GIN: A Clustering Model for Capturing Dual Heterogeneity in Networked Data

Jialu Liu[1], Chi Wang[2], Jing Gao[3], Quanquan Gu[4], Charu Aggarwal[5], Lance Kaplan[6], and Jiawei Han[1]

[1]University of Illinois at Urbana-Champaign
[2]Microsoft Research Redmond
[3]University at Buffalo
[4]University of Virginia
[5]IBM T. J. Watson Research Center
[6]U.S. Army Research Laboratory

Abstract

Networked data often consists of interconnected multi-typed nodes and links. A common assumption behind such heterogeneity is the shared clustering structure. However, existing network clustering approaches over-simplify the heterogeneity by either treating nodes or links in a homogeneous fashion, resulting in massive loss of information. In addition, these studies are more or less restricted to specific network schemas or applications, losing generality. In this paper, we introduce a flexible model to explain the process of forming heterogeneous links based on shared clustering information of heterogeneous nodes. Specifically, we categorize the link generation process into binary and weighted cases and model them respectively. We show these two cases can be seamlessly integrated into a unified model. We propose to maximize a joint log-likelihood function to infer the model efficiently with Expectation Maximization (EM) algorithms. Experiments on real-world networked data sets demonstrate the effectiveness and flexibility of the proposed method in fully capturing the dual heterogeneity of both nodes and links.

1 Introduction

Many real-world data can be represented as a network, which is composed of nodes interconnected with each other via meaningful links. Examples include friendship networks in Facebook, web pages connected by hyperlinks, and co-author networks in the bibliographic data. Mining networked data has attracted wide attention in recent years [13, 1, 12] not only because of the prevalence of networked data, but also due to the fact that network structure is very useful for exploiting the intrinsic characteristics of data. Among these mining tasks, clustering on a network aims at partitioning the nodes into connected subnetworks hardly or softly such that nodes within the same cluster have more connections.

Limitations of Existing Work. Many of the existing studies on clustering networked data focus on homogeneous [12, 15, 1] or bipartite networks [4, 7]. In homogeneous networks, nodes and links are of a single type. For example, in co-author networks, authors are connected by their collaborations. In bipartite networks, although there exist two types of nodes, the links only have one type–Each link must go from one type of node to the other type of node. Obviously, homogeneous and bipartite networks cannot cover all the complicated networked data in real world. It is common that in real networks multiple types of nodes are connected by multiple types of links, forming a network with dual heterogeneity of both nodes and links. For example, networked data extracted from Twitter contains multi-typed objects including tweets, users and places. Various types of links are formed between different types of nodes, such as user follow relationship (binary link), user-words publish relationship (aggregated link) and user-location connections (aggregated link). When analysing on such networked data, it is important to maintain the variety of node or link types in the network as each type has its unique characteristics.

Efforts have been devoted to developing effective clustering approaches to characterize the heterogeneity of networked data recently [18, 20, 19, 3, 16]. RankClus [18] focuses on clustering discovery on top of a specific topological network structure called bi-typed network and NetClus [20] focuses on the network with star schema. PathSelClus [19] detects clusters for a particular object type by analyzing aggregated links extracted from a collection of paths, each of which encodes a sequence of relations among the target type and the other object types. Some other multi-typed network clustering approaches [3, 16] study a specific domain, i.e., topic modeling in linked text data, and cannot be easily applied to the clustering task on general data types. None of these works is general enough to cover any schema, any type and any application. Moreover, these heterogeneous methods oversimplify the heterogeneity of links. These two limitations motivate us to develop a general model to fully capture both node and link types, covering a wide spectrum of scenarios in clustering heterogeneous networks.

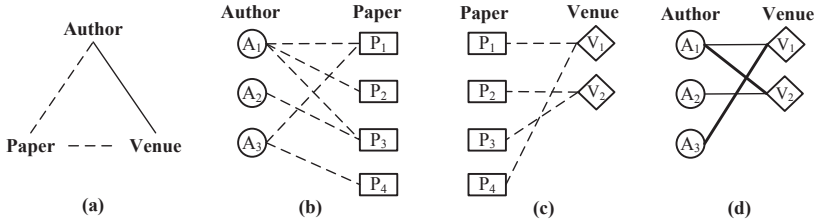

Figure 1: A simple network explaining binary/weighted links. (a) network schema; (b) binary links between authors and papers; (c) binary links between papers and venues; (d) weighted links between authors and venues.

Proposed Model. In this paper, we aim to find a general clustering solution from the schema-free heterogeneous networks (undirected), in which each cluster consists of multiple types of nodes and their corresponding links. The major challenge in designing this model is how to take into account the rich heterogeneity of both nodes and link types which have not been studied thoroughly in the literature.

Fig. 1 (a) is an example of a bibliography network showing such heterogeneity, in which we may have three types of nodes, i.e., authors, papers and venues, and three types of links defined based on the connections between them. Despite the wide variety of link types, these links fall into two major categories depending on whether these links are observed to be binary or not. In the figure, both author-paper and paper-venue links are binary in the sense that each author is either in a paper or not, and each paper is published in one venue (link weight is 0 or 1). In contrast, author-venue links are derived or aggregated from author-paper and paper-venue links–An author may appear multiple times in the same venue as one can publish different papers (link weight is an integer). Previous studies [20, 19] tend to model these two categories of links altogether without differentiating them, which fail to extract their unique characteristics and loses clustering power correspondingly.

In contrast, we propose two generative processes, each of which describes the formation of one of the link categories. We then integrate these two models into a unified model, based on which we obtain a joint log-likelihood function to be maximized. Efficient EM update equations are derived to infer both clustering assignment and model parameters. The contributions of this paper are:

- The proposed approach provides a unified solution to study different kinds of subnetworks by analysing dual heterogeneity of both nodes and links, outperforming existing methods in all four real data sets.
- The clustering framework is general in terms of the schema. The learning framework is easy to implement with an optimized time complexity proportional to the size of links and nodes.
- In addition to non-zero links, we propose to sample and model a set of non-linked node pairs, which is demonstrated to help improve performance.

2 Preliminaries

In this section, we define the problem of clustering in heterogeneous networks and introduce related concepts and necessary notations.

Definition 1. ***Information Network***. *An information network comprises objects from T types $\mathcal{X} = \{X^{(t)}\}_{t=1}^{T}$, where $X^{(t)}$ is a set of objects belonging to the $t-th$ type. Such a network with different types of objects can be denoted as a graph $G = (\mathcal{X}, \mathcal{E})$, where $\mathcal{X}$ is a set of nodes representing different typed objects and $\mathcal{E}$ is a set of links representing relations between objects. $W : \mathcal{E} \to \mathbb{R}_0^+$ is a weight mapping from a link $e \in \mathcal{E}$ to a non-negative real number $w \in \mathbb{R}_0^+$. Specially, a network with the number of object types $T = 1$ is called **homogeneous network**; it is called **bipartite network** when $T = 2$ and links only exist between two object types; it is called **heterogeneous network** otherwise.*

Note that as each link $e \in \mathcal{E}$ can be connected to at most two object types, we view heterogeneous network as the combinations of homogeneous and bipartite networks in this paper. In particular, we use $E^{(uv)}$ to denote the set of all links in subnetwork $G^{(uv)}$ between object type u and v including zero-weighted links. This subnetwork could be either bipartite if $u \neq v$ or homogeneous if $u = v$. Other notations are in Table 1.

We formalize our clustering problem as: given an undirected heterogeneous network G and the number of clusters K, the task is to find K partitions $C_1, C_2, \cdots, C_K$ on nodes $\mathcal{X}$ of all types such that nodes within the same cluster have more links than nodes in different clusters, satisfying $\cup_{k=1}^{K} \mathcal{X}(C_k) = \mathcal{X}(G)$.

3 Modeling Dual Heterogeneity

In this section, we propose a flexible model for clustering networked data following the intuition that if a pair of nodes have more links of the same cluster, they will

Table 1: List of notations

Notation	Explanation
G	graph denoting the information network
$\mathcal{X}$	set of nodes in G
$\mathcal{E}$	set of links in G
N	number of nodes in G
M^+	number of non-zero links in G
K	number of clusters in G
$X^{(t)}$	set of nodes of object type t
$x_i^{(t)}$	node x_i of object type t
$M_i^{(t)}$	sum of link weights connected to $x_i^{(t)}$
$\theta_{ik}^{(t)}$	cluster membership of node $x_i^{(t)}$
$\sigma_i^{(t)}$	popularity of node $x_i^{(t)}$
Θ	parameter set for $\theta_{ik}^{(t)}$'s over all object types
Σ	parameter set for $\sigma_i^{(t)}$'s over all object types
$G^{(uv)}$	subnetwork between object type u and v
$\alpha^{(uv)}$	link strength of $G^{(uv)}$
$e_{ij}^{(uv)}$	link between $x_i^{(u)}$ and $x_j^{(v)}$
$E^{(uv)}$	set of all links in $G^{(uv)}$
$N^{(uv)}$	number of nodes in $G^{(uv)}$
$M^{(uv)}$	sum of link weights in $G^{(uv)}$
$M^{(uv)}+$	number of non-zero links in $G^{(uv)}$
$W_{ij}^{(uv)}$	weight of link $e_{ij}^{(uv)}$

have greater probability to get connected; and a node will be more likely to belong to a cluster if more of its connected links are in that cluster. Specifically, we study two categories of links which are commonly seen in heterogeneous networks as introduced in Fig. 1, i.e., binary and weighted links, and derive two generative models separately. Later they are integrated to a unified model capturing heterogeneity of both nodes and links.

3.1 Subnetworks with Binary Links For this category of subnetworks, a binary link indicates the existence of the connection between two nodes. That is to say, the observed link weight $W_{ij}^{(uv)}$ between two nodes $x_i^{(u)}$ and $x_j^{(v)}$ is either 1 or 0.

We propose to simply factorize the expectation of observing a link into the cluster assignments of two end nodes. Assume now we are given a network denoted as $G^{(uv)}$ (can be homogeneous or bipartite network depending on whether u equals v). Then the probability of a link between nodes $x_i^{(u)}$ and $x_j^{(v)}$ is $P(e_{ij}^{(uv)}=1)$. Specifically, we factorize $P(e_{ij}^{(uv)}=1)$ into $\sum_{k=1}^{K}\theta_{ik}^{(u)}\theta_{jk}^{(v)}$ where $\{\theta_{ik}^{(u)}\}_{k=1}^{K}$ is a vector with length K indicating the cluster membership of node $x_i^{(u)}$. This factorization implies that two nodes get connected more easily if they share the same cluster distribution. The underlying generative process for link $e_{ij}^{(uv)}$ is as follows:

$$e_{ij}^{(uv)} \sim Bernoulli(\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}).$$

For the whole set of binary links $E^{(uv)}$, the following likelihood can be derived to estimate parameters:

$$\prod_{i<j}\Big(P(e_{ij}^{(uv)}=1)\Big)^{W_{ij}^{(uv)}}\Big(P(e_{ij}^{(uv)}=0)\Big)^{1-W_{ij}^{(uv)}}. \tag{3.1}$$

From Eq. 3.1 we note that the second term is to model all the non-linked node pairs. However, in many real world networks, nodes are usually not linked as the network is observed incomplete. Based on this fact, we can sample a small set of unlinked nodes. This can make the model less biased and also more efficient to be trained. Sec. 3.4 provides additional discussions on choosing these non-linked node pairs.

3.2 Subnetworks with Weighted Links Compared to binary links introduced in the last subsection, weighted links are usually derived as the counts of aggregation on meta-paths [19] where a meta-path is a path defined on object types. For example, aggregated links between two co-authors can be computed via "author-paper-author" path, "author-paper-venue-paper-author" path, and so on. Intuitively, different paths may carry different or even orthogonal semantics.

For ease of modeling, we first assume the weight of links in this case to be discrete. Later we are able to show that the objective function essentially can be optimized by relaxing the link weight to be nonnegative real number. In the rest of this subsection, we assume $W_{ij}^{(uv)}$ either to be 0 or greater than 0.

Similar to the Bernoulli setting in the previous subsection, we first model the existence of a link between a given pair of nodes. If the link exists, we adopt Poisson distribution to model the weight of the link. The expected total weight of links between two nodes $x_i^{(u)}$ and $x_j^{(v)}$ are also based on the summation of their matched cluster memberships. In addition to the cluster membership vector $\{\theta_{ik}^{(u)}\}_{k=1}^{K}$, we incorporate a scale parameter $\sigma_i^{(u)}$ for each node $x_i^{(u)}$ in consideration of the weighted setting. Then we can come up with the following nested generative process for weighted links:

$$\begin{aligned} &\text{(a) } e_{ij}^{(uv)} \sim Bernoulli(\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}) \\ &\text{(b) If } e_{ij}^{(uv)}=1, \quad \omega_{ij}^{(uv)} \sim Poisson(\sigma_i^{(u)}\sigma_j^{(v)}\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}) \end{aligned} \tag{3.2}$$

where discrete random variable $\omega_{ij}^{(uv)}$ is the weight of the link. The value of $\sigma_i^{(u)}$ has an intuitive interpretation: it is the node popularity correlated with the total weights of connected links. For the convenience of representation, we require the $\sigma_i^{(u)}$ to be shared by the same node accross different subnetworks. This might not be correct in some cases where a node should not necessarily be popular in all different networks. For example, a tweet retweeted a lot may be short in length. Fortunately, this constraint can be easily relaxed by assuming the popularities to be partially shared on specified subnewtorks since the learning frameworks are still similar.

For the set of weighted links $E^{(uv)}$, the following

likelihood can be derived according to the definition of Bernoulli-Poisson generative process[1]:

$$\prod_{W_{ij}^{(uv)}=0} \big(1-\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}\big)\times \prod_{W_{ij}^{(uv)}>0} \big(\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}\big) \frac{\big(\sigma_i^{(u)}\sigma_j^{(v)}\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}\big)^{W_{ij}^{(uv)}}}{W_{ij}^{(uv)}!} \times e^{-\sigma_i^{(u)}\sigma_j^{(v)}\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}}. \tag{3.3}$$

where $\sum_{W_{ij}^{(uv)}=0}$ means iterating over all links with weight 0 and summing up following terms. Eq. 3.3 is similar to the previous binary setting computed over all possible pairs between nodes $x_i^{(u)}$ and $x_j^{(v)}$. In addition, in the Supplementary Material we further discover that this Bernoulli-Poisson-based model is closely connected to a Multinomial-based model.

3.3 The Unified Model In the previous two subsections we focused on modeling individual subnetworks with different categories of links. Now we move to propose a unified model such that information of different typed nodes and links can propagate in the heterogeneous networks effectively. The model learning will also be covered in this subsection.

To achieve this, we first define two sets of subnetworks belonging to the same heterogeneous network G: $\mathcal{B}$ and $\mathcal{W}$. They represent subnetworks having binary and weighted links respectively, satisfying that $\mathcal{B}\cup\mathcal{W}=G$ and $\mathcal{B}\cap\mathcal{W}=\varnothing$.

Then, via multiplying Eqs. 3.1 and 3.3 for all typed links in the heterogeneous networks, we get

$$\prod_{G^{(uv)}\in\mathcal{B}}\prod_{i<j}\big(\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}\big)^{W_{ij}^{(uv)}}\big(1-\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}\big)^{1-W_{ij}^{(uv)}} \times \prod_{G^{(uv)}\in\mathcal{W}}\prod_{W_{ij}^{(uv)}=0}\big(1-\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}\big) \times \prod_{W_{ij}^{(uv)}>0}\big(\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}\big)\frac{\big(\sigma_i^{(u)}\sigma_j^{(v)}\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}\big)^{W_{ij}^{(uv)}}}{W_{ij}^{(uv)}!} \times e^{-\sigma_i^{(u)}\sigma_j^{(v)}\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}}. \tag{3.4}$$

It is worth noting that in this unified model, for each object type u we use parameter set $\Theta^{(u)}$ and $\Sigma^{(u)}$ to describe involved nodes. These parameters are shared across all subnetworks where these nodes appear. Via multiplication between different subnetworks, we can learn the parameter by collectively optimizing the objectives of all subnetworks. In this way, our model can utilize both compatible and complementary perspectives of different subnetworks. For example, in an "author-venue-term" network, we have an author A submit to SDM with one set of terms, forming an "author-venue-term" hyper-link. When A submits to SIGMOD, a different set of terms is used, which forms another hyper-link. In this case, our unified likelihood can be considered as maximizing the joint probability of these hyper-links[2]. Of course, our model is more complicated in the sense that we additionally model non-linked node pairs. It is worthy of mentioning here that we spend efforts to lower the computational challenge brought by these additional pairs in the next subsection.

To directly optimize the log-likelihood of Eq. 3.4 is difficult since the cluster assignments for each link are latent, making logarithm appear before the summation. To solve this, one of the most intuitive and effective way is to use EM algorithm to iteratively determine the cluster distribution of the links and maximize the expected log-likelihood. Specifically, we use $\phi_{ijk_1k_2}^{(uv)}$ to denote the posterior probability of an unobserved link generated from $x_i^{(u)}$ and $x_j^{(v)}$ with the constraint that these two nodes' cluster assignments are different, i.e., $k_1\neq k_2$. Meanwhile, we use $\psi_{ijk}^{(uv)}$ to denote the posterior probability of a link resulted from the same cluster assignments of two end nodes. These two posterior distributions are defined for two disjoint set of links (i.e., linked and non-linked node pairs).

These two sets of posterior probabilities are helpful to transform the above log-likelihood function to the following *expected complete log-likelihood* function of the whole heterogeneous network using Jensen's inequality:

$$\begin{aligned}\mathcal{L}(\Theta,\Sigma) = &\sum_{G^{(uv)}\in\mathcal{B}}\sum_{W_{ij}^{(uv)}=1}\sum_k \psi_{ijk}^{(uv)}\log\theta_{ik}^{(u)}\theta_{jk}^{(v)} \\ &+\sum_{G^{(uv)}\in\mathcal{W}}\sum_{W_{ij}^{(uv)}>0}\big(W_{ij}^{(uv)}+1\big)\sum_k \psi_{ijk}^{(uv)}\log\theta_{ik}^{(u)}\theta_{jk}^{(v)} \\ &+\sum_{G^{(uv)}\in G}\sum_{W_{ij}^{(uv)}=0}\sum_{k_1\neq k_2}\phi_{ijk_1k_2}^{(uv)}\log\theta_{ik_1}^{(u)}\theta_{jk_2}^{(v)} \\ &+\sum_{G^{(uv)}\in\mathcal{W}}\sum_{W_{ij}^{(uv)}>0}W_{ij}^{(uv)}\log\sigma_i^{(u)}\sigma_j^{(v)} \\ &-\sum_{G^{(uv)}\in\mathcal{W}}\sum_{W_{ij}^{(uv)}>0}\sigma_i^{(u)}\sigma_j^{(v)}\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}.\end{aligned} \tag{3.5}$$

Note that to obtain the third line we apply a trick that $1-\sum_k \theta_{ik}^{(u)}\theta_{jk}^{(v)}=\sum_{k_1\neq k_2}\theta_{ik_1}^{(u)}\theta_{jk_2}^{(v)}$.

In the EM algorithm, the E-step can be viewed as computing the posterior probabilities (ϕ and ψ) using current estimation of Θ. With the Bayes' theorem, we compute $\phi_{ijk_1k_2}^{(uv)}$ for each non-linked node pair over all

[1] We assume zero-weighted links do not go through to step (b).

[2] For weighted subnetworks, a weighted link is split into weighted child-links and they are modeled separately for different hyper-links. In this case, the update function is a little different from this paper but performs similarly.

$K(K-1)$ cases where $k_1 \neq k_2$:

$$\phi_{ijk_1k_2}^{(uv)} = \frac{\theta_{ik_1}^{(u)}\theta_{jk_2}^{(v)}}{\sum_{l_1\neq l_2}\theta_{il_1}^{(u)}\theta_{jl_2}^{(v)}}. \tag{3.6}$$

Similarly, as $\psi_{ijk}^{(uv)}$ is restricted to the same cluster assignments between two linked nodes, we have

$$\psi_{ijk}^{(uv)} = \frac{\theta_{ik}^{(u)}\theta_{jk}^{(v)}}{\sum_{l}\theta_{il}^{(u)}\theta_{jl}^{(v)}}. \tag{3.7}$$

We can see that node popularity Σ is not involved in the E-step and thus can be computed after EM.

The M-step is to estimate parameters Θ by maximizing the expected complete log-likelihood with precomputed ϕ and ψ in the E-step. This step is more complicated as nodes in our network can be connected by two categories of links simultaneously.

Before we try to derive the update function for $\theta_{ik}^{(u)}$, we first take a look at the update function of $\sigma_i^{(u)}$ even though it is computed outside the EM-steps. We will see that the equation for $\sigma_i^{(u)}$ helps computing $\theta_{ik}^{(u)}$.

Taking the derivative of $\mathcal{L}$ with respect to $\sigma_i^{(u)}$ results in the following equation:

$$\frac{\sum_{G^{(uv)}\in\mathcal{W}}\sum_j W_{ij}^{(uv)}}{\sigma_i^{(u)}} = \sum_{G^{(uv)}\in\mathcal{W}}\sum_{W_{ij}^{(uv)}>0}\sigma_j^{(v)}\sum_k\theta_{ik}^{(u)}\theta_{jk}^{(v)}. \tag{3.8}$$

An intuitive interpretation behind this equation is that the popularity of a node is decided by two factors: its own node degree and effective popularities of other nodes towards it. The popularity is large if the node's degree is large and other nodes in the network do not contribute much in this node's potential links. Considering all nodes in the weighted network, we have a nonlinear system where each equation in the system is similar to Eq. 3.8 but for different nodes. Newton's method can be applied to iteratively find the numerical solutions, which is time consuming. We will introduce a way to lower time complexity in the Supplementary Material by relaxing the constraints in Eq. 3.8.

Nevertheless, if we aggregate both sides of Eq. 3.8 over all nodes, we can get

$$\sum_{G^{(uv)}\in\mathcal{W}}\sum_{i<j}W_{ij}^{(uv)} = \sum_{G^{(uv)}\in\mathcal{W}}\sum_{W_{ij}^{(uv)}>0}\sigma_i^{(u)}\sigma_j^{(v)}\sum_k\theta_{ik}^{(u)}\theta_{jk}^{(v)}. \tag{3.9}$$

The right-hand side expression of the above expression is exactly the last term in Eq. 3.5.

It is clear now that only the first three terms in $\mathcal{L}$ are related to $\theta_{ik}^{(u)}$. Taking the derivative of $\mathcal{L}$ with respect to $\theta_{ik}^{(u)}$ under the constraints that $\sum_k\theta_{ik}^{(u)}=1$ and set the derivative to 0, we get the closed-form equations as follows:

$$\begin{aligned}\theta_{ik}^{(u)} \propto & \sum_{G^{(uv)}\in\mathcal{B}}\sum_{W_{ij}^{(uv)}=1}\psi_{ijk}^{(uv)} \\ & + \sum_{G^{(uv)}\in\mathcal{W}}\sum_{W_{ij}^{(uv)}>0}\left(W_{ij}^{(uv)}+1\right)\psi_{ijk}^{(uv)} \\ & + \sum_{G^{(uv)}\in G}\sum_{W_{ij}^{(uv)}=0}\sum_{l\neq k}\phi_{ijkl}^{(uv)}.\end{aligned} \tag{3.10}$$

Note that Eqs. 3.6, 3.7 and Eq. 3.10 are iteratively updated for all typed links and nodes until the log-likelihood L converges to the local optimum.

Combining both the E- and M-steps, we observe the following rule: if a pair of nodes $x_i^{(u)}$ and $x_j^{(v)}$ are (resp., are not) observed to be connected by a link, $\theta_{ik}^{(u)}$ and $\theta_{jk}^{(v)}$ for these two nodes will become larger (resp., smaller), resulting in the likelihood increasing. This exactly meets our expectation that if a pair of nodes have more links of the same cluster, they will have greater probability to get connected.

Until now we have modeled all subnetworks without considering their relative importance. That is to say the link strengths of different subnetworks are equivalent. To support weighting, one can add coefficient $\alpha^{(uv)}$ for each subnetwork:

$$\begin{aligned}\theta_{ik}^{(u)} \propto & \sum_{G^{(uv)}\in\mathcal{B}}\alpha^{(uv)}\sum_{W_{ij}^{(uv)}=1}\psi_{ijk}^{(uv)} \\ & + \sum_{G^{(uv)}\in\mathcal{W}}\alpha^{(uv)}\sum_{W_{ij}^{(uv)}>0}\left(W_{ij}^{(uv)}+1\right)\psi_{ijk}^{(uv)} \\ & + \sum_{G^{(uv)}\in G}\alpha^{(uv)}\sum_{W_{ij}^{(uv)}=0}\sum_{l\neq k}\phi_{ijkl}^{(uv)}.\end{aligned} \tag{3.11}$$

We introduce how to choose proper strength for each subnetwork in the Supplementary Material.

3.4 Complexity Analysis For each node or non-zero link in the whole network, a vector of parameters $\theta_{ik}^{(u)}$'s or $\psi_{ijk}^{(uv)}$'s with length K is used to keep the values. Just considering this part of the network we have M^+K+NK parameters to estimate where M^+ is the size of linked node pairs. The time complexity for updating this part in each iteration is $O(M^+K+NK)$.

As for the unobserved links in the network the spatial/time complexity increases significantly if we need to go over all $N\times N-M^+$ possible links and $K(K-1)$ cluster combinations. One of our strategies to alleviate such burden is to sample a potential neighbourhood for each node. This is reasonable because unobserved links do not always imply cluster inconsistency and modeling them all can incorporate unnecessary noises. We keep all the non-zero links and sample ηM^+ unobserved links to make its size proportional to the total number of links M^+ (we choose $\eta=0.1$ in the experiments).

Algorithm 1 Cost-optimized Model

Input: Heterogeneous Network G, number of clusters K, weight vectors α, sample coefficient η
Output: Clustering of nodes in G

```
initialize {θ_ik^(u)}^K randomly
sample ηM^(uv)+ non-linked node pairs
allocate and reset {θ'_ik^(u)}^K for all types of nodes
repeat
  foreach subnetwork G^(uv) in G
    foreach non-zero link between nodes x_i^(u) x_j^(v)
      compute {ψ_ijk^(uv)}^K as in Eq. 3.7
      if G^(uv) ∈ B
        multiply this vector by scalar α^(uv)
      else
        multiply by scalar α^(uv)(W_ij^(uv) + 1)
      add to {θ'_ik^(u)}^K and {θ'_jk^(v)}^K
    foreach non-linked node pairs x_i^(u) x_j^(v)
      compute {Σ_{l≠k} φ_ijkl^(uv)}^K as in Eq. 3.12
      multiply this vector by scalar α^(uv)
      add this scaled vector to vectors {θ'_ik^(u)}^K
      repeat line 14-16 for {θ'_jk^(v)}^K
  foreach node x_i^(u) over each type u
    normalize vector {θ'_ik^(u)}^K s.t. Σ_k θ'_ik^(u) = 1
  copy {θ'_ik^(u)}^K to {θ_ik^(u)}^K and reset {θ'_ik^(u)}^K
until L(Θ, Ω) converges
compute σ_i^(u) if interested
```

Another effective strategy is for every non-linked node pair, we only compute $\{\sum_{l\neq k}\phi_{ijkl}^{(uv)}\}^K$ where

$$\sum_{l\neq k}\phi_{ijkl}^{(uv)} = \frac{\sum_{l\neq k}\theta_{ik}^{(u)}\theta_{jl}^{(v)}}{\sum_{l_1\neq l_2}\theta_{il_1}^{(u)}\theta_{jl_2}^{(v)}} = \frac{\theta_{ik}^{(u)} - \theta_{ik}^{(u)}\theta_{jk}^{(v)}}{1-\sum_l \theta_{il}^{(u)}\theta_{jl}^{(v)}}. \tag{3.12}$$

In this way, we actually compute the aggregations of $\phi_{ijkl}^{(uv)}$ instead of every individual one. This helps to lower the computational cost of each sampled non-linked node pairs from $O(K(K-1))$ to $O(2K)$ considering the nonsymmetric property of Eq. 3.12.

After combining these two types of networks, our aggregated complexity is $O((1+\eta)M^+K + NK)$ for each iteration. Alg. 1 explains the detailed procedures.

4 Experiments

In this section, several experiments were conducted to show the effectiveness of the proposed approach.

4.1 Data Sets Four real world data sets were used. The statistics of these data sets are summarized in Table 2 and their schemas are shown in Fig. 2:

- The **DBLP** data set is a collection of bibliographic information on computer science publications. We use a subset of its records that belong to four research areas. This network follows star schema: paper is linked to author, venue and term[3].
- The **4Groups** data set is a bi-typed network extracted from the above DBLP data set. Both co-author and author-term[4] relationships are kept in it and researchers are selected from four data mining and machine learning research groups.
- The **Flickr** data set is a network containing three types of objects: image, user and tag. Links exist between image-user and image-tag.
- The **NSF** data set is a subset of NSF Research Awards Abstracts describing NSF awards for basic research from 1990 to 2003. We use a subset of documents associated with terms and investigators that belong to the largest 10 research programs.

4.2 Compared Algorithms To demonstrate the effectiveness of the proposed approach, we compared the following clustering algorithms:

- A Generative Model for Heterogeneous Information Networks (**GIN**). This is the proposed algorithm. Non-linked node pairs are sampled with a budget of 10% of total observed links. Link strengths are set automatically.
- **NetClus** [20]. It is a rank-based algorithm proposed recently to integrate ranking and clustering together for networks with star schema.
- Spectral Clustering for Heterogeneous Information Networks (**SCIN**). We derived this algorithm by extending the work in [4] to the heterogeneous networks following the strategy in [8].
- Standard Spectral Clustering (**SC**), a spectral-based algorithm [14] which is designed to segment graphs and is shown to be effective on networks.
- A Poisson Model for Homogeneous Information Networks (**PHIN**). This work [2] is recently proposed to cluster homogeneous network data.

Note that the first three algorithms are proposed for heterogeneous networks directly, among which NetClus can only be applied to networks following star schema. As a good initialization is important for EM algorithm, we use the output of SCIN to set initial values of Θ in GIN. This strategy is also applied to NetClus and shows to be effective. PHIN and SC are both homogeneous network algorithms, in which all nodes and links of different types are modelled equally.

[3] Terms are extracted from paper titles and the links are binary.
[4] For author-term subnetwork, link weights are count of terms aggregated from papers published by the same author.

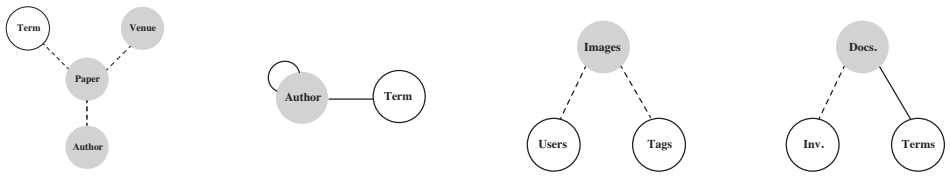

Figure 2: Network schemas of all data sets in which circles of labelled object types are in grey. Dashed (resp., solid) lines refer to binary (resp., weighted) links.

Table 2: Statistics of the four data sets

Data set	DBLP	4Groups	Flickr	NSF
#Nodes	70,536	1,618	4,076	30,995
#Links	332,388	5,568	14,396	1,883,682
Sparsity	6.7e-5	2.1e-3	8.7e-4	2.0e-3
#Clusters	4	4	8	10
#Objects	4	2	3	3
#Subnet.	3	2	2	2
Link Cat.	Binary	Weighted	Binary	Fused

Table 3: Clustering accuracy on the four data sets (%)

Data set	DBLP				4Groups	Flickr	NSF
Object	Author	Paper	Venue	Average	Author	Image	Doc.
GIN	**93.01**	**84.75**	**100.00**	**92.85**	**97.16**	**48.44**	**74.48**
NetClus	89.90	80.00	**100.00**	89.72	-	44.94	70.42
SCIN	86.26	81.00	90.00	86.16	89.89	42.12	72.29
SC	46.03	41.00	30.00	45.84	56.14	37.74	44.62
PHIN	75.71	63.00	60.00	75.35	62.28	43.97	61.95
#Labels	4,236	100	20	-	99	1,028	10,606

4.3 Results We are mainly interested in whether our proposed algorithm can boost the performance of clustering nodes by modeling different types of nodes and links. Therefore, we first compare the overall clustering performance between the proposed method and baseline approaches quantitatively. Then both case and parameter studies are offered to help better understand our algorithm.

4.3.1 Quantitative Analysis For our proposed GIN, the estimated Θ is used to infer the cluster label of each node. The prediction is simply the index of its maximum value among all cluster candidates. The clustering performance is evaluated by comparing the predicted labels with the ground truth labels provided by data sets. *Accuracy* is used to measure the performance, which is defined to be the ratio of correctly predicted nodes by the total number of nodes.

The ground truth labels are obtained in different ways for our four data sets. Firstly for DBLP, we manually label a subset of nodes according to the domain knowledge. Since 4Groups is extracted from DBLP, we adopt the same procedure to obtain labels of authors. For Flickr data set, group information of images is crawled from the website and is adopted as the ground truth. Finally for NSF data set, different research programs for documents are viewed as the cluster assignments. Sizes of labelled nodes for each data set are listed in the bottom line of Table 3. Their corresponding object types are filled in grey as in Fig. 2. Although there is only one object type labelled for some data sets, we expect to demonstrate the superiority of GIN in heterogeneous network clustering.

Table 3 shows the clustering performance of different algorithms on the four data sets where 20 test runs were conducted and the average performance is reported. From Table 3, one can easily observe GIN outperforming all other competitors. One reason is that we explicitly model different categories of links and different kinds of object types. Then the generative processes of different subnetworks are seamlessly integrated into a unified model. Additionally, GIN samples non-linked node pairs to help infer cluster assignments and train the model. Among other competitors, NetClus is no better than SCIN, owing to the reason that NetClus is originally designed for the bibliographic data and is not general enough. For the rest two algorithms, i.e., PHIN and SC, they perform relatively poor as expected because they ignore the type difference of nodes and links in heterogeneous networks.

On the DBLP data set, as this heterogeneous network is nicely structured and clean, GIN, NetClus and SCIN all achieve outstanding performance, which implies the data set to be a relatively easy task for the heterogeneous clustering algorithms. Nevertheless, GIN still outperforms the second best algorithm by 3% and get 92.85% accuracy on average, which is an achievement considering the limited room for improvement.

The 4Groups data set is a bi-typed network comprising two aggregated subnetworks, and NetClus cannot be applied as it is restricted to networks with star schema. For this data set we only show the results for authors because there is no available label for terms. The same strategy is used for the remaining two data sets. In general, we can observe similar results as the DBLP experiment: the gap between homogeneous and heterogeneous algorithms are still significant and GIN performs the best. Specifically, GIN outperforms SCIN by 7.27%, SC by 41.02% and PHIN by 34.88%.

For the Flickr data set, the results are slightly different. We observe that PHIN achieves similar to NetClus and SCIN. This implies the heterogeneity are not utilized well, which reflects the difficulty of this data set. Correspondingly, GIN fits the data well and beats

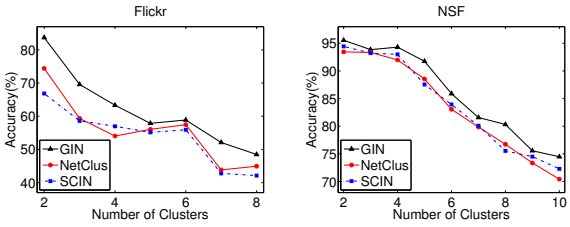

Figure 3: Clustering performance on Flickr and NSF.

the second-best algorithm by 3.5%.

On the NSF data set, all three heterogeneous algorithms are relatively close in their performance. This may be led by the limitation of heterogeneity in the network. Recall that in the DBLP data set, we have more object types and they altogether help enhance the performance. Another reason is that the room left for potential improvement is limited considering the number of clusters and the accuracy we have achieved.

4.3.2 Case Study To further examine the behavior of these three heterogeneous algorithms, we chose Flickr and NSF data sets and conducted a thorough study since they have more clusters than the others. In the previous quantitative experiment listed in Table 3, we have tested for the clustering performance of all clusters. Here we show the accuracies of the three algorithms with varying number of clusters in Fig. 3.

We notice that tasks become more difficult when the number of clusters increases and consequently the corresponding performance of all algorithms drops. However, we can see that GIN is almost always the best among all the three algorithms, which demonstrates the effectiveness and stability of our algorithm. NetClus performs close to SCIN in most cases. The gap between these two algorithms and GIN implies that besides modeling different types of nodes, links should also be treated differently and modeling non-linked node pairs help.

4.3.3 Parameter Study One parameter in our model is the sample size of non-linked node pairs. In previous section we suggest to sample non-linked node pairs proportional to the total size of non-zero links M^+. To find the best sample proportion η, we have conducted the experiments of comparing performance of different settings in Fig. 4. We have tested various values of η in the range of $[10^{-3}, 10^0]$. As we can see, GIN performs relatively stable and consistent with respect to this parameter. When η is small, the effect of non-linked node pairs is limited in order to help estimate Θ. When its value is too large, a lot of noises are incorporated and affect the performance. On these four data sets, GIN achieves the best performance when η is around 0.1.

In Fig. 4, the running time for each iteration is also provided w.r.t. the sample proportion η. As

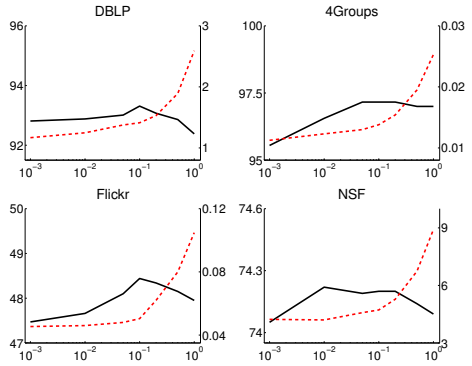

Figure 4: Clustering accuracy and running time (in seconds) of GIN v.s. sample proportion η. Dashed (resp., solid) lines refer to running time (resp., accuracy).

one can see, the time complexity doubles when η is 1, which is expected according to our complexity analysis. The quadratic curve does not imply quadratic time complexity as the horizontal axis is in log scale.

5 Related Work

Clustering on homogeneous networks and graphs has been widely studied in the literature. Previous work includes spectral clustering-based methods [15], modularity-based algorithms [12], density-based algorithms [22], matrix factorization [5] and probabilistic models [1, 9]. For bipartite networks and graphs, BSGP [4] is proposed to solve the co-clustering problem using singular vectors of a scaled adjacency matrix between two object types. PLSA [7] is designed as a topic modeling method for document-word concurrence matrix assuming documents are mixtures of topics. NMF [10] tackles this problem from a view of matrix factorization and is proved to be equivalent with PLSA. All these methods are based on the assumption that links exist only within the same or bi-typed typed nodes.

Some newly proposed clustering algorithms are designed for heterogeneous networks. RankClus [18] combines ranking and clustering in order to perform more accurate analysis in the network with two object types. NetClus [20] was proposed as a ranking-based algorithm extending RankClus [18] to the networks with star schema. GenClus [17] is designed for modeling node attributes together with directed networks. In [19], the authors choose to keep useful information in extracted relationships encoded by meta-paths and then integrate meta-path selection with user-guided information to cluster nodes in networks. In [11], clustering is obtained through spectral methods but partitions for different typed nodes are not one-to-one associated, i.e., some cluster association matrices are used to describe the relationship between partitions on

different object types. In [6], the authors follow BSGP [4] and treat the problem as the fusion of multiple pairwise co-clustering subproblems with the constraint of the star schema. Some other work such as [3, 16, 21] uses multi-typed nodes but only focus on a specific domain like topic modeling or recommendation.

6 Conclusions

In this paper, we proposed a unified generative model to characterize the formation of various links in a heterogeneous network with shared clustering structure. In order to model dual heterogeneity of both nodes and links without the schema constraint, we divided the types of links into binary and weighted categories. Accordingly, we proposed two different generative models that describe the link formation processes. We demonstrated how these two generative processes can be integrated into a unified model that can capture the overall formation of a multi-typed heterogeneous network. To infer the model and the hidden clustering structure, we developed an efficient EM-based approach to maximize the log likelihood function. We also demonstrate that sampling non-linked node pairs can help improve the performance. Experimental results on four real network data showed strong power of the proposed model.

7 Acknowledgements

Research was sponsored in part by the Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), the Army Research Office under Cooperative Agreement No. W911NF-13-1-0193, National Science Foundation IIS-1017362, IIS-1320617, and IIS-1354329, HDTRA1-10-1-0120, and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *JMLR*, 9:1981–2014, 2008.

[2] B. Ball, B. Karrer, and M. Newman. Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103, 2011.

[3] H. Deng, J. Han, B. Zhao, Y. Yu, and C. X. Lin. Probabilistic topic models with biased propagation on heterogeneous information networks. In *KDD*, pages 1271–1279, 2011.

[4] I. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, pages 269–274, 2001.

[5] C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*, pages 606–610, 2005.

[6] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *KDD*, pages 41–50, 2005.

[7] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[8] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *KDD*, pages 1298–1306, 2011.

[9] B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[10] D. Lee and S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[11] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu. Spectral clustering for multi-type relational data. In *ICML*, pages 585–592, 2006.

[12] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[14] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888 –905, 1997.

[15] P. Smyth and S. White. A spectral clustering approach to finding communities in graphs. In *SDM*, volume 119, page 274, 2005.

[16] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *KDD*, pages 306–315, 2004.

[17] Y. Sun, C. C. Aggarwal, and J. Han. Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. *VLDB*, 5(5):394–405, 2011.

[18] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In *KDD*, pages 565–576, 2009.

[19] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu. Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. In *KDD*, pages 1348–1356, 2012.

[20] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, pages 797–806, 2009.

[21] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, pages 448–456, 2011.

[22] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger. Scan: a structural clustering algorithm for networks. In *KDD*, pages 824–833, 2007.