PLOS | ONE

RESEARCH ARTICLE

FctClus: A Fast Clustering Algorithm for Heterogeneous Information Networks

Jing Yang[1], **Limin Chen**[1,2]*, **Jianpei Zhang**[1]

1 Institute of Computer Science and Technology, Harbin Engineering University, Harbin, China, **2** Institute of Computer Science and Technology, Mudanjiang Teachers College, Mudanjiang, China

* chenlimin_clm@126.com

Abstract

It is important to cluster heterogeneous information networks. A fast clustering algorithm based on an approximate commute time embedding for heterogeneous information networks with a star network schema is proposed in this paper by utilizing the sparsity of heterogeneous information networks. First, a heterogeneous information network is transformed into multiple compatible bipartite graphs from the compatible point of view. Second, the approximate commute time embedding of each bipartite graph is computed using random mapping and a linear time solver. All of the indicator subsets in each embedding simultaneously determine the target dataset. Finally, a general model is formulated by these indicator subsets, and a fast algorithm is derived by simultaneously clustering all of the indicator subsets using the sum of the weighted distances for all indicators for an identical target object. The proposed fast algorithm, FctClus, is shown to be efficient and generalizable and exhibits high clustering accuracy and fast computation speed based on a theoretic analysis and experimental verification.

CrossMark
click for updates

OPEN ACCESS

Citation: Yang J, Chen L, Zhang J (2015) FctClus: A Fast Clustering Algorithm for Heterogeneous Information Networks. PLoS ONE 10(6): e0130086. doi:10.1371/journal.pone.0130086

Academic Editor: Enrique Hernandez-Lemus, National Institute of Genomic Medicine, MEXICO

Received: December 27, 2014

Accepted: May 15, 2015

Published: June 19, 2015

Copyright: © 2015 Yang et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

Introduction

Information networks are ubiquitous and include social information networks and DBLP bibliographic networks. Numerous studies on homogeneous information networks, which consist of a single type of data object, have been performed; however, little research has been performed on the clustering of heterogeneous information networks, which consist of multiple types of data objects. Clustering on a heterogeneous network may lead to better understanding the hidden structures and deeper meanings of the networks[1].

The star network schema is popular and important in the field of heterogeneous information networks. The star network schema includes one data object target type and multiple data object attribute types, whereby each relation is the target data objects and all attribute data objects linking to it.

Algorithms based on compatible bipartite graphs can effectively consider multiple types of relational data. Various classical clustering algorithms, such as algorithms based on semi-definite programming[2,3], algorithms based on information theory[4] and spectral clustering algorithms for multi-type relational data[5], have been proposed for heterogeneous data from the compatible point of view. These algorithms are generalizable, but the computational complexity of these algorithms is too great for use in clustering heterogeneous information networks.

Sun et al. presents an algorithm, NetClus[6], and a PathSim-based clustering algorithm[7] for clustering heterogeneous information networks. NetClus is effective for DBLP bibliographic networks, but the algorithm is not a general model for clustering other heterogeneous information networks; NetClus is not sufficiently stable. The concept behind NetClus is also used for clustering service webs[8,9]. The PathSim-based clustering algorithm requires a user guide, and the clustering quality reflects the requirements of users rather than the requirements of the network. ComClus[10] is a derivation algorithm of NetClus for use with hybrid networks that simultaneously include heterogeneous and homogeneous relations. NetClus and ComClus are not general and depend on the given application.

Dynamic link inference in heterogeneous networks[11] requires more accurate initial clustering. A high clustering quality is necessary for network analysis, but low computation speed is intolerable because of the large network scales involved. The accuracy of the LDCC algorithm[12] is improved, while both the heterogeneous and homogeneous data relations are explored. The CESC algorithm [13] is very effective for clustering homogeneous data using an approximate commute time embedding. A heterogeneous information network with a star network schema can transform into multiple compatible bipartite graphs from the compatible point of view. When the relation between any two nodes of the bipartite graph is presented with the commute time, the relation of both heterogeneous and homogenous data objects can be explored; the clustering accuracy can also be improved. The heterogeneous information networks are large but very sparse; therefore, the approximate commute time embedding of each bipartite graph can be quickly computed using random mapping and a linear time solver[14]. All of the indicator subsets in each embedding indicate the target dataset, and subsequently, a general model for clustering heterogeneous information networks is formulated based on all indicator subsets. All weighted distances between the indicators and the cluster centers in the respective indicator subsets are computed. All indicator subsets can be simultaneously clustered according to the sum of the weighted distances for all indicators for an identical target object. Based on the above discussion, an effective clustering algorithm, FctClus, which is based on the approximate commute time embedding for heterogeneous information networks, is proposed in this paper. The computation speed and clustering accuracy of FctClus are high.

Methods

Commute Time Embedding of the Bipartite Graph

Given two types of datasets, $X_0 = \{x_1^{(0)}, x_2^{(0)}, \cdots, x_{n_0}^{(0)}\}$ and $X_1 = \{x_1^{(1)}, x_2^{(1)}, \cdots, x_{n_1}^{(1)}\}$, the graph $G_b = \langle V, E\rangle$ is called a bipartite graph if $V(G_b) = X_0 \cup X_1$ and $E(G_b) = \{\langle x_i^{(0)}, x_j^{(1)}\}$, where $1 \le i \le n_0$, $1 \le j \le n_1$. $W_{n_0 \times n_1}$ is the relation matrix between X_0 and X_1, where the element w_{ij} is the edge weight between $x_i^{(0)}$ and $x_j^{(1)}$. Then, the adjacency matrix of the bipartite graph G_b can be denoted as

$$\tilde{W}_{n\times n} = \begin{bmatrix} 0 & W_{n_0\times n_1} \\ (W^T)_{n_1\times n_0} & 0 \end{bmatrix}$$

D_1 and D_2 are the diagonal matrices, where the diagonal element of D_1 is $d_i = \sum_{j=1}^{n_1} w_{ij}$ and the diagonal element of D_2 is $d_j = \sum_{i=1}^{n_0} w_{ij}$.

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$$

thus the Laplacian matrix of the bipartite graph G_b is $L = D - \tilde{W}$. L can be eigen-decomposed into $L = \Phi\Lambda\Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$ is a diagonal matrix composed of the eigenvalues of L and $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$, $\Phi = (\phi_1, \phi_2, \cdots, \phi_n)$ is an eigenmatrix and ϕ_i is an eigenvector corresponding to the eigenvalue λ_i. Let L^+ be a pseudo-inverse matrix of L and $L^+ = \sum_{i=2}^{n} \frac{1}{\lambda_i} \phi_i \phi_i^T$. The bipartite graph is also an undirected weighted graph. According to the literature[15], the commute time c_{ij} between nodes i and j of G_b can be computed by the pseudo-inverse matrix L^+.

$$c_{ij} = g_v(l_{ii}^+ + l_{jj}^+ - 2l_{ij}^+) = g_v(e_i - e_j)^T L^+ (e_i - e_j) \tag{1}$$

where l_{ij}^+ is the (i, j) element of L^+, $g_v = \sum w_{ij}$, e_i is a unit column vector in which the i-th element is 1; that is, $e_i = [\underset{1}{0}, \cdots, \underset{i-1}{0}, \underset{i}{1}, \underset{i+1}{0}, \cdots, \underset{n}{0}]^{\mathrm{T}}$.

According to the literature[15,16], the commute time c_{ij} between nodes i and j of G_b is

$$\begin{aligned} c_{ij} &= g_v(e_i - e_j)^T L^+ (e_i - e_j) \\ &= g_v(e_i - e_j)^T \Phi\Lambda^{-1}\Phi^T (e_i - e_j) \\ &= [\sqrt{g_v}\Lambda^{-1/2}\Phi^T(e_i - e_j)]^T [\sqrt{g_v}\Lambda^{-1/2}\Phi^T(e_i - e_j)] \end{aligned}$$

Thus, the commute time c_{ij} is the square pairwise Euclidean distance between the row vectors in the space $(\sqrt{g_v}\Lambda^{-1/2}\Phi^T)^T$ or the column vectors in the space $\sqrt{g_v}\Lambda^{-1/2}\Phi^T$ [13], $(\sqrt{g_v}\Lambda^{-1/2}\Phi^T)^T$ or $\sqrt{g_v}\Lambda^{-1/2}\Phi^T$ is called the commute time embedding of the bipartite graph G_b. c_{ij} is the average path length between two nodes rather than the shortest path between two nodes. Using the commute time for clustering the noisy data increases robustness and captures the complex clusters. Therefore clustering in the commute time embedding can also effectively capture the complex clusters. $\sqrt{g_v}\Lambda^{-1/2}\Phi^T$ is used in this paper. If a normal Laplacian matrix $L_n = D^{-1/2} L D^{-1/2}$ is used, the commute time embedding is $\sqrt{g_v}\Lambda^{-1/2}\Phi^T D^{-1/2}$ [13].

Approximate Commute Time Embedding of the Bipartite Graph

If directly computing $\sqrt{g_v}\Lambda^{-1/2}\Phi^T$ or $\sqrt{g_v}\Lambda^{-1/2}\Phi^T D^{-1/2}$, the process requires $O(n^3)$ time for the eigen-decomposition of the Laplacian matrix L or L_n. $n = n_0 + n_1$ is the number of nodes and s is the number of edges in the bipartite graph G_b. According to the literature[17], if the edges in G_b are oriented and

$$B(i,j) = \begin{cases} 1 & i \text{ is tail}, j \text{ is head} \\ -1 & i \text{ is head}, j \text{ is tail} \\ 0 & \text{others} \end{cases}$$

where i and j are nodes of G_b, then $B_{s\times n}$ is a directed edge-node incidence matrix. Using $\hat{W}_{s\times s}$ as a diagonal matrix whose entries are the edge weights, thus $L = B^T \hat{W} B$. Furthermore,

$$\psi = \sqrt{g_v}\hat{W}^{1/2} B L^+ \in R^{s\times n} \tag{2}$$

thus, ψ is the commute time embedding of the bipartite graph G_b, where the square root of the

commute time is the Euclidean distance between i and j in ψ because

$$\begin{aligned} c_{ij} &= g_v(e_i - e_j)^T L^+(e_i - e_j) \\ &= g_v(e_i - e_j)^T L^+ L L^+(e_i - e_j) \\ &= g_v(e_i - e_j)^T L^+ B^T \hat{W} B L^+(e_i - e_j) \\ &= [\sqrt{g_v}\hat{W}^{1/2} B L^+(e_i - e_j)]^T [\sqrt{g_v}\hat{W}^{1/2} B L^+(e_i - e_j)] \end{aligned}$$

According to the literature[18], given vectors $v_1, \cdots, v_n \in R^s$ and $\varepsilon > 0$, $Q_{k_r \times s}$ is a random matrix of row vectors, where $Q(i,j) = \pm 1/\sqrt{k_r}$ is equivalent when $k_r = O(\log n / \varepsilon^2)$. With probability 1−1 / n, at least

$$(1-\varepsilon)\| v_i - v_j \|^2 \le \| Qv_i - Qv_j \|^2 \le (1+\varepsilon)\| v_i - v_j \|^2 \tag{3}$$

for all pairs.

Therefore, given the bipartite graph G_b with n nodes and s edges, $\varepsilon > 0$, and a matrix $Y_{k_r \times n} = \sqrt{g_v} Q \hat{W}^{1/2} B L^+$ with probability of at least 1−1 / n:

$$(1-\varepsilon)c_{ij} \le \| Y(e_i - e_j) \|^2 \le (1+\varepsilon)c_{ij} \tag{4}$$

for any nodes $i, j \in G_b$, where $k_r = O(\log n / \varepsilon^2)$.

The proof of Eq (4) comes directly from Eq (2) and Eq (3). $c_{ij} \approx \|Y(e_i - e_j)\|^2$ with an error ε based on Eq (4). If directly computing $Y_{k_r \times n} = \sqrt{g_v} Q \hat{W}^{1/2} B L^+$, L^+ must first be computed, but the computational complexity of directly computing L^+ is excessive. However, using the method in the literature[19,20] to compute $Y_{k_r \times n}$, the complexity is decreased. Let $\theta = \sqrt{g_v} Q(\hat{W}^{1/2} B)$; then, $Y = \theta L^+$, which is equal to $YL = \theta$. First, $\theta = \sqrt{g_v} Q(\hat{W}^{1/2} B)$is computed, and then, $YL = \theta$. Each row of Y, y_i, is computed by solving the system $y_i L = \theta_i$, where θ_i is the i-th row of θ. The linear time solver of Spielman and Teng[19,20] requires only $\tilde{O}(s)$ time to solve the system. Because $\|y_i - \hat{y}_i\|_L \le \varepsilon \|y_i\|_L$ [17], where $\hat{y}_i$ is the solution, $y_i L = \theta_i$ using the linear time solver. Then, [17]

$$(1-\varepsilon)^2 c_{ij} \le \|\hat{Y}(e_i - e_j)\|^2 \le (1+\varepsilon)^2 c_{ij}$$

Therefore, $c_{ij} \approx \|\hat{Y}(e_i - e_j)\|^2$ with an error bound of ε^2. The component of the algorithm for the approximate commute time embedding of the bipartite graph is illustrated as follows.

Algorithm1 ApCte (Approximate Commute Time Embedding of the Bipartite Graph)

1. input the relation matrix $W_{n_0 \times n_1}$;
2. compute the matrices B, $\hat{W}$ and L using $W_{n_0 \times n_1}$;
3. compute $\theta = \sqrt{g_v} Q(\hat{W}^{1/2} B)$;
4. compute each $\hat{y}_i$ using the system $y_i L = \theta_i$ by calling to the Spielman-Teng solver k_r times [14], $1 \le i \le k_r$;
5. output the approximate commute time embedding $\hat{Y}$.

All data objects of X_0 and X_1 are mapped into a common subspace $\hat{Y}$, where the first n_0 column vectors of $\hat{Y}$ indicate X_0 and the last n_1 column vectors of $\hat{Y}$ indicate X_1. The dataset is composed of the $n = n_0 + n_1$ column vectors of $\hat{Y}$ is called an indicator dataset. The input matrix $W_{n_0 \times n_1}$ is a sparse matrix with s nonzero elements. Therefore, the complexity of computing the

matrices B, $\hat{W}$ and L in step 2 is $O(2s) + O(s) + O(n)$. The sparse matrix B has $2s$ nonzero elements, and the diagonal matrix $\hat{W}$ has s nonzero elements. Computing $\theta = \sqrt{g_v}Q(\hat{W}^{1/2}B)$ takes $O(2sk_r + s)$ time in step 3. Because the linear time solver of Spielman and Teng[19,20] requires only $\tilde{O}(s)$ time to solve for each y_i of system $y_iL = \theta_i$, constructing $\hat{Y}$ takes $\tilde{O}(sk_r)$ time in step 4. Therefore, the complexity of algorithm1, ApCte, is only $O(2s) + O(s) + O(n) + O(2sk_r + s) + \tilde{O}(sk_r) = \tilde{O}(4s + n + 3sk_r)$. In practice, $k_r = O(\log n / \varepsilon^2)$ is small and does not vary between different datasets. The indicator dataset includes low-dimensional homogeneous data; therefore, traditional algorithms can be used for the indicator dataset.

A General Model Formulation

Given a dataset $\chi = \{X_t\}_{t=0}^{T}$ with T+1 types, where X_t is a dataset belonging to the t-th type, a weighted graph $G = < V, E, W >$ on χ is called an information network; if $V(G) = \chi$, the $E(G)$ is a binary relation on V and W: $E \rightarrow R^+$. Such an information network is called a heterogeneous information network when $T \geq 1$ and a homogeneous information network when $T = 0$[6].

An information network $G = < V, E, W >$ on χ is called a heterogeneous information network with a star network schema if $\forall e = \langle x_i, x_j \rangle \in E$, $x_i \in X_0$ and $x_j \in X_t$ $(t \neq 0)$. X_0 is the target dataset, and X_t $(t \neq 0)$ is the attribute dataset.

To derive a general model for clustering the target dataset, a heterogeneous information network with a star network schema using the dataset $\chi = \{X_t\}_{t=0}^{T}$ with T+1 types is given, where X_0 is the target dataset and $\{X_t\}_{t=1}^{T}$ are the attribute datasets. $X_t = \{x_1^{(t)}, x_2^{(t)}, \cdots, x_{n_t}^{(t)}\}$, where n_t is the object number of X_t. $W^{(0t)} \in R^{n_0 \times n_t}$ denotes the relation matrix between the target dataset X_0 and the attribute dataset X_t, where the element $w_{ij}^{(0t)}$ denotes the relation between $x_i^{(0)}$ of X_0 and $x_j^{(t)}$ of X_t. If an edge between $x_i^{(0)}$ and $x_j^{(t)}$ exists, its edge weight is $w_{ij}^{(0t)}$. If no edge exists, $w_{ij}^{(0t)} = 0$. T relation matrices $\{W^{(0t)}\}_{t=1}^{T}$ exist in the heterogeneous information network with a star network schema.

The target dataset X_0 and the attribute dataset X_t constitute a bipartite graph, $G^{(0t)}$, which corresponds to the relation matrix $W^{(0t)}$. The indicator dataset $Y^{(0t)} = \{y_1^{(0t)}, y_2^{(0t)}, \cdots, y_{n_0+n_t}^{(0t)}\}$ which also is the approximate commute time embedding of $G^{(0t)}$ can be quickly computed by ApCte, where the first n_0 data of $Y^{(0t)}$ indicate X_0 and the last n_t data of $Y^{(0t)}$ indicate the attribute dataset X_t. $Y_t^{(0)}$ consists of the first n_0 data of $Y^{(0t)}$, and $Y^{(t)}$ consists of the last n_t data of $Y^{(0t)}$. $Y_t^{(0)}$ and $Y^{(t)}$ are called the indicator subsets. $y_i^{(t)} \in Y_t^{(0)}$ indicates the i-th object of X_0 and is called an indicator for $1 \leq i \leq n_0$. There exists a one-to-one correspondence between the indicators of $Y_t^{(0)}$ and the objects of X_0. Because T bipartite graphs correspond to T indicator datasets, the target dataset X_0 is simultaneously indicated by the T indicator subsets $\{Y_t^{(0)}\}_{t=1}^{T}$, and each object of X_0 is simultaneously indicated by T indicators.

$\beta^{(t)}$ is the weight of the relation matrix $W^{(0t)}$, where $\sum_{t=1}^{T} \beta^{(t)} = 1, \beta^{(t)} > 0$. The target dataset X_0 is partitioned into K clusters. The indicators of $\{Y_t^{(0)}\}_{t=1}^{T}$, which indicate the identical object of X_0, belong to T clusters. The T clusters are in T different indicator subsets and are denoted using the same label. Let

$$F = \sum_{t=1}^{T}(\beta^{(t)}\gamma\|Y_t^{(0)} - \omega^{(t)}\|^2) = \sum_{t=1}^{T}(\beta^{(t)}\sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}\|y_i^{(t)} - \omega_j^{(t)}\|^2)) \quad (5)$$

where $\omega_j^{(t)}$ is the j-th cluster center of the indicator subset $Y_t^{(0)}$. There exists a one-to-one correspondence between the indicator function $\gamma = \{\gamma_{ij}\}_{i=1}^{n_0}$ and the objects of X_0. If all indicators, $\{y_i^{(t)}\}_{t=1}^{T}$, that indicates the i-th object of X_0 belong to the j-th cluster, $\gamma_{ij} = 1$; otherwise, $\gamma_{ij} = 0$.

If the objective function F in Eq (5) is minimized, the clusters of X_0 are optimal from the compatible point of view because each indicator subset reflects the relation between the target dataset and the attribute dataset. Obviously, determining the global minimum of Eq (5) is NP hard.

Derivation of Fast Algorithm for Clustering Heterogeneous Information Networks

The following steps allow for the local minimum of F in Eq (5) to be quickly achieved by simultaneously clustering all of the indicator subsets.

Setting the Cluster Label

When given the cluster label of each indicator subset, the modeling process can be simplified. Suppose that the labels of the K clusters of each $Y_t^{(0)}$ are set. Let $q_1, q_2 \in X_0$, $y_1^{(1)}, y_2^{(1)} \in Y_1^{(0)}$, $\cdots y_1^{(T)}, y_2^{(T)} \in Y_T^{(0)}$. $\{y_1^{(t)}\}_{t=1}^{T}$ indicate q_1, and $\{y_2^{(t)}\}_{t=1}^{T}$ indicate q_2. The clusters which indicators for an identical target object belong to have the same label. If one indicator of $\{y_1^{(t)}\}_{t=1}^{T}$ belongs to the j-th cluster, all of the other indicators of $\{y_1^{(t)}\}_{t=1}^{T}$ also belong to the j-th cluster in their respective indicator subset. If $\{y_1^{(t)}\}_{t=1}^{T}$ belongs to the j-th cluster, then all $\{y_2^{(t)}\}_{t=1}^{T}$ either belong to the j-th cluster in their respective indicator subset or none belong to the j-th cluster.

Each cluster of $Y_t^{(0)}$ has an initial center. K random objects are selected from the target dataset X_0. The indicators indicating the K objects are taken as the initial cluster centers for each $Y_t^{(0)}$ and for the clusters whose center indicates an identical target object with the same label. Then, all of the other indicators for an identical target object only belong to the j-th cluster in each $Y_t^{(0)}$ or no indicators belong to the j-th cluster, where $1 \leq j \leq K$. Therefore, the K clusters of $\{Y_t^{(0)}\}_{t=1}^{T}$ are set labels.

The sum of the Weighted Distances

An object of X_0 is indicated by T indicators. All of the T distances between the indicator and the center in each $Y_t^{(0)}$ affect the object allocation. The target object allocation is determined by the sum of the weighted distances for the T indicators. Setting $q_i \in X_0$, $y_i^{(1)} \in Y_1^{(0)}$, $\cdots y_i^{(T)} \in Y_T^{(0)}$, $\{y_i^{(t)}\}_{t=1}^{T}$ indicates q_i. The weighted distance between $y_i^{(t)}$ and the j-th cluster center in $Y_t^{(0)}$ is $\beta^{(t)}\|y_i^{(t)} - \omega_j^{(t)}\|^2$. The sum of the weighted distances is $dis = \sum_{t=1}^{T}(\beta^{(t)}\|y_i^{(t)} - \omega_j^{(t)}\|^2)$, which determines the cluster that the object q_i belongs.

$$j = \arg\min \sum_{t=1}^{T}(\beta^{(t)}\|y_i^{(t)} - \omega_j^{(t)}\|^2) \tag{6}$$

where j is the cluster label.

The Local Minimum of F

F in Eq (5) can also be expressed as

$$F = \sum_{t=1}^{T}(\beta^{(t)}\sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}\|y_i^{(t)} - \omega_j^{(t)}\|^2)) = \sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}\sum_{t=1}^{T}(\beta^{(t)}\|y_i^{(t)} - \omega_j^{(t)}\|^2)) \quad (7)$$

Obviously, Eq (7) is another representation of Eq (5).

Given the initial centers $\{\omega_j^{(t)}\}_{j=1}^{K}$ and the cluster labels in the T indicator subsets $\{Y_t^{(0)}\}_{t=1}^{T}$, $\{Y_t^{(0)}\}_{t=1}^{T}$ is first partitioned by computing Eq (6) and setting $F = F_0$ in Eq (7). The cluster centers of $\{Y_t^{(0)}\}_{t=2}^{T}$ remain the same, and γ_{ij} is unchanged. The new center $\{\hat{\omega}_j^{(1)}\}_{j=1}^{K}$ of each cluster in $Y_1^{(0)}$ is computed. The new center is the mean of all data of each cluster. The new centers $\{\hat{\omega}_j^{(1)}\}_{j=1}^{K}$ of $Y_1^{(0)}$ replace the old centers, and subsequently, Eq (7) is used to set $F = F_1$. Then,

$$F_1 = \sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}(\beta^{(1)}\|y_i^{(1)} - \hat{\omega}_j^{(1)}\|^2 + \sum_{t=2}^{T}(\beta^{(t)}\|y_i^{(t)} - \omega_j^{(t)}\|^2))) \leq F_0 \quad (8)$$

proving

$$F_1 = \sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}(\beta^{(1)}\|y_i^{(1)} - \hat{\omega}_j^{(1)}\|^2 + \sum_{t=2}^{T}(\beta^{(t)}\|y_i^{(t)} - \omega_j^{(t)}\|^2)))$$

Because only the new centers $\{\hat{\omega}_j^{(1)}\}_{j=1}^{K}$ of $Y_1^{(0)}$ replace the old centers, γ_{ij} remains unchanged. Therefore

$$F_1 = \sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}(\beta^{(1)}\|y_i^{(1)} - \hat{\omega}_j^{(1)}\|^2)) + \sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}\sum_{t=2}^{T}(\beta^{(t)}\|y_i^{(t)} - \omega_j^{(t)}\|^2))$$

Because the cluster centers of $\{Y_t^{(0)}\}_{t=2}^{T}$ also remain unchanged, $\sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}\sum_{t=2}^{T}(\beta^{(t)}\|y_i^{(t)} - \omega_j^{(t)}\|^2))$ is constant, and $\sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}(\beta^{(1)}\|y_i^{(1)} - \hat{\omega}_j^{(1)}\|^2)) \leq \sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}(\beta^{(1)}\|y_i^{(1)} - \omega_j^{(1)}\|^2))$. Subsequently,

$$F_1 \leq \sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}(\beta^{(1)}\|y_i^{(1)} - \omega_j^{(1)}\|^2)) + \sum_{i=1}^{n_0}(\gamma_{ij}\sum_{j=1}^{K}\sum_{t=2}^{T}(\beta^{(t)}\|y_i^{(t)} - \omega_j^{(t)}\|^2)) = F_0$$

Thus, the cluster centers of $Y_1^{(0)}$, for $F_1 \leq F_0$, are replaced.

The new centers $\{\hat{\omega}_j^{(1)}\}_{j=1}^{K}$ of $Y_1^{(0)}$ replace the old centers, while the centers of $\{Y_t^{(0)}\}_{t=2}^{T}$ remain unchanged. Re-clustering $\{Y_t^{(0)}\}_{t=1}^{T}$ using Eq (6), where the corresponding value is $F = F_2$ in Eq (7), gives $F_2 \leq F_1$.

Partitioning $\{Y_t^{(0)}\}_{t=1}^{T}$ using Eq (6) computes the new cluster centers $\{\hat{\omega}_j^{(1)}\}_{j=1}^{K}$ of $Y_1^{(0)}$; the new centers replace the old centers $\{\omega_j^{(1)}\}_{j=1}^{K}$. Then, the same procedure is repeated for each $\{Y_t^{(0)}\}_{t=2}^{T}$. The value of F decreases in this case. The above procedures are repeated until F in Eq (7) converges; then, the local minimum of F in Eq (7) is obtained. The algorithm based on the approximate commute time embedding for heterogeneous information networks is shown below.

Algorithm 2 FctClus (Fast Clustering Algorithm based on the Approximate Commute Time Embedding for Heterogeneous Information Networks)

1. Input relation matrices $\{W^{(0t)} \in R^{n_0 \times n_t}\}_{t=1}^{T}$, weights $\{\beta^{(t)} > 0\}_{t=1}^{T}$ and cluster number K;
2. **for** $t = 1$ to T **do**
3. Compute indicator dataset $Y^{(0t)}$ of the bipartite graph corresponding to $W^{(0t)}$ using algorithm 1;
4. Constitute the indicator subset $Y_t^{(0)}$ that indicates X_0;
5. end for
6. Initialize the K initial cluster centers $\{\omega_j^{(t)}\}_{j=1}^{K}$ of $\{Y_t^{(0)}\}_{t=1}^{T}$ and set the cluster label;
7. loop
8. **for** $t = 1$ to T **do**
9. Partition $\{Y_t^{(0)}\}_{t=1}^{T}$ into K clusters by computing Eq (6);
10. Re-compute the new cluster centers $\{\hat{\omega}_j^{(t)}\}_{j=1}^{K}$ of $Y_t^{(0)}$;
11. $\{\omega_j^{(t)} = \hat{\omega}_j^{(t)}\}_{j=1}^{K}$;
12. end for
13. end loop
14. Output the clusters of X_0.

The computational complexity of steps 2~5 is $\tilde{O}(\sum_{t=1}^{T}(4s_t + n_t + 3s_t k_r))$ in algorithm 2, where T is the number of relational matrices in the heterogeneous information network and k_r is the data dimension of $Y_t^{(0)}$. n_t and s_t are the node number and edge number of the t-th bipartite graph, respectively. Step 6 requires only $O(K)$ time; the time is constant. The object number of X_0 is equal to the indicator number of each indicator subset, thus the computational complexity of steps 7~13 is $O(uTKk_r n_0)$, where K is the number of clusters of each $Y_t^{(0)}$; n_0 is the data number of each $Y_t^{(0)}$; and u is the iteration number for F in Eq (7) convergence. Therefore, the computational complexity of algorithm 2, FctClus, is $\tilde{O}(\sum_{t=1}^{T}(4s_t + n_t + 3s_t k_r)) + O(uTKk_r n_0)$, where k_r and u are small and T and K are constant.

Experiments

The Experimental Dataset

The experimental datasets are composed of real data selected from the DBLP data. The DBLP is a typical heterogeneous information network in computer science domain and contains 4 types of objects, including papers, authors, terms and venues. Two different-scaled heterogeous datasets called S_{small} and S_{large} respectively are used in experiments.

S_{small} is the small test dataset and is called the "four-area dataset", as in the literature[6]. S_{small} extracted from the DBLP dataset downloaded in 2011 contains four areas related to data mining: databases, data mining, information retrieval and machine learning. Five

representative conferences for each area are chosen, and all papers and terms that appear in the titles are included. S_{small} is showed in S1 File.

S_{large} is the large test dataset and extracted from the Chinese DBLP dataset, which are sharing resources released by Institute of automation, Chinese Academy of Sciences. S_{large} includes 34 computer science journals, 16, 567 papers, 47, 701 authors and 52,262 terms(keywords). S_{large} is showed in S2 File.

When analyzing the papers, this object is the target dataset, and the other objects are the attribute datasets. There is no direct link between papers because the DBLP provided very limited citation information. When analyzing the authors, this object is the target dataset, while papers and venues are the attribute datasets. However, there is a direct link between authors because of the co-author relation between various authors; therefore, authors are another attribute dataset related to the target dataset.

The experiments are performed in the MATLAB 7.0 programming environment. The matlab source codes for our algorithm are showed in S3 File and are available online at https://github.com/lsy917/chenlimin, which include a main program and three function programs. FctClus.m is the main program which output the clusters of the object dataset, and ApCte.m, Prematrix.m and Net_Branches.m are function programs. The Koutis CMG solver[14] is used in all experiments as the nearly linear time solver to create the embedding. The solver uses symmetric, diagonally dominant matrices that are available online at http://www.cs.cmu.edu/~jkoutis/cmg.html.

The Relational Matrix

Papers are the target dataset, while authors, venues and terms are the attribute datasets. X_0 denotes papers, and X_1, X_2 and X_3 denote authors, venues and terms, respectively. $W^{(0t)}$ is the relation matrix between X_0 and X_t, $1 \le t \le 3$. The element of $\{W^{(0t)}\}_{t=1}^{3}$ is

$$w_{ij}^{(0t)} = \begin{cases} 1 & \text{if } i \in X_0, j \in X_1 \cup X_2, \text{node } i \text{ links to } j; \\ p & \text{if } i \in X_0, j \in X_3, \text{ node } i \text{ appears } p \text{ times in node } j; \\ 0 & \text{otherwise;} \end{cases}$$

When authors are the target dataset, papers and venues are the attribute datasets. Authors are also an attribute dataset because of the co-author relation existing between authors. X_0 denotes authors when X_1 and X_2 denote papers and venues, respectively. $W^{(0t)}$ is the relation matrix between X_0 and X_t, $0 \le t \le 2$. The element of $\{W^{(0t)}\}_{t=0}^{2}$ is

$$w_{ij}^{(0t)} = \begin{cases} 1 & \text{if } i \in X_0, j \in X_1 \cup X_2, \text{node } i \text{ links to } j; \\ p & \text{if } i \in X_0, j \in X_0, \text{ node } i \text{ and } j \text{ co} - \text{author } p \text{ papers;} \\ 0 & \text{otherwise;} \end{cases}$$

All the algorithms use the same relation matrix for all experiments.

Parameter Analysis

Analysis of Parameter k_r. The equation [13]

$$Accuracy = \frac{\sum_{i=1}^{n} \delta[map(c_i) = label(i)]}{n}$$

is used to compute the clustering accuracy in the experiments, where n is the object number of dataset, *label(i)* is the cluster label, and c_i is the predicted label of an object i. $\delta(\cdot)$ is an indicator function:

$$\delta(\cdot) = \begin{cases} 1 & map(i) = label(i) \\ 0 & map(i) \neq label(i) \end{cases}.$$

k_r is small in practice, and minimal differences exist among the various datasets[13]. The literature[13] has proved that the accuracy curve is flat for clustering different homogeneous datasets when $k_r \geq 50$.

Using the small dataset S_{small}, the clustering accuracy as a function of k_r in a heterogeneous information network is studied.

An experiment with different k_r is conducted in the small dataset, S_{small}. In the FctClus algorithm, the weight of $\{W^{(0t)}\}_{t=1}^{3}$ is taken as $\beta^{(1)} = 0.3$, $\beta^{(2)} = 0.4$ and $\beta^{(3)} = 0.3$ for clustering papers; the weight of $\{W^{(0t)}\}_{t=0}^{2}$ is taken as $\beta^{(1)} = 0.4$, $\beta^{(2)} = 0.2$ and $\beta^{(3)} = 0.4$ for clustering authors. The clustering accuracy is affected by k_r, as shown in Fig 1 and Fig 2.

The parameter k_r is quite small because the accuracy curve is flat when k_r obtains a certain value. $k_r = 60$ is suitable for the dataset in the experiment. k_r is small and does not considerably affect the computation speed of FctClus. It is advantageous that FctClus is not sensitive to k_r in terms of both accuracy and performance. All weights of the relation matrix and $k_r = 60$ are studied in other experiments.

Analysis of Iteration *u*

An experiment is conducted in the small dataset S_{small} to compare the influence of iteration u on the clustering result, where $k_r = 60$. The influence of the iteration u on clustering papers and authors is shown in Fig 3 and Fig 4. The algorithm quickly convergences when $u = 30$. $u = 40$ is examined in the other experiments.

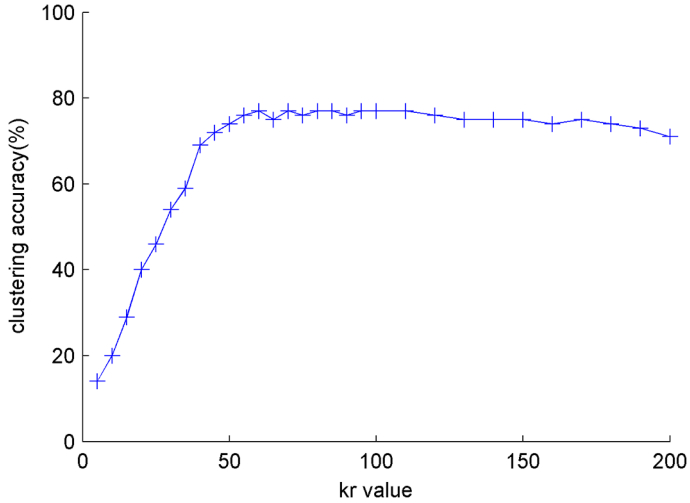

Fig 1. The influence of k_r for clustering papers on s_{small}.

doi:10.1371/journal.pone.0130086.g001

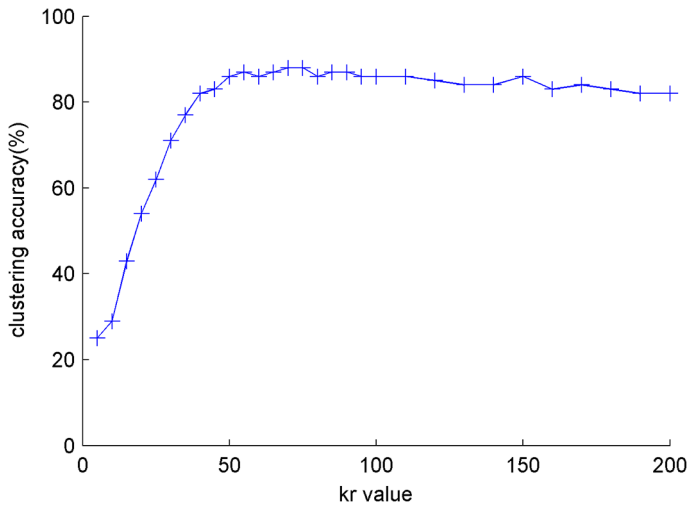

Fig 2. The influence of k_r for clustering authors on s_{small}.

doi:10.1371/journal.pone.0130086.g002

Comparison of Clustering Accuracy and Computation Speed

The complexity of the algorithms is too high for large-scale networks based on semi-definite programming[2,3] and spectral clustering algorithms for multi-type relational data[5]. The low-complexity algorithms CIT[4], NetClus[6] and ComClus[10] are selected for comparison with the FctClus algorithm in terms of clustering accuracy and computation speed; the datasets S_{small} and S_{large} are also chosen for this experiment.

The initial cluster centers of FctClus or the initial cluster partitions of the other three algorithms are randomly selected 3 times. The best clustering accuracy of the 3 measurements is used as the clustering accuracy of the four algorithms, and the computation speed at this time is considered as the measured computation speed. The parameters in literature[6] are used as the parameters in NetClus, and the parameters in literature[10] are used as the parameters in ComClus in this experiment. The comparison results are shown in Table 1 and Table 2.

The clustering accuracy of FctClus is the highest of all four algorithms. The clustering accuracy of CIT is lower than that of FctClus because the bipartite graphs of the heterogeneous

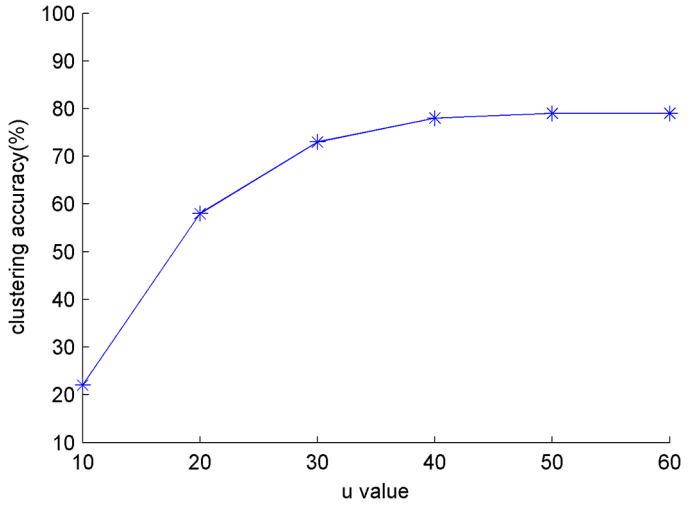

Fig 3. The influence of *u* for clustering papers on s_{small}.

doi:10.1371/journal.pone.0130086.g003

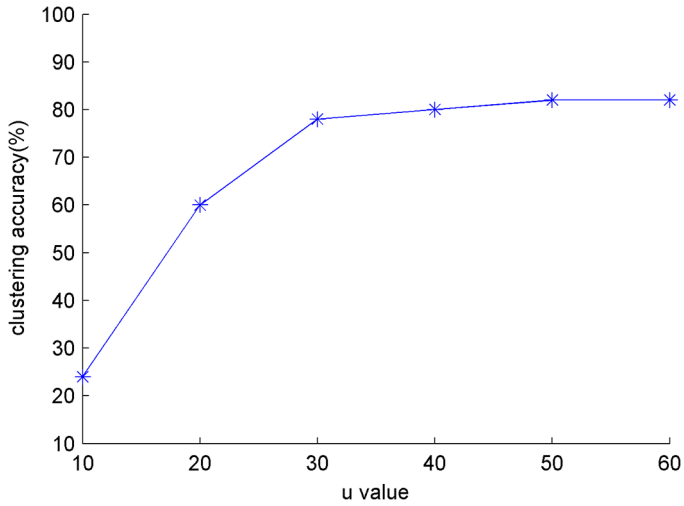

Fig 4. The influence of *u* for clustering authors on s_{small}.

doi:10.1371/journal.pone.0130086.g004

information networks are sparse. The computational complexity of CIT is $O(n^2)$, and the convergence speed of CIT is low when the heterogeneous information network is sparse. The clustering accuracy of NetClus is low because only heterogeneous relations are used. Homogeneous and heterogeneous relations are both used in ComClus; therefore, the accuracy of ComClus is higher than that of NetClus. FctClus is an algorithm based on commute time embedding. The data relations are explored using commute time and the direct relations of the target dataset are considered. FctClus is not affected by the sparsity of networks; thus, FctClus is highly accurate.

The computation speed of FctClus is nearly as fast as NetClus. The experiment demonstrates that FctClus is effective. FctClus is more universal and can be adapted for clustering any heterogeneous information network with a star network schema. However, NetClus and ComClus can only be adapted for clustering bibliographic networks because NetClus and ComClus depend on a ranking function of a specific application field.

Comparison of Clustering Stability

To compare the stability of the FctClus, NetClus and CIT algorithms, the small dataset S_{small} is used for clustering papers in this experiment. ComClus is a derivation algorithm of NetClus; it has the same properties as NetClus. ComClus is not considered in this study.

The initial cluster centers of FctClus and the initial cluster partitions of NetClus and CIT are randomly recorded 10 times, and the three algorithms are executed 10 times respectively. The clustering accuracy of the three algorithms for 10 times is shown in Fig 5. Although the computation speeds of FctClus and NetClus are both high, Fig 5 shows that the stability of

Table 1. Comparison of clustering accuracy (%).

target object &dataset	CIT	NetClus	ComClus	FctClus
Papers on s_{small}	73.91	71.54	72.83	78.87
Authors on s_{small}	74.41	69.13	74.91	81.33
Papers on s_{large}	70.84	71.28	72.93	76.36
Authors on s_{large}	71.02	68.29	73.01	77.94

doi:10.1371/journal.pone.0130086.t001

Table 2. Comparison of computation speed(s).

target object &dataset	CIT	NetClus	ComClus	FctClus
Papers on s_{small}	78.5	37.3	40.3	37.1
Authors on s_{small}	79.8	36.9	39.8	38.3
Papers on s_{large}	1469.3	802.6	827.3	808.4
Authors on s_{large}	1484.7	743.7	781.4	774.9

doi:10.1371/journal.pone.0130086.t002

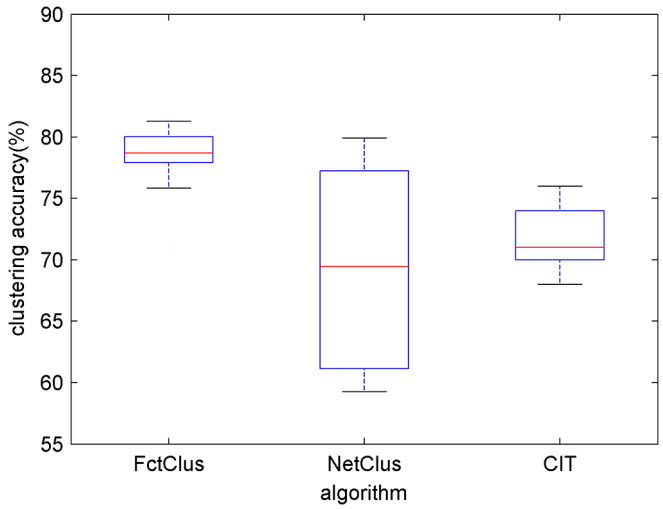

Fig 5. A stability comparison of the 3 algorithms for 10 times.

doi:10.1371/journal.pone.0130086.g005

Table 3. Distribution of running time for FctClus.

target object &dataset	Embedding time(s)	Clustering time(s)	Total time(s)
Papers on s_{small}	19.6	17.5	37.1
Authors on s_{small}	18.1	20.2	38.3
Papers on s_{large}	398.8	409.6	808.4
Authors on s_{large}	382.4	392.5	774.9

doi:10.1371/journal.pone.0130086.t003

FctClus is higher than that of NetClus and that the initial centers do not greatly impact the clustering result of FctClus. However, NetClus is very unstable, and the initial clusters greatly impact the clustering accuracy and convergence speed of NetClus. CIT is more stable than NetClus, but the clustering accuracy is low.

Running Time Analysis of the FctClus Algorithm

The running time distributions of FctClus on the two datasets are shown in Table 3. The experimental data show that FctClus is effective. The running time for serial computing the three embedding is less than 50% of the total running time. When utilizing parallel computing for the three embedding, the computation speed is higher. When clustering indicator subsets in parallel, the computation speed may also be increased.

Conclusions

The relation between the original data described by the commute time guarantees the accuracy and performance of the FctClus algorithm. Because heterogeneous information networks are sparse, FctClus can use random mapping and a linear time solver[14] to compute the approximate commute time embedding, which guarantees the high computation speed. FctClus is effective and may be broadly implemented for large heterogeneous information networks, as demonstrated in theory and experimentally. The weight of the relation matrix impacts the target function, but the weight cannot be determined self-adaptively; this requires further research. The relations of data in the real world are typically high-order heterogeneous, so effective clustering algorithms for heterogeneous information networks with any schema will be studied in the future.

Supporting Information

S1 File. S_{small} dataset.
(TXT)

S2 File. S_{large} dataset.
(TXT)

S3 File. The matlab source codes for algorithm.
(TXT)

Author Contributions

Conceived and designed the experiments: JY LMC. Performed the experiments: LMC. Analyzed the data: JPZ. Contributed reagents/materials/analysis tools: JY LMC. Wrote the paper: LMC.

References

1. Sun Y, Han J (2012) Mining heterogeneous information networks: principles and methodologies. Synthesis Lectures on Data Mining and Knowledge Discovery, 3 (2). pp.1–159.
2. Gao B, Liu TY, Qin T, Zheng X, Cheng QS, Ma WY (2005) Web image clustering by consistent utilization of visual features and surrounding texts. In Proceedings of the 13th annual ACM international conference on Multimedia.pp.112-121.
3. Gao B, Liu TY, Zheng X, Cheng QS, Ma WY (2005) Consistent bipartite graph co-partitioning for star-structrured high-order heterogeneous data co-clustering. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. pp.41-50.
4. Gao B, Liu TY, Ma WY (2006) Star-structured high-order heterogeous data co-clustering based on cosistent information theory. In Data Mining, 2006. ICDM'06. Sixth International Conference on. pp.880-884.
5. Long B, Zhang ZM, Wu X, Yu PS (2006) Spectral clustering for multi-type relational data. In Proceedings of the 23rd international conference on Machine learning. pp.585-592.
6. Sun Y, Yu Y, Han J (2009) Rankclus: 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. pp.797-806.
7. Sun Y, Norick B, Han J, Yan X, Yu PS, Yu, X (2012) Integrating meta-path selection with user-guided object clustering in heterogeneous information networks, In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp.1348-1356.
8. Li P, Wen J, Li X (2013) SNTClus: A novel service clustering algorithm based on network analysis and service tags. Przegląd Elektrotechniczny.pp.89.
9. Li P, Chen L, Li X, Wen J (2013) RNRank: Network-based ranking on relational tuples, In Behavior and Social Computing. Springer International Publishing. pp. 139–150.

10. Wang R, Shi C, Yu PS, Wu B (2013) Integrating clustering and ranking on hybrid heterogeneous information network. In Advances in Knowledge Discovery and Data Mining.pp.583-594.
11. Aggarwal CC, Xie Y, Philip SY (2012) Dynamic link inference in heterogeneous networks. In SDM. pp.415-426.
12. Zhang L, Chen C, Bu J, Chen Z, Cai D, Han J (2012)Locally discriminative coclustering. Knowledge and Data Engineering, IEEE Transactions on, 24 (6).pp.1025–1035.
13. Khoa NLD, Chawla S (2011) Large scale spectral clustering using approximate commute time embedding. arXiv preprint arXiv:1111.4541.
14. Koutis I, Miller GL, Tolliver D (2011) Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. Computer Vision and Image Understanding, 115 (12). pp.1638–1646.
15. Fouss F, Pirotte A,Renders JM, Saerens M (2007)Random walk computation of similarities between nodes of a graph with application to collaborative recommendation. Knowledge and Data Engineering, IEEE Transactions on, 19 (3).pp.355–369.
16. Qiu H, Hancock ER (2007)Clustering and embedding using commute times. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29 (11).pp.1873–1890. PMID: 17848771
17. Spielman DA, Srivastava N (2008) Graph sparsification by effective resistances. In Proceedings of the 40th annual ACM symposium on Theory of computing, STOC '08.pp.563-568.
18. Achlioptas D (2001) Database-friendly random projections, in Proceedings of the twentieth ACM SIGMOD SIGACT SIGART symposium on Principles of database systems, PODS '01.pp.274-281.
19. Spielman DA, Teng SH (2004) Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, STOC '04.pp.81-90.
20. Spielman DA, Teng SH (2014) Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. SIAM Journal on Matrix Analysis and Applications, 35 (3).pp.835–885.