use a mutual updating optimization method. Firstly, given the weights $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$, we find the optimal clustering $\{\boldsymbol{z}_r\}_{r=1}^{k}$. Secondly, given $\{\boldsymbol{z}_r\}_{r=1}^{k}$, we find the optimal $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$. We iterate until $\mathcal{J}$ is smaller than a given threshold ϵ. We will briefly explain how the two update steps are performed.

Step 1: Optimize $\{\boldsymbol{z}_r\}_{r=1}^{k}$ given $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$

In this step, the task is to find the best clustering $\{\boldsymbol{z}_r\}_{r=1}^{k}$.We define a matrix $\hat{Z}$ with the r-th column of $\hat{Z}$ equals to $D^{\frac{1}{2}}\boldsymbol{z}_r/(\boldsymbol{z}_r^T D\boldsymbol{z}_r)^{\frac{1}{2}}$, s.t. $\hat{Z}^T\hat{Z} = I_k$, where I_k is the identity matrix of $\mathbb{R}^{k\times k}$. Since the weights of meta-paths and attributes, namely $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$, are given, the objective function $\mathcal{J}(\boldsymbol{\lambda}, \boldsymbol{\omega}, \{\boldsymbol{z}_r\}_{r=1}^{k})$ becomes a function of clustering indicator matrix $\{\boldsymbol{z}_r\}_{r=1}^{k}$. The minimization of $\mathcal{J}$ can be derived to a trace maximization problem:

$$\max_{\hat{Z}^T\hat{Z}=I_k} trace(\hat{Z}^T D^{-\frac{1}{2}}(S + W \circ S)D^{-\frac{1}{2}}\hat{Z}), \tag{12}$$

which has a closed form solution by calculating top k eigenvectors [2]. K-means is then adopted to obtain hard clustering results $\{\boldsymbol{z}_r\}_{r=1}^{k}$.

Step 2: Optimize $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ given $\{\boldsymbol{z}_r\}_{r=1}^{k}$

In this step, the model finds the best $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ given fixed cluster $\{\boldsymbol{z}_r\}_{r=1}^{k}$. As proved in SCHAIN [9], minimizing $\mathcal{J}$ is equivalent to maximizing:

$$\max_{\boldsymbol{\lambda},\boldsymbol{\omega}} \sum_{r=1}^{k} \frac{\boldsymbol{z}_r^T(S + W \circ S)\boldsymbol{z}_r}{\boldsymbol{z}_r^T D\boldsymbol{z}_r} - \gamma(\|\boldsymbol{\lambda}\|^2 + \|\boldsymbol{\omega}\|^2), \tag{13}$$

which can be rewritten as:

$$H(\boldsymbol{\lambda}, \boldsymbol{\omega}) = \max_{\boldsymbol{\lambda},\boldsymbol{\omega}} \frac{f(\boldsymbol{\lambda}, \boldsymbol{\omega})}{g(\boldsymbol{\lambda}, \boldsymbol{\omega})}, \tag{14}$$

where $f(\boldsymbol{\lambda}, \boldsymbol{\omega})$ and $g(\boldsymbol{\lambda}, \boldsymbol{\omega})$ are two nonlinear multivariate polynomial function. This can be optimized by solving the following non-linear parametric programming problem: Let $f(\boldsymbol{\lambda}, \boldsymbol{\omega})$ and $g(\boldsymbol{\lambda}, \boldsymbol{\omega})$ be two multivariate polynomial functions. For a given μ, find $F(\mu) = f(\boldsymbol{\lambda}, \boldsymbol{\omega}) - \mu g(\boldsymbol{\lambda}, \boldsymbol{\omega})$, s.t. $\sum_{r=1}^{k} \boldsymbol{z}_r(a) = 1$; $\boldsymbol{z}_r(a) \in \{0, 1\}$; $\sum_{l=1}^{|A_i|} \omega_l = 1$; $\omega_l > 0$ and $\lambda_j > 0$. Readers may refer to SCHAIN [9] for proving and other details.

To sum up the whole model, SCAN firstly calculates the coupled attribute node similarity and link similarity between nodes. Secondly, SCAN computes the overall similarity by aggregating the similarity matrices with weighting vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ and a balance factor α. The constraint matrix is generated by supervision using supervision importance β. Finally, SCAN adopts an iterative mutual update process to learn the clustering and weighting vectors.

5 Experiments

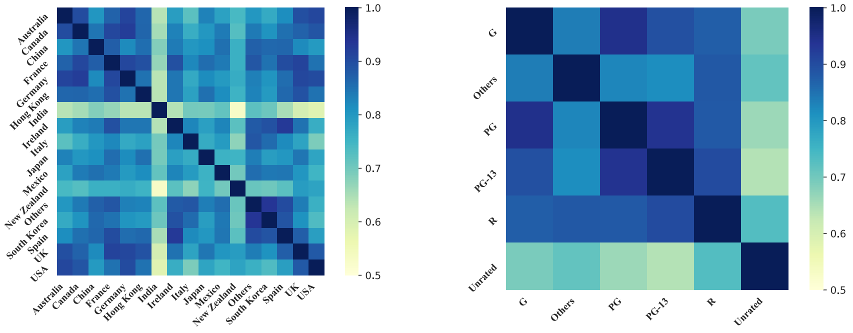
In this section, we firstly introduce the datasets used in the experiments. Then we discuss the attribute similarity matrix of the IMDB dataset. Further, we show the effectiveness of the SCAN model against 6 representative algorithms. Finally, we explain the influence of different hyper-parameters and the weight learning process.

5.1 Datasets

Yelp[1] This experiment is similar to the experiment of Yelp-Restaurant dataset used in SCHAIN [9]. The clustering task is to cluster restaurants by three subcategories: “Fast Food”, “Sushi Bars” and “American (New) Food”. The AHIN is composed of 2,614 business objects (B); 33,360 review objects (R); 1,286 user objects (U) and 82 food relevant keyword objects (K). 5 attributes are considered: 3 categorical attributes (reservation, service, and parking) and 2 numerical attributes (review count and quality star). For meta-paths: we choose BRURB (businesses reviewed by the same user) and BRKRB (businesses receive the same keyword in the reviews).

IMDB5k[2] We extracted an AHIN network from the IMDB5k dataset. The AHIN is composed of 4 node types: 4,140 movies (M), 4,907 actors(A), 1,867 directors (D) and 24 Genres (G). The movies are extracted into three types by their IMDB scores. Each movie has 2 categorical attributes (content rating and country) and 3 numerical attributes (gross, critic reviews and movie Facebook likes). Three meta-paths: MAM (movies with the same actor), MGM (movies with the same genre) and MDM (movies with the same director) are chosen in our experiments for link-based algorithms. Note that, as mentioned before, PathSim based clustering algorithms cannot handle asymmetric meta-paths, though SCAN is able to handle asymmetric meta-paths, we choose symmetric meta-paths for the sake of fairness.

5.2 Discussion of Attribute Value Similarity



(a) Attribute similarity of IMDB-country (b) Attribute similarity of IMDB-ratings

Fig. 2: Attribute value similarity of IMDB

[1] http://www.yelp.com/academic_dataset

[2] https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset

Table 1: The MPAA rating system.

Content rating	Description
G	General Audiences: All ages admitted.
PG	Parental Guidance Suggested: May not be suitable for children.
PG-13	Parents Strongly Cautioned: Inappropriate for children under 13.
R	Restricted: Children under 17 requires adult guardian.
Unrated	Not Rated or Unrated: The film has not been submitted for a rating.

The SCAN model captures the coupling relationships between attributes and therefore obtains better node attribute similarity. We will take a closer look at the attribute value similarity of the IMDB dataset. There are two categorical attributes in the IMDB dataset, namely “country” and “content ratings”. The similarity between the attribute values are visualized in Fig.2. Attribute value pairs with higher similarity are visualized with darker colors.

The attribute similarity of “country” attribute of node type “movie” is shown in Fig.2a. Top 15 countries in the dataset are analyzed, other countries are combined to the “Others” category. As known, the USA, UK, France, Canada and Australia are all developed countries while India is a developing country. By mining the dependency between movie attributes, our model successfully discovers that USA, UK, France, Canada, and Australia are similar countries, while India is dissimilar to any of the other countries shown in the table. Interestingly, our algorithm also reveals that New Zealand and India are the most dissimilar attribute values in the selected countries.

The Motion Picture Association of America (MPAA) film rating system[3] is used to rate a film’s suitability for certain audiences based on its content The attribute similarity of “content rating” attribute of node type “movie” is shown in Fig.2b. Top 5 genres in the dataset are analyzed (“G”, “PG”, “PG-13”, “R”, and “Unrated”), other genres are combined to the “Others” category. The meanings of the rating levels are shown in the Table 1. Generally speaking, our attribute value similarity obtained is reasonable. To illustrate, our model finds that “G” is most similar to “PG” and “G” is more similar to “PG-13” than “R”. Meanwhile, “R” is most similar to “PG-13”, which is its neighbor level in Table 1; “Others” is somehow similar to other rating levels since its a combined categorical value. “Unrated” is relatively dissimilar to other rating levels.

5.3 Clustering Performance

We conduct experiments of three groups of comparison algorithms: attribute-only, link-only and attribute+link. For each group of comparison algorithms, we consider two algorithms listed as follows.

- **Attribute-only:** The clustering algorithms in the first group consider only object attributes. These are traditional methods which ignore the network

[3] https://www.mpaa.org/film-ratings/

structure of an AHIN. We choose Spectral-Learning [6] and a semi-supervised version of normalized cuts [8] as representatives, which are denoted SL and SNcuts, respectively.

– **Link-only** These methods utilize only the link information of the network and they ignore object attribute values. GNetMine [4], PathSelClus [16] are chosen as representative methods of this category.
– **Attribute+Link**: Methods of this group use both attribute and link information. We consider FocusCO [11], SCHAIN [9], and SCAN-C. Since FocusCO does not consider the heterogeneity of networks, the AHINs are constructed as homogeneous networks for experiments of FocusCO. The SCAN-C is another version of our model using CASV [17] as the categorical attribute value similarity measure. Since CASV only considers categorical data, in the SCAN-C related experiments, numerical attributes are converted to categorical attributes.

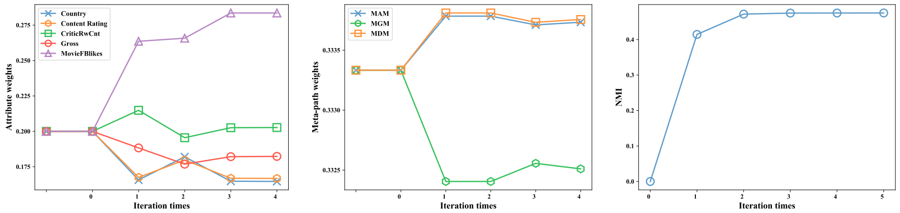
Table 2 shows the clustering performance in terms of NMI for Yelp-Restaurant and IMDB5k respectively. We run every experiment 10 times and calculate the average NMI. As can be observed, for both datasets, the clustering performance of SCAN outperforms other algorithms.

For Yelp-Restaurant, the attribute-only algorithms perform relatively worse than link-only algorithms, suggesting that link similarity is more significant of this clustering objective. Since our improvement is mostly on attribute similarity, SCAN turns out to be slightly better than SCHAIN.

For IMDB, due to the sparsity of the network, the attribute-only algorithms perform much better than link-only algorithms, indicating that attribute-based similarity is more significant to the clustering objective. Taking into account the coupling relationships between attributes, SCAN outperforms other algorithms significantly on IMDB dataset. Additionally, SCAN generally outperforms SCAN-C, which proves the importance of mining the coupling relationships between mixed type attributes.

Table 2: NMI Comparison on Yelp-Restaurant and IMDB5k

Dataset	seeds	SL	SNCuts	GNetMine	PathSelClus	FocusCO	SCHAIN	SCAN-C	SCAN
Yelp	5%	0.156	0.190	0.278	0.564	0.088	0.681	0.683	**0.708**
	10%	0.179	0.192	0.288	0.612	0.087	0.703	0.692	**0.727**
	15%	0.289	0.194	0.364	0.632	0.093	0.709	0.714	**0.746**
	20%	0.295	0.198	0.380	0.631	0.090	0.736	0.736	**0.754**
	25%	0.303	0.253	0.399	0.637	0.090	0.742	0.752	**0.764**
IMDB5k	5%	0.118	0.102	0.083	0.085	0.072	0.132	0.180	**0.325**
	10%	0.108	0.191	0.127	0.135	0.075	0.137	0.165	**0.335**
	15%	0.149	0.175	0.157	0.168	0.082	0.191	0.187	**0.370**
	20%	0.156	0.220	0.157	0.203	0.092	0.223	0.240	**0.450**
	25%	0.415	0.259	0.246	0.247	0.090	0.361	0.371	**0.493**

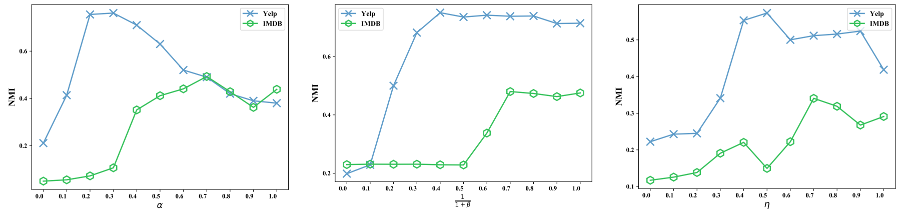


(a) Attribute weight learning (b) Meta-path weight learning (c) Clustering performance

Fig. 3: Weight learning on IMDB

5.4 Weight Learning

SCAN retains the ability of SCHAIN [9] to learn the weights of attributes and meta-paths. We will take clustering of IMDB dataset for illustration. In the following discussion, we assume 25% seed objects. Fig.3a and Fig.3b exhibit how our algorithm learns the weight through iteration for attributes and meta-paths. From Fig.3c, we see that SCAN identifies that the "Facebook likes" attribute to be the most useful attribute to cluster movies. As shown in Fig.3b, the weights of meta-paths remain almost unchanged through optimization, since the network is really sparse and link similarity does not contribute much to the clustering task. In Fig.3c, we observe that the overall clustering performance is improving through iterations and finally to an optimal value.



(a) Varying α (b) Varying β (c) Varying η

Fig. 4: Influence of hyper-parameters on Yelp

5.5 Parameters Discussion

There are three hyper-parameters to be tuned in our model: The balance coefficient α for link similarity and attribute similarity, the supervision importance β to leverage constraints, and the coupled coefficient η to balance plain and coupled similarity. The influence of hyper-parameters for Yelp and IMDB are shown in Fig.4.

Fig.4a shows the relationship between clustering performance and the balance coefficient α. If $\alpha = 0$, only link similarity is considered; if $\alpha = 1$, only attribute similarity is considered. We can observe that the attribute similarity is more important for IMDB and link similarity is more important for Yelp in terms of clustering performance. Therefore, it is better to balance between link similarity and attribute similarity.

Fig.4b shows the relationship between clustering performance and supervision importance β. For visualization, we map the value of β from $[0, \infty]$ to $[0, 1]$, and plot $\frac{1}{\beta+1}$. We can observe that, in both datasets, focusing too much (large β) on the supervision always lead to inferior results. In Yelp, the optimal $\beta = 1.5$, whilst in IMDB, it is better to set $\beta = 0.43$. This result indicates that the importance of supervision in terms of clustering is different, it's better to leverage the supervision and node similarity flexibly.

Fig.4c shows the relationship between clustering performance and the coupled coefficient η. If $\eta = 0$, plain similarity (SMS [7]) is used as our categorical value similarity; if $\eta = 1$, the plain similarity is ignored and only coupled similarity is concerned. We can observe that, for both datasets, it is better to balance the plain similarity and the coupled similarity. The optimal values of η for Yelp and IMDB are 0.5 and 0.7 respectively, which indicates that the attributes from different datasets have different degree of coupling relationships. For highly correlated datasets, high coupled coefficients are recommended.

6 Conclusion

In this paper, we study semi-supervised clustering in attributed heterogeneous information networks. To mine the coupling relationships between node attributes, we propose a new model SCAN. In our model, We present a coupled attribute node similarity measure to capture the dependency between mixed attribute types. Furthermore, we use a flexible approach to better leverage the importance of supervision information and network similarity. Extensive experiments are conducted on two real-world datasets to prove that our model is capable of capturing the coupling relationships between attributes and outperforms other state-of-the-art algorithms.

7 Acknowledgement

This work is supported by the National Key Research and Development Program of China (2017YFB0803304), the National Natural Science Foundation of China (No. 61772082, 61806020, 61702296) the Beijing Municipal Natural Science Foundation (4182043), the CCF-Tencent Open Fund and the Fundamental Research Funds for the Central Universities.

References

1. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: In Proceedings of 19th International Conference on Machine Learning (ICML-2002. Citeseer (2002)
2. Bhatia, R.: Matrix analysis, volume 169 of graduate texts in mathematics (1997)
3. Cao, L.: Coupling learning of complex interactions. Information Processing & Management **51**(2), 167–186 (2015)
4. Ji, M., Sun, Y., Danilevsky, M., Han, J., Gao, J.: Graph regularized transductive classification on heterogeneous information networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 570–586. Springer (2010)
5. Jin, R., Breitbart, Y., Muoh, C.: Data discretization unification. Knowledge and Information Systems **19**(1), 1 (2009)
6. Kamvar, K., Sepandar, S., Klein, K., Dan, D., Manning, M., Christopher, C.: Spectral learning. In: International Joint Conference of Artificial Intelligence. Stanford InfoLab (2003)
7. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis, vol. 344. John Wiley & Sons (2009)
8. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: a kernel approach. Machine learning **74**(1), 1–22 (2009)
9. Li, X., Wu, Y., Ester, M., Kao, B., Wang, X., Zheng, Y.: Semi-supervised clustering in attributed heterogeneous information networks. In: Proceedings of the 26th International Conference on World Wide Web. pp. 1621–1629. International World Wide Web Conferences Steering Committee (2017)
10. Luo, C., Pang, W., Wang, Z.: Semi-supervised clustering on heterogeneous information networks. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 548–559. Springer (2014)
11. Perozzi, B., Akoglu, L., Iglesias Sánchez, P., Müller, E.: Focused clustering and outlier detection in large attributed graphs. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1346–1355. ACM (2014)
12. Rubner, Y., Tomasi, C.: The earth movers distance. In: Perceptual Metrics for Image Database Navigation, pp. 13–28. Springer (2001)
13. Shi, C., Kong, X., Huang, Y., Philip, S.Y., Wu, B.: Hetesim: A general framework for relevance measure in heterogeneous networks. IEEE Trans. Knowl. Data Eng. **26**(10), 2479–2492 (2014)
14. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence **22**(8), 888–905 (2000)
15. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment **4**(11), 992–1003 (2011)
16. Sun, Y., Norick, B., Han, J., Yan, X., Yu, P.S., Yu, X.: Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. ACM Transactions on Knowledge Discovery from Data (TKDD) **7**(3), 11 (2013)
17. Wang, C., Dong, X., Zhou, F., Cao, L., Chi, C.H.: Coupled attribute similarity learning on categorical data. IEEE transactions on neural networks and learning systems **26**(4), 781–797 (2015)