Flexible and Robust Co-Regularized Multi-Domain Graph Clustering

Wei Cheng [1], Xiang Zhang [2], Zhishan Guo [1], Yubao Wu [2],
Patrick F. Sullivan [3], and Wei Wang [4]
[1]Department of Computer Science, University of North Carolina at Chapel Hill, NC 27599, USA,
[2]Department of Electrical Engineering and Computer Science, Case Western Reserve University, OH 44106, USA, [3]Departments of Genetics and Psychiatry, University of North Carolina at Chapel Hill, NC 27599, USA, [4]Department of Computer Science, University of California, Los Angeles, CA 90095, USA
[1]{weicheng,zsguo}@cs.unc.edu,[2]{xiang.zhang,yubao.wu}@case.edu,
[3]pfsulliv@med.unc.edu,[4]weiwang@cs.ucla.edu

ABSTRACT

Multi-view graph clustering aims to enhance clustering performance by integrating heterogeneous information collected in different domains. Each domain provides a different view of the data instances. Leveraging cross-domain information has been demonstrated an effective way to achieve better clustering results. Despite the previous success, existing multi-view graph clustering methods usually assume that different views are available for the *same* set of instances. Thus instances in different domains can be treated as having strict *one-to-one* relationship. In many real-life applications, however, data instances in one domain may correspond to multiple instances in another domain. Moreover, relationships between instances in different domains may be associated with weights based on prior (partial) knowledge. In this paper, we propose a flexible and robust framework, CGC (Co-regularized Graph Clustering), based on non-negative matrix factorization (NMF), to tackle these challenges. CGC has several advantages over the existing methods. First, it supports *many-to-many* cross-domain instance relationship. Second, it incorporates weight on cross-domain relationship. Third, it allows partial cross-domain mapping so that graphs in different domains may have different sizes. Finally, it provides users with the extent to which the cross-domain instance relationship violates the in-domain clustering structure, and thus enables users to re-evaluate the consistency of the relationship. Extensive experimental results on UCI benchmark data sets, newsgroup data sets and biological interaction networks demonstrate the effectiveness of our approach.

Categories and Subject Descriptors

H.8 [**Database management**]: Database applications—*Data mining*

General Terms

Algorithms, Experimentation, Theory

KDD '13 Chicago, Illinois USA

Keywords

graph clustering, nonnegative matrix factorization, co-regularization

1. INTRODUCTION

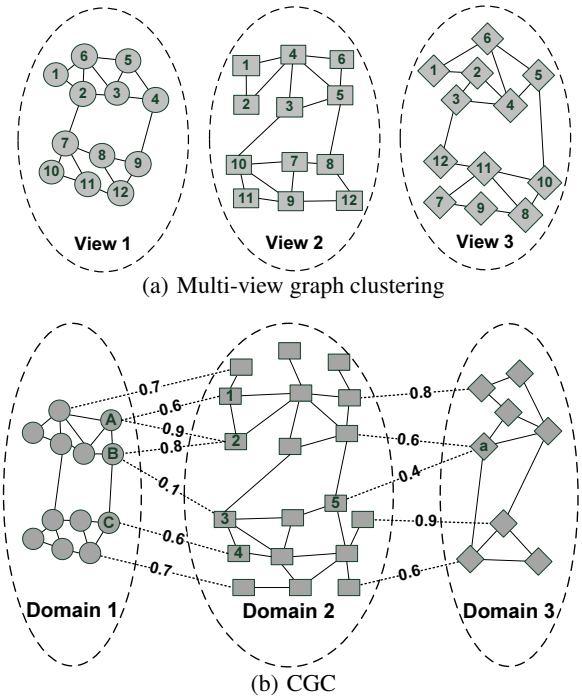
Graphs are ubiquitous in real-life applications. A large volume of graph data have been generated, such as social networks [21], biology interaction networks [11], and literature citation networks [28]. Graph clustering has attracted increasing research interest recently. Several effective approaches have been proposed in the literature, such as spectral clustering [24], symmetric Non-negative Matrix Factorization (symNMF) [17], Markov clustering (MCL) [32], and Ncut [25].

In many applications, graph data may be collected from heterogeneous domains (sources) [13]. For example, the gene expression levels may be reported by different techniques or on different sample sets, thus the gene co-expression networks built on them are heterogeneous; the proximity networks between researchers such as co-citation network and co-author network are also heterogeneous. By exploiting multi-domain information to refine clustering and resolve ambiguity, multi-view graph clustering methods have the potential to dramatically increase the accuracy of the final results [3, 19, 5]. The key assumption of these methods is that the same set of data instances may have multiple representations, and different views are generated from the same underlying distribution [5]. These views should agree on a consensus partition of the instances that reflects the hidden ground truth [22]. The learning objective is thus to find the most consensus clustering structure across different domains.

Existing multi-view graph clustering methods usually assume that information collected in different domains are for the same set of instances. Thus, the cross-domain instance relationships are strictly *one-to-one*. This also implies that different views are of the same size. For example, Fig. 1(a) shows a typical scenario of multi-view graph clustering, where the same set of 12 data instances has 3 different views. Each view gives a different graph representation of the instances.

In many real-life applications, it is common to have cross-domain relationship as shown in Fig. 1(b). This example illustrates several key properties that are different from the traditional multi-view graph clustering scenario.

- An instance in one domain may be mapped to multiple instances in another domain. For example, in Fig. 1(b), instance Ⓐ in domain 1 is mapped to two instances ① and ②



(a) Multi-view graph clustering

(b) CGC

Figure 1: Multi-view graph clustering vs co-regularized multi-domain graph clustering (CGC)

in domain 2. The cross-domain relationship is many-to-many rather than one-to-one.

- Mapping between cross-domain instances may be associated with weights, which is a generalization of a binary relationship. As shown in Fig. 1(b), each cross-domain mapping is coupled with a weight. Users may specify these weights based on their prior knowledge.
- The cross-domain instance relationship may be a partial mapping. Graphs in different domains may have different sizes. Some instance in one domain may not have corresponding instance in another. As shown in Fig. 1(b), mapping between instances in different domains is not complete.

One important problem in bioinformatics research is protein functional module detection [16]. A widely used approach is to cluster protein-protein interaction (PPI) networks [2]. In a PPI network, each instance (node) is a protein and an edge represents the strength of the interaction between two connected proteins. To improve the accuracy of the clustering results, we may explore the data collected in multiple domains, such as gene co-expression networks [15] and genetic interaction networks [7]. The relationship across gene, protein and genetic variant domains can be many-to-many. For example, multiple proteins may be synthesized from one gene and one gene may contain many genetic variants. Consider another application of text clustering, where we want to cluster journal paper corps (domain 1) and conference paper corps (domain 2). We may construct two affinity (similarity) graphs for domains 1 and 2 respectively, in which each instance (node) is a paper and an edge represents the similarity between two papers (e.g., cosine similarity between term-frequency vectors of the two papers). Some journal papers may be extended versions of one or multiple conference papers. Thus the mappings between papers in two domains may be many-to-many.

These emerging applications call for novel cross-domain graph clustering methods. In this paper, we propose CGC (Co-regularized Graph Clustering), a flexible and robust approach to integrate heterogenous graph data. Our contributions are summarized as follows.

1. We propose and investigate the problem of clustering multiple heterogenous graph data, where the cross-domain instance relationship is *many-to-many*. This problem has a wide range of applications and poses new technical challenges that cannot be directly tackled by traditional multi-view graph clustering methods.

2. We develop a method, CGC, based on collective symmetric non-negative matrix factorization with co-regularized penalty to manipulate cross-domain relationships. CGC allows weighted cross-domain relationships. It also allows partial mapping and can handle graphs of different sizes. Such flexibility is crucial for real-life applications. We also provide rigid theoretical analysis of the performance of the proposed method.

3. We develop effective and efficient techniques to handle the situation when the cross-domain relationship contains noise. Our method supports users to evaluate the accuracy of the specified relationships based on single-domain clustering structure. For example, in Fig. 1(b), mapping between (Ⓑ–③) in domains 1 and 2, and (⑤–ⓐ) in domains 2 and 3, may not be accurate as they are inconsistent with in-domain clustering structure. (Note that each domain contains two clusters, one on the top and one at the bottom.)

4. We evaluate the proposed method on benchmark UCI data sets, newsgroup data sets and various biological interaction networks. The experimental results demonstrate the effectiveness of our method.

2. PROBLEM FORMULATION

Suppose that we have d graphs, each from a domain in $\{\mathcal{D}_1,\mathcal{D}_2,...,\mathcal{D}_d\}$. We use n_π to denote the number of instances (nodes) in the graph from domain $\mathcal{D}_\pi$ ($1 \leq \pi \leq d$). Each graph is represented by an affinity (similarity) matrix. The affinity matrix of the graph in domain $\mathcal{D}_\pi$ is denoted as $\mathbf{A}^{(\pi)} \in \mathbb{R}_+^{n_\pi \times n_\pi}$. In this paper, we follow the convention and assume that $\mathbf{A}^{(\pi)}$ is a symmetric and non-negative matrix [24, 17]. We denote the set of pairwise cross-domain relationships as $\mathcal{I} = \{(i,j)\}$ where i and j are domain indices. For example, $\mathcal{I} = \{(1,3),(2,5)\}$ contains two cross-domain relationships (mappings): the relationship between instances in $\mathcal{D}_1$ and $\mathcal{D}_3$, and the relationship between instances in $\mathcal{D}_2$ and $\mathcal{D}_5$. Each relationship $(i,j) \in \mathcal{I}$ is coupled with a matrix $\mathbf{S}^{(i,j)} \in \mathbb{R}_+^{n_j \times n_i}$, indicating the (weighted) mapping between instances in $\mathcal{D}_i$ and $\mathcal{D}_j$, where n_i and n_j represent the number of instances in $\mathcal{D}_i$ and $\mathcal{D}_j$ respectively. We use $\mathbf{S}_{a,b}^{(i,j)}$ to denote the weight between the a-th instance in $\mathcal{D}_j$ and the b-th instance in $\mathcal{D}_i$, which can be either binary (0 or 1) or quantitative (any value between [0,1]). Important notations are listed in Table 1.

Our goal is to partition each $\mathbf{A}^{(\pi)}$ into k_π clusters while considering the co-regularizing constraints implicitly represented by the cross-domain relationships in $\mathcal{I}$.

3. CO-REGULARIZED MULTI-DOMAIN GRAPH CLUSTERING

Table 1: Summary of symbols and their meanings

Symbols	Description
d	The number of domains
$\mathcal{D}_\pi$	The π-th domain
n_π	The number of instances in the graph from $\mathcal{D}_\pi$
k_π	The number of clusters in $\mathcal{D}_\pi$
$\mathbf{A}^{(\pi)}$	The affinity matrix of graph in $\mathcal{D}_\pi$
$\mathcal{I}$	The set of cross-domain relationships
$\mathbf{S}^{(i,j)}$	The relationship matrix between instances in $\mathcal{D}_i$ and $\mathcal{D}_j$
$\mathbf{W}^{(i,j)}$	The confidence matrix of relationship matrix $\mathbf{S}^{(i,j)}$
$\mathbf{H}^{(\pi)}$	The clustering indicator matrix of $\mathcal{D}_\pi$

In this section, we present the Co-regularized Graph Clustering (CGC) method. We model cross-domain graph clustering as a joint matrix optimization problem. The proposed CGC method simultaneously optimizes the empirical likelihood in multiple domains and take into account the cross-domain relationships.

3.1 Objective Function

3.1.1 Single-Domain Clustering

Graph clustering in a single domain has been extensively studied. We adopt the widely used non-negative matrix factorization (NMF) approach [20]. In particular, we use the symmetric version of NMF [17, 9] to formulate the objective of clustering on $\mathbf{A}^{(\pi)}$ as minimizing the objective function:

$$\mathcal{L}^{(\pi)} = ||\mathbf{A}^{(\pi)} - \mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}||_F^2 \tag{1}$$

where $||\cdot||_F$ denotes the Frobenius norm, $\mathbf{H}^{(\pi)}$ is a non-negative matrix of size $n_\pi \times k_\pi$, and k_π is the number of clusters requested. We have $\mathbf{H}^{(\pi)} = [\mathbf{h}_{1*}^{(\pi)}, \mathbf{h}_{2*}^{(\pi)}, ..., \mathbf{h}_{n_\pi *}^{(\pi)}]^{\mathrm{T}} \in \mathbb{R}^{n_\pi \times k_\pi}$, where each $\mathbf{h}_{a*}^{(\pi)}$ $(1 \le a \le n_\pi)$ represents the cluster assignment (distribution) of the a-th instance in domain $\mathcal{D}_\pi$. For hard clustering, $\mathrm{argmax}_j\, \mathbf{h}_{aj}^{(\pi)}$ is often used as the cluster assignment.

3.1.2 Cross-Domain Co-Regularization

To incorporate the cross-domain relationship, the key idea is to add pairwise co-regularizers to the single-domain clustering objective function. We develop two loss functions to regularize the cross-domain clustering structure. Both loss functions are designed to penalize cluster assignment inconsistency with the given cross-domain relationships. The *residual sum of squares (RSS) loss* requires that graphs in different domains are partitioned into the same number of clusters. The *clustering disagreement loss* has no such restriction.

A). Residual sum of squares (RSS) loss function

We first consider the case where the number of clusters is the same in different domains, i.e. $k_1 = k_2 = ... = k_d = k$. For simplicity, we denote the instances in domain $\mathcal{D}_\pi$ as $\{x_1^{(\pi)}, x_2^{(\pi)}, ..., x_{n_\pi}^{(\pi)}\}$. If an instance $x_a^{(i)}$ in $\mathcal{D}_i$ is mapped to an instance $x_b^{(j)}$ in $\mathcal{D}_j$, then the clustering assignments $\mathbf{h}_{a*}^{(i)}$ and $\mathbf{h}_{b*}^{(j)}$ should be similar. We now generalize the relationship to many-to-many. We use $\mathcal{N}^{(i,j)}(x_b^{(j)})$ to denote the set of indices of instances in $\mathcal{D}_i$ that are mapped to $x_b^{(j)}$ with positive weights, and $|\mathcal{N}^{(i,j)}(x_b^{(j)})|$ represents its cardinality. To penalize the inconsistency of cross-domain cluster partitions, for the l-th cluster in $\mathcal{D}_i$, the loss function (residual) for the b-th instance is

$$\mathcal{J}_{b,l}^{(i,j)} = (\mathbb{E}^{(i,j)}(x_b^{(j)}, l) - \mathbf{h}_{b,l}^{(j)})^2 \tag{2}$$

where

$$\mathbb{E}^{(i,j)}(x_b^{(j)}, l) = \frac{1}{|\mathcal{N}^{(i,j)}(x_b^{(j)})|} \sum_{a \in \mathcal{N}^{(i,j)}(x_b^{(j)})} \mathbf{S}_{b,a}^{(i,j)} \mathbf{h}_{a,l}^{(i)} \tag{3}$$

is the weighted mean of cluster assignment of instances mapped to $x_b^{(j)}$, for the l-th cluster.

We assume every non-zero row of $\mathbf{S}^{(i,j)}$ is normalized. By summing up Eq. (2) over all instances in $\mathcal{D}_j$ and k clusters, we have the following residual of sum of squares loss function

$$\mathcal{J}_{RSS}^{(i,j)} = \sum_{l=1}^{k} \sum_{b=1}^{n_j} \mathcal{J}_{b,l}^{(i,j)} = ||\mathbf{S}^{(i,j)}\mathbf{H}^{(i)} - \mathbf{H}^{(j)}||_F^2 \tag{4}$$

B). Clustering disagreement (CD) loss function

When the number of clusters in different domains varies, we can no longer use the RSS loss to quantify the inconsistency of cross-domain partitions. From the previous discussion, we observe that $\mathbf{S}^{(i,j)}\mathbf{H}^{(i)}$ in fact serves as a weighted projection of instances in domain $\mathcal{D}_i$ to instances in domain $\mathcal{D}_j$. For simplicity, we denote the matrix $\widetilde{\mathbf{H}}^{(i\to j)} = \mathbf{S}^{(i,j)}\mathbf{H}^{(i)}$. Recall that $\mathbf{h}_{a*}^{(j)}$ represents a cluster assignment over k_j clusters for the a-th instance in $\mathcal{D}_j$. Then $\widetilde{\mathbf{H}}_{a*}^{(i\to j)}$ corresponds to $\mathbf{H}_{a*}^{(j)}$ for the a-th instance in domain $\mathcal{D}_j$. The previous RSS loss compares them directly to measure the clustering inconsistency. However, it is inapplicable to the case where different domains have different numbers of clusters. To tackle this problem, we first measure the similarity between $\widetilde{\mathbf{H}}_{a*}^{(i\to j)}$ and $\widetilde{\mathbf{H}}_{b*}^{(i\to j)}$, and the similarity between $\mathbf{H}_{a*}^{(j)}$ and $\mathbf{H}_{b*}^{(j)}$. Then we measure the difference between these two similarity values. Taking Fig. 1(b) as an example. Note that Ⓐ and Ⓑ in domain 1 are mapped to ② in domain 2, and Ⓒ is mapped to ④. Intuitively, if the similarity between clustering assignments for ② and ④ is small, the similarity of clustering assignments between Ⓐ and Ⓒ and the similarity between Ⓑ and Ⓒ should also be small. Note that symmetric NMF can handle both linearity and nonlinearity [17]. Thus in this paper, we choose a linear kernel to measure the in-domain cluster assignment similarity, i.e., $K(\mathbf{h}_{a*}^{(j)}, \mathbf{h}_{b*}^{(j)}) = \mathbf{h}_{a*}^{(j)}(\mathbf{h}_{b*}^{(j)})^{\mathrm{T}}$. The cross-domain clustering disagreement loss function is thus defined as

$$\begin{aligned}\mathcal{J}_{CD}^{(i,j)} &= \sum_{a=1}^{n_j}\sum_{b=1}^{n_j} \left(K(\widetilde{\mathbf{H}}_{a*}^{(i\to j)}, \widetilde{\mathbf{H}}_{b*}^{(i\to j)}) - K(\mathbf{h}_{a*}^{(j)}, \mathbf{h}_{b*}^{(j)})\right)^2 \\ &= ||\mathbf{S}^{(i,j)}\mathbf{H}^{(i)}(\mathbf{S}^{(i,j)}\mathbf{H}^{(i)})^{\mathrm{T}} - \mathbf{H}^{(j)}(\mathbf{H}^{(j)})^{\mathrm{T}}||_F^2\end{aligned} \tag{5}$$

3.1.3 Joint Matrix Optimization

We can integrate the domain-specific objective and the loss function quantifying the inconsistency of cross-domain partitions into a unified objective function

$$\min_{\mathbf{H}^{(\pi)} \ge 0 (1 \le \pi \le d)} \mathcal{O} = \sum_{i=1}^{d} \mathcal{L}^{(i)} + \sum_{(i,j)\in\mathcal{I}} \lambda^{(i,j)} \mathcal{J}^{(i,j)} \tag{6}$$

where $\mathcal{J}^{(i,j)}$ can be either $\mathcal{J}_{RSS}^{(i,j)}$ or $\mathcal{J}_{CD}^{(i,j)}$. $\lambda^{(i,j)} \ge 0$ is a tuning parameter balancing between in-domain clustering objective and cross-domain regularizer. When all $\lambda^{(i,j)}$= 0, Eq. (6) degenerates to d independent graph clusterings. Intuitively, the more reliable the prior cross-domain relationship, the larger the value of $\lambda^{(i,j)}$.

3.2 Learning Algorithm

In this section, we present an alternating scheme to optimize the objective function in Eq. (6), that is, we optimize the objective with respect to one variable while fixing others. This procedure continues until convergence. The objective function is invariant under these updates if and only if $\mathbf{H}^{(\pi)}$'s are at a stationary point [20].

Specifically, the solution to the optimization problem in Eq. (6) is based on the following two theorems, which is derived from the Karush-Kuhn-Tucker (KKT) complementarity condition [4]. Detailed theoretical analysis of the optimization procedure will be presented in the next section.

Theorem 3.1. *For RSS loss, updating* $\mathbf{H}^{(\pi)}$ *according to Eq.* (7) *will monotonically decrease the objective function in Eq.* (6) *until convergence.*

$$\mathbf{H}^{(\pi)} \leftarrow \mathbf{H}^{(\pi)} \circ \left(\frac{\mathbf{\Psi}'(\mathbf{H}^{(\pi)})}{\mathbf{\Xi}'(\mathbf{H}^{(\pi)})} \right)^{\frac{1}{4}} \tag{7}$$

where

$$\begin{aligned} \mathbf{\Psi}'(\mathbf{H}^{(\pi)}) &= \mathbf{A}^{(\pi)}\mathbf{H}^{(\pi)} + \sum_{(i,\pi)\in\mathcal{I}} \frac{\lambda^{(i,\pi)}}{2}\mathbf{S}^{(i,\pi)}\mathbf{H}^{(i)} \\ &+ \sum_{(\pi,j)\in\mathcal{I}} \frac{\lambda^{(\pi,j)}}{2}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{H}^{(j)} \end{aligned} \tag{8}$$

and

$$\begin{aligned} \mathbf{\Xi}'(\mathbf{H}^{(\pi)}) &= \mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} + \sum_{(i,\pi)\in\mathcal{I}} \frac{\lambda^{(i,\pi)}}{2}\mathbf{H}^{(\pi)} \\ &+ \sum_{(\pi,j)\in\mathcal{I}} \frac{\lambda^{(\pi,j)}}{2}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \end{aligned} \tag{9}$$

Theorem 3.2. *For CD loss, updating* $\mathbf{H}^{(\pi)}$ *according to Eq.* (10) *will monotonically decrease the objective function in Eq.* (6) *until convergence.*

$$\mathbf{H}^{(\pi)} \leftarrow \mathbf{H}^{(\pi)} \circ \left(\frac{\mathbf{\Psi}(\mathbf{H}^{(\pi)})}{\mathbf{\Xi}(\mathbf{H}^{(\pi)})} \right)^{\frac{1}{4}} \tag{10}$$

where

$$\begin{aligned} \mathbf{\Psi}(\mathbf{H}^{(\pi)}) &= \mathbf{A}^{(\pi)}\mathbf{H}^{(\pi)} \\ &+ \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}\mathbf{S}^{(i,\pi)}\mathbf{H}^{(i)}(\mathbf{H}^{(i)})^{\mathrm{T}}(\mathbf{S}^{(i,\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\ &+ \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{H}^{(j)}(\mathbf{H}^{(j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \end{aligned} \tag{11}$$

and

$$\begin{aligned} \mathbf{\Xi}(\mathbf{H}^{(\pi)}) &= \mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\ &+ \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}\mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\ &+ \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \end{aligned} \tag{12}$$

where $\circ$, $\frac{[\cdot]}{[\cdot]}$ *and* $(\cdot)^{\frac{1}{4}}$ *are element-wise operators.*

Based on Theorem 3.1 and Theorem 3.2, we develop the iterative multiplicative updating algorithm for optimization and summarize it in Algorithm 1.

3.3 Theoretical Analysis

3.3.1 Derivation

We derive the solution to Eq. (6) following the constrained optimization theory [4]. Since the objective function is not jointly convex, we adopt an effective alternating minimization algorithm to find a locally optimal solution. We prove Theorem 3.2 in the following. The proof of Theorem 3.1 is similar and hence omitted.

Algorithm 1: Co-regularized Graph Clustering (CGC)

Input: graphs from d domains, each of which is represented by an affinity matrix $\mathbf{A}^{(\pi)}$, k_π(number of clusters in domain $\mathcal{D}_\pi$), a set of pairwise relationships $\mathcal{I}$ and the corresponding matrices $\{\mathbf{S}^{(i,j)}\}$, parameters $\{\lambda^{(i,j)}\}$

Output: clustering results for each domain (inferred from $\mathbf{H}^{(\pi)}$)

```
begin
    Normalize all graph affinity matrices by Frobenius norm;
    foreach (i, j) ∈ I do
        Normalize non-zero rows of S^(i,j);
    end
    for π ← 1 to d do
        Initialize H^(π) with random values between (0,1];
    end
    repeat
        for π ← 1 to d do
            Update H^(π) by Eq. (7) or (10);
        end
    until convergence;
end
```

We formulate the Lagrange function for optimization

$$\begin{aligned} &L(\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, ..., \mathbf{H}^{(d)}) \\ &= \sum_{i=1}^{d} ||\mathbf{A}^{(i)} - \mathbf{H}^{(i)}(\mathbf{H}^{(i)})^{\mathrm{T}}||_F^2 \\ &+ \sum_{(i,j)\in\mathcal{I}} \lambda^{(i,j)} ||\mathbf{S}^{(i,j)}\mathbf{H}^{(i)}(\mathbf{S}^{(i,j)}\mathbf{H}^{(i)})^{\mathrm{T}} - \mathbf{H}^{(j)}(\mathbf{H}^{(j)})^{\mathrm{T}}||_F^2 \end{aligned} \tag{13}$$

Without loss of generality, we only show the derivation of the updating rule for one domain π ($\pi \in [1, d]$). The partial derivative of Lagrange function with respect to $\mathbf{H}^{(\pi)}$ is:

$$\begin{aligned} &\nabla_{\mathbf{H}^{(\pi)}} L = \\ &- \mathbf{A}^{(\pi)}\mathbf{H}^{(\pi)} + \mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\ &+ \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \\ &- \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)}(\mathbf{S}^{(\pi,j)})^{\mathrm{T}}\mathbf{H}^{(j)}(\mathbf{H}^{(j)})^{\mathrm{T}}\mathbf{S}^{(\pi,j)}\mathbf{H}^{(\pi)} \\ &- \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}\mathbf{S}^{(i,\pi)}\mathbf{H}^{(i)}(\mathbf{H}^{(i)})^{\mathrm{T}}(\mathbf{S}^{(i,\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \\ &+ \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}\mathbf{H}^{(\pi)}(\mathbf{H}^{(\pi)})^{\mathrm{T}}\mathbf{H}^{(\pi)} \end{aligned} \tag{14}$$

Using the Karush-Kuhn-Tucker (KKT) complementarity condition [4] for the non-negative constraint on $\mathbf{H}^{(\pi)}$, we have

$$\nabla_{\mathbf{H}^{(\pi)}} L \circ \mathbf{H}^{(\pi)} = \mathbf{0} \tag{15}$$

The above formula leads to the updating rule for $\mathbf{H}^{(\pi)}$ in Eq. (10).

3.3.2 Convergence

We use the auxiliary function approach [20] to prove the convergence of Eq. (10) in Theorem 3.2. We first introduce the definition of auxiliary function as follows.

Definition 3.3. $Z(h, \tilde{h})$ is an auxiliary function for $L(h)$ if the conditions

$$Z(h, \tilde{h}) \geq L(h) \qquad and \qquad Z(h, h) = L(h), \tag{16}$$

are satisfied for any given $h, \tilde{h}$ [20].

Lemma 3.4. *If Z is an auxiliary function for L, then L is non-increasing under the update [20].*

$$h^{(t+1)} = \operatorname*{argmin}_{h} Z(h, h^{(t)}) \tag{17}$$

Theorem 3.5. *Let $L(\mathbf{H}^{(\pi)})$ denote the sum of all terms in L containing $\mathbf{H}^{(\pi)}$. The following function*

$$\begin{aligned} Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)}) &= -2 \sum_{klm} \mathbf{A}^{(\pi)}_{ml} P(k,l,m) \\ &+ (1 + \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)}) \sum_{kl} \left(\tilde{\mathbf{H}}^{(\pi)} (\tilde{\mathbf{H}}^{(\pi)})^{\mathrm{T}} \tilde{\mathbf{H}}^{(\pi)}\right)_{kl} \cdot \frac{(\mathbf{H}^{(\pi)}_{kl})^4}{(\tilde{\mathbf{H}}^{(\pi)}_{kl})^3} \\ &- 2 \sum_{(i,\pi)\in\mathcal{I}} \lambda^{(i,\pi)} \sum_{klm} \left(\mathbf{S}^{(i,\pi)} \mathbf{H}^{(i)} (\mathbf{H}^{(i)})^{\mathrm{T}} (\mathbf{S}^{(i,\pi)})^{\mathrm{T}}\right)_{lm} P(k,l,m) \\ &+ \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)} \sum_{kl} (\mathbf{Q}(j))_{kl} \cdot \frac{(\mathbf{H}^{(\pi)}_{lk})^4}{(\tilde{\mathbf{H}}^{(\pi)}_{lk})^3} \\ &- 2 \sum_{(\pi,j)\in\mathcal{I}} \lambda^{(\pi,j)} \sum_{klm} \left((\mathbf{S}^{(\pi,j)})^{\mathrm{T}} \mathbf{H}^{(j)} (\mathbf{H}^{(j)})^{\mathrm{T}} \mathbf{S}^{(\pi,j)}\right)_{lm} P(k,l,m) \end{aligned} \tag{18}$$

is an auxiliary function for $L(\mathbf{H}^{(\pi)})$, where $P(k,l,m) = \tilde{\mathbf{H}}^{(\pi)}_{lk} \tilde{\mathbf{H}}^{(\pi)}_{mk} \left(1 + \log \frac{\mathbf{H}^{(\pi)}_{lk} \mathbf{H}^{(\pi)}_{mk}}{\tilde{\mathbf{H}}^{(\pi)}_{lk} \tilde{\mathbf{H}}^{(\pi)}_{mk}}\right)$ and $\mathbf{Q}(j) = (\tilde{\mathbf{H}}^{(\pi)})^{\mathrm{T}} (\mathbf{S}^{(\pi,j)})^{\mathrm{T}} \mathbf{S}^{(\pi,j)} \tilde{\mathbf{H}}^{(\pi)}$ $(\tilde{\mathbf{H}}^{(\pi)})^{\mathrm{T}} (\mathbf{S}^{(\pi,j)})^{\mathrm{T}} \mathbf{S}^{(\pi,j)}$. Furthermore, it is a convex function in $\mathbf{H}^{(\pi)}$ and has a global minimum.

Theorem 3.5 can be proved using a similar idea to that in [9] by validating $Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)}) \geq L(\mathbf{H}^{(\pi)})$, $Z(\mathbf{H}^{(\pi)}, \mathbf{H}^{(\pi)}) = L(\mathbf{H}^{(\pi)})$, and the Hessian matrix $\nabla\nabla_{\mathbf{H}^{(\pi)}} Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)}) \succeq \mathbf{0}$. Due to space limitation, we omit the details.

Based on Theorem 3.5, we can minimize $Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)})$ with respect to $\mathbf{H}^{(\pi)}$ with $\tilde{\mathbf{H}}^{(\pi)}$ fixed. We set $\nabla_{\mathbf{H}^{(\pi)}} Z(\mathbf{H}^{(\pi)}, \tilde{\mathbf{H}}^{(\pi)}) = \mathbf{0}$, and get the following updating formula

$$\mathbf{H}^{(\pi)} \leftarrow \tilde{\mathbf{H}}^{(\pi)} \circ \left(\frac{\mathbf{\Psi}(\tilde{\mathbf{H}}^{(\pi)})}{\mathbf{\Xi}(\tilde{\mathbf{H}}^{(\pi)})}\right)^{\frac{1}{4}},$$

which is consistent with the updating formula derived from the KKT condition aforementioned.

From Lemma 3.4 and Theorem 3.5, for each subsequent iteration of updating $\mathbf{H}^{(\pi)}$, we have $L((\mathbf{H}^{(\pi)})^0) = Z((\mathbf{H}^{(\pi)})^0, (\mathbf{H}^{(\pi)})^0) \geq Z((\mathbf{H}^{(\pi)})^1, (\mathbf{H}^{(\pi)})^0) \geq Z((\mathbf{H}^{(\pi)})^1, (\mathbf{H}^{(\pi)})^1) = L((\mathbf{H}^{(\pi)})^1) \geq ... \geq L((\mathbf{H}^{(\pi)})^{Iter})$. Thus $L(\mathbf{H}^{(\pi)})$ monotonically decreases. This is also true for the other variables. Since the objective function Eq. (6) is lower bounded by 0, the correctness of Theorem 3.2 is proved. Theorem 3.1 can be proven with a similar strategy.

3.3.3 Complexity Analysis

The time complexity of Algorithm 1 (for both loss functions) is $\mathcal{O}(Iter \cdot d|\mathcal{I}|(\tilde{n}^3 + \tilde{n}^2\tilde{k}))$, where $\tilde{n}$ is the largest n_π ($1 \leq \pi \leq d$), $\tilde{k}$ is the largest k_π and $Iter$ is the number of iterations needed before convergence. In practice, $|\mathcal{I}|$ and d are usually small constants. Moreover, from Eq. (10) and Eq. (7), we observe that the $\tilde{n}^3$ term is from the matrix multiplication $(\mathbf{S}^{(\pi,j)})^{\mathrm{T}} \mathbf{S}^{(\pi,j)}$. Since $\mathbf{S}^{(\pi,j)}$ is the input matrix and often very sparse, we can compute $(\mathbf{S}^{(\pi,j)})^{\mathrm{T}} \mathbf{S}^{(\pi,j)}$ in advance in sparse form. In this way, the complexity of Algorithm 1 is reduced to $\mathcal{O}(Iter \cdot \tilde{n}^2\tilde{k})$.

3.4 Re-Evaluating Cross-Domain Relationship

In real applications, the cross-domain instance relationship based on prior knowledge may contain noise. Thus, it is crucial to allow users to evaluate whether the provided relationships violate any single-domain clustering structures. In this section, we develop a principled way to archive this goal. In fact, we only need to slightly modify the co-regularization loss functions in Section 3.1.2 by multiplying a confidence matrix $\mathbf{W}^{(i,j)}$ to each $\mathbf{S}^{(i,j)}$. Each element in the confidence matrix $\mathbf{W}^{(i,j)}$ is initialized to 1. For RSS loss, we give the modified loss function below (the case for CD loss is similar).

$$\mathcal{J}^{(i,j)}_W = ||(\mathbf{W}^{(i,j)} \circ \mathbf{S}^{(i,j)})\mathbf{H}^{(i)} - \mathbf{H}^{(j)}||^2_F \tag{19}$$

Here, $\circ$ is element-wise product. By optimizing the following objective function, we can learn the optimal confidence matrix

$$\min_{\mathbf{W}\geq 0, \mathbf{H}^{(\pi)}\geq 0 (1\leq\pi\leq d)} \mathcal{O} = \sum_{i=1}^{d} \mathcal{L}^{(i)} + \sum_{(i,j)\in\mathcal{I}} \lambda^{(i,j)} \mathcal{J}^{(i,j)}_W \tag{20}$$

Eq. (20) can be optimized by iteratively implementing following two steps until convergence: 1) replace $\mathbf{S}^{(\pi,j)}$ and $\mathbf{S}^{(i,\pi)}$ in Eq. (7) with $(\mathbf{W}^{(\pi,j)} \circ \mathbf{S}^{(\pi,j)})$ and $(\mathbf{W}^{(i,\pi)} \circ \mathbf{S}^{(i,\pi)})$ respectively, and use the replaced formula to update each $\mathbf{H}^{(\pi)}$; 2) use the following formula to update each $\mathbf{W}^{(i,j)}$

$$\mathbf{W}^{(i,j)} \leftarrow \mathbf{W}^{(i,j)} \circ \sqrt{\frac{(\mathbf{H}^{(j)}(\mathbf{H}^{(i)})^{\mathrm{T}}) \circ \mathbf{S}^{(i,j)}}{((\mathbf{W}^{(i,j)} \circ \mathbf{S}^{(i,j)})\mathbf{H}^{(i)}(\mathbf{H}^{(i)})^{\mathrm{T}}) \circ \mathbf{S}^{(i,j)}}} \tag{21}$$

Here, $\sqrt{\cdot}$ is element-wise square root. Note that many elements in $\mathbf{S}^{(i,j)}$ are 0. We only update the elements in $\mathbf{W}^{(i,j)}$ whose corresponding elements in $\mathbf{S}^{(i,j)}$ are positive. In the following, we only focus on such elements. The learned confidence matrix minimizes the inconsistency between the original single-domain clustering structure and the prior cross-domain relationship. Thus for any element $\mathbf{W}^{(i,j)}_{a,b}$, the smaller the value, the stronger the inconsistency between $\mathbf{S}^{(i,j)}_{a,b}$ and single-domain clustering structures in $\mathcal{D}_i$ and $\mathcal{D}_j$. Therefore, we can sort the values of $\mathbf{W}^{(i,j)}$ and report to users the smallest elements and their corresponding cross-domain relationships. Accurate relationship can help to improve the overall results. On the other hand, inaccurate relationship may provide wrong guidance of the clustering process. Our method allows the users to examine these critical relationships and improve the accuracy of the results.

Table 2: The UCI benchmarks

Identifier	#Instances	#Attributes
Iris	100	4
Wine	119	13
Ionosphere	351	34
WDBC	569	30

4. EMPIRICAL STUDY

In this section, we present extensive experimental results on evaluating the performance of our method.

4.1 Effectiveness Evaluation

We evaluate the proposed method by clustering benchmark data sets from the UCI Archive [1]. We use four data sets with class label information, namely Iris, Wine, Ionosphere and Breast Cancer Wisconsin (Diagnostic) data sets. They are from four different domains. To make each data set contain the same number of (e.g., two) clusters, we follow the preprocessing step in [33] to remove the SETOSA class from the Iris data set and Class 1 from the Wine data set. The statistics of the resulting data sets are shown in Table 2.

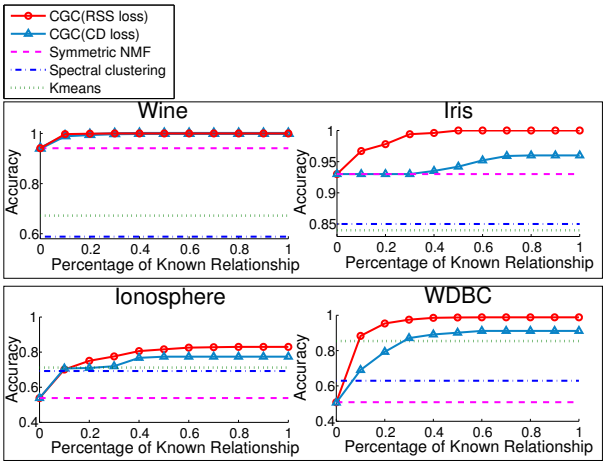


Figure 2: Clustering results on UCI datasets(Wine v.s. Iris, Ionosphere v.s. WDBC)

For each data set, we compute the affinity matrix using the RBF kernel [4]. Our goal is to examine whether cross-domain relationship can help to enhance the accuracy of the clustering results. We construct two cross-domain relationships: Wine-Iris and Ionosphere-WDBC. The relationships are generated based on the class labels, i.e., positive (negative) instances in one domain can only be mapped to positive (negative) instances in another domain. We use the widely used Clustering Accuracy [35] to measure the quality of the clustering results. Parameter λ is set to 1 throughout the experiments. Since no existing method can handle the multi-domain co-regularized graph clustering problem, we compare our CGC method with three representative single-domain methods: symmetric NMF [17], K-means [26] and spectral clustering [24]. We report the accuracy when varying the available cross-domain instance relationships (from 0 to 1 with 10% increment). The accuracy shown in Fig. 2 is averaged over 100 sets of randomly generated relationships.

We have several key observations from Fig. 2. First, CGC significantly outperforms all single-domain graph clustering methods, even though single-domain methods may perform differently on different data sets. For example, symmetric NMF works better on Wine and Iris data sets, while K-means works better on Ionosphere and WDBC data sets. Note that when the percentage of available relationships is 0, CGC degrades to symmetric NMF. CGC outperforms all alternative methods when cross-domain relationships are available. This demonstrates the effectiveness of the cross-domain relationship co-regularized method. We also notice that the performance of CGC dramatically improves when the available relationships increase from 0 to 30%, suggesting that our method can effectively improve the clustering result even with limited information on cross-domain relationship. This is crucial for many real-life applications. Finally, we can see that RSS loss is more effective than CD loss. This is because RSS loss directly measures the weights of clustering assignment, while the CD loss does this indirectly by using linear kernel similarity first (see Section 3.1). Thus, for a given percentage of cross-domain relationships, the method using RSS loss gains more improvements over the single-domain clustering than that using CD loss.

4.2 Robustness Evaluation

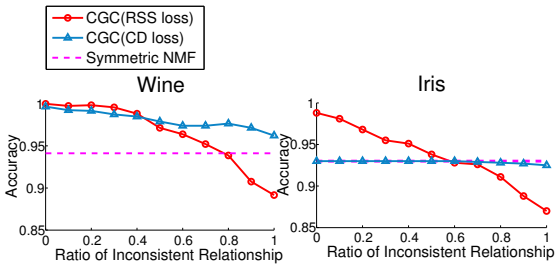
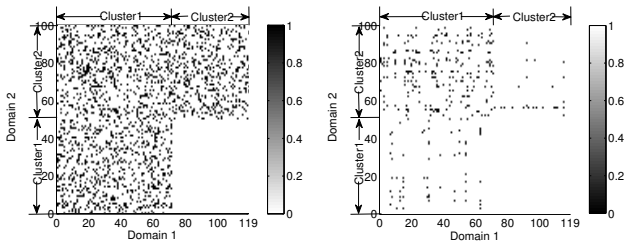


Figure 3: Clustering with inconsistent cross-domain relationship



(a) S (b) W

Figure 4: Relationship matrix S and confidence matrix W on Wine-Iris data set)

In real-life applications, both graph data and cross-domain instance relationship may contain noise. In this section, we 1) evaluate whether CGC is sensitive to the inconsistent relationships, and 2) study the effectiveness of the relationship re-evaluation strategy proposed in Section 3.4. Due to space limitation, we only report the results on Wine-Iris data set used in the previous section. Similar results can be observed in other data sets.

We add inconsistency into matrix $\mathbf{S}$ with ratio r. The results are shown in Fig. 3. The percentage of available cross-domain relationships is fixed at 20%. Single-domain symmetric NMF is used as a reference method. We observe that, even when the inconsistency ratio r is close to 50%, CGC still outperforms the single-domain symmetric NMF method. This indicates that our method is robust to noisy relationships. We also observe that, when r is very large, CD loss works better than RSS loss, although when r is small, RSS loss outperforms the CD loss (as discussed in Section 4.1). When r reaches 1, the relationship is full of noise. From the figure, we can see that CD loss is immune to noise.

In Section 3.4, we provide a method to report the cross-domain relationships that violate the single-domain clustering structure. We still use the Wine-Iris data set to evaluate its effectiveness. As shown in Fig. 4, in the relationship matrix $\mathbf{S}$, each black point represents a cross-domain relationship (all with value 1) mapping classes between the two domains. We leave the bottom right part of the matrix blank intentionally so that the inconsistent relationships only appear between instances in cluster 1 of domain 1 and cluster 2 of domain 2. The learned confidence matrix $\mathbf{W}$ is shown in the figure (entries normalized to [0,1]). The smaller the value is, the stronger the evidence that the cross-domain relationship violates the original single-domain clustering structure. Reporting these suspicious relationships to users will allow them to examine the cross-domain relationships that are likely resulting from inaccurate prior knowledge.

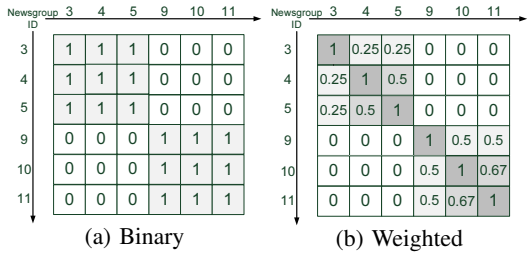
Table 3: The newsgroup data

Group Id	Label
3	comp.os.ms-windows.misc
4	comp.sys.ibm.pc.hardware
5	comp.sys.mac.hardware
9	rec.motorcycles
10	rec.sport.baseball
11	rec.sport.hockey

4.3 Binary v.s. Weighted Relationship

In this section, we demonstrate that CGC can effectively incorporate weighted cross-domain relationship, which may carry richer information than binary relationship. The 20 Newsgroup data set[1] contains documents organized by a hierarchy of topic classes. We choose 6 groups as shown in Table 3. For example, at the top level, the 6 groups belong to two topics, computer (groups {3,4,5}) or recreation (groups {9,10,11}). The computer related data sets can be further partitioned into two subcategories, os (group 3) and sys (groups {4, 5}). Similarly, the recreation related data sets consist of subcategories motocycles (group 9) and sport (groups 10 and 11).

We generate two domains, each contains randomly sampled 300 documents from the 6 groups (50 documents from each group). To generate binary relationships, two articles are related if they are from the same high-level topic, i.e., computer or recreation, as shown in Fig. 5(a). Weighted relationships are generated based on the topic hierarchy. Given two group labels, we compute the longest common prefix. The weight is assigned to be the ratio of the length of the common prefix over the length of the shorter of the two labels. The weighted relationship matrix is shown in Fig. 5(b). For example, if two documents come from the same group, we set the corresponding entry to 1; if one document is from rec.sport.baseball and the other from rec.sport.hockey, we set the corresponding entry to 0.67; if they do not share any label term at all, we set the entry to 0.



(a) Binary (b) Weighted

Figure 5: Binary and weighted relationship matrices

We perform experiments using binary and weighted relationships respectively. The affinity matrix of documents is computed based on cosine similarity. We cluster the data set into either 2 or 6 clusters and results are shown in Fig. 6. We observe that when each domain is partitioned into 2 clusters, the binary relationship outperforms the weighted one. This is because the binary relationship better represents the top-level topics, computer and recreation. On the other hand, for the domain partitioned into 6 clusters, the weighted relationship performs significantly better than the binary one. This is because weights provide more detailed information on cross-domain relationships than the binary relationships.

4.4 Performance Evaluation

[1]http://qwone.com/ jason/20Newsgroups/

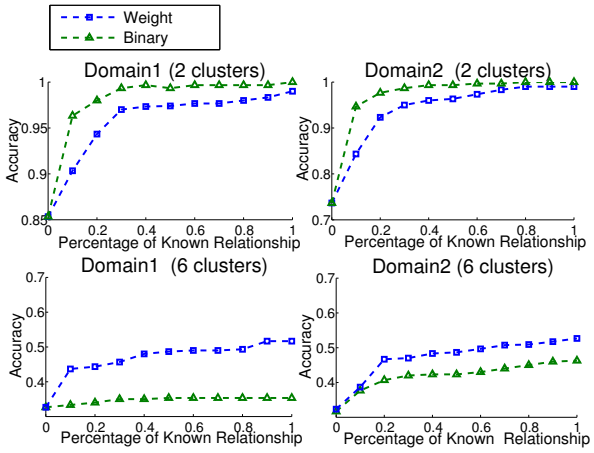


Figure 6: Clustering results on the newsgroup data set with binary or weighted relationships

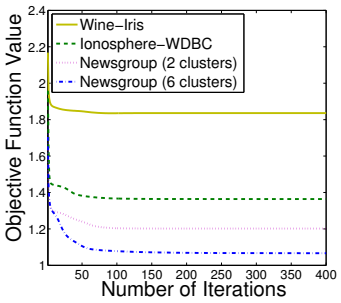


Figure 7: Number of iterations to converge (CGC)

In this section, we study the performance of the proposed methods: the number of iterations before converging to a local optima. Fig. 7 shows the value of the objective function with respect to the number of iterations on different data sets. We observe that the objective function value decreases steadily with more iterations. Usually, less than 100 iterations are needed before convergence.

4.5 Protein Module Detection by Integrating Multi-Domain Heterogenous Data

In this section, we apply the proposed method to detect protein functional modules [16]. The goal is to identify clusters of proteins that have strong interconnection with each other. A common approach is to cluster the protein-protein interaction (PPI) networks [2]. We show that, by integrating multi-domain heterogenous information, such as gene co-expression network [15] and genetic interaction network [7], the performance of the detection algorithm can be dramatically improved.

We download the widely used human PPI network from BioGrid[2]. Three Hypertension related gene expression data sets are downloaded from Gene Expression Ominbus[3] with ids GSE2559, GSE703, and GSE4737. In total, 5412 genes included in all three data sets are used to construct gene co-expression network. Pearson correlation coefficients(normalized between [0 1]) are used as the weights on edges between genes. The genetic interaction network is con-

[2]http://thebiogrid.org/download.php
[3]http://www.ncbi.nlm.nih.gov/gds

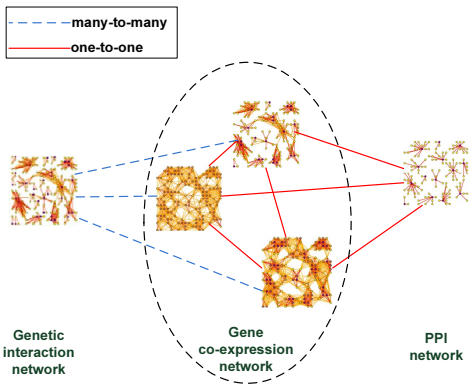


Figure 8: Protein-protein interaction network, gene co-expression network, genetic interaction network and cross-domain relationships

structed using a large-scale Hypertension genetic data [10], which contains 490032 genetic markers across 4890 (1952 disease and 2938 healthy) samples. We use 1 million top-ranked genetic marker-pairs to construct the network and the test statistics are used as the weights on the edges between markers [36]. The constructed heterogenous networks are shown in Fig. 8. The relationship between genes and genetic markers is many-to-many, since multiple genetic markers may be covered by a gene and each marker may be covered by multiple genes due to the overlapping between genes. The relationship between proteins and genes is one-to-one.

We apply CGC (with RSS loss) to cluster the generated multi-domain graphs. We use the standard Gene Set Enrichment Analysis (GSEA) [23] to evaluate the significance of the inferred clusters. In particular, for each inferred cluster (protein/gene set) T, we identify the most significantly enriched Gene Ontology categories [31, 6]. The significance (p-value) is determined by the Fisher's exact test. The raw p-values are further calibrated to correct for the multiple testing problem [34]. To compute calibrated p-values for each T, we perform a randomization test, wherein we apply the same test to 1000 randomly created gene sets that have the same number of genes as T.

The calibrated p-values of the gene sets learned by CGC and single-domain graph clustering methods, Ncut [25], symmetric NMF [17], Markov clustering [32] and spectral clustering, when applied on PPI network, are shown in Fig. 9. The clusters are arranged in ascending order of their p-values. As can be seen from the figure, by integrating three types of heterogenous networks, CGC achieves better performance than the single-domain methods. Table 4 shows the number of significant (calibrated p-value ≤ 0.05) modules identified by different methods. We find that CGC reports more significant functional modules than the single-domain methods. We also apply existing multi-view graph clustering method [19, 30] on the gene co-expression networks and PPI network. Since these four networks are of the same size, multi-view method can be applied. In total, less than 20 significant modules are identified. This is because the gene expression data are very noisy. Multi-view graph clustering methods forced to find one common clustering assignment over different data sets and thus are more sensitive to noise.

5. RELATED WORK

To our best knowledge, this is the first work to study co-regularized multi-domain graph clustering with many-to-many cross-domain relationship. Existing work on multi-view graph clustering relies on a fundamental assumption that all views are with respect to the same set of instances. This set of instances have multiple representations and different views are generated from the same underlying distribution [5]. In multi-view graph clustering, research has been done to explore the most consensus clustering structure from different views [18, 19, 30]. Another common approach in multi-view graph clustering is a two-step approach, which first combines multiple views into one view, then does clustering on the resulting view [29, 37]. However, these methods do not address the many-to-many cross-domain relationship. Note that our work is different from transfer clustering [8] and multi-task clustering [14]. These methods assume that there are some common features shared by different domains. They are also not designed for graph data.

Clustering ensemble approaches also aim to find consensus clusters from multiple data sources. Strehl and Ghosh [27] proposed instance-based and cluster-based approaches for combining multiple partitions. Fern and Brodley [12] developed a hybrid bipartite graph formulation to infer ensemble clustering result. These approaches try to combine multiple clustering structures for a set of instances into a single consolidated clustering structure. Similar to multi-view graph clustering, they cannot handle many-to-many cross-domain relationships.

Method	Number of significant modules
Markov Clustering	21
NCut	25
Spectral Clustering	44
Symmetric NMF	77
CGC	**84**

Table 4: Gene Ontology (GO) enrichment analysis of the gene sets identified by different methods

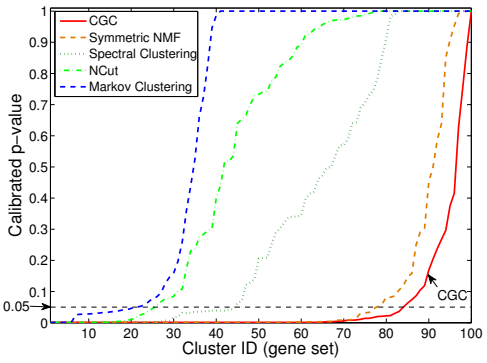


Figure 9: Comparison of CGC and single-domain graph clustering ($k = 100$)

6. CONCLUSION AND DISCUSSION

Integrating multiple data sources for graph clustering is an important problem in data mining research. Robust and flexible approaches that can incorporate multiple sources to enhance graph clustering performance are highly desirable. We develop CGC, which utilizes cross-domain relationship as co-regularizing penalty to guide the search of consensus clustering structure. CGC is robust even when the cross-domain relationships based on prior knowledge are noisy. Using various benchmark and real-life data sets, we show that the proposed CGC method can dramatically

improve the graph clustering performance compared with single-domain methods.

7. ACKNOWLEDGMENTS

The work was partially supported by US National Science Foundation IIS-0812464, IIS-1313606, IIS-1162374 and IIS-1218036.

References

[1] A. Asuncion and D. Newman. Uci machine learning repository. 2007.

[2] S. Asur, D. Ucar, and S. Parthasarathy. An ensemble framework for clustering protein-protein interaction networks. In *Bioinformatics*, pages 29–40, 2007.

[3] S. Bickel and T. Scheffer. Multi-view clustering. In *ICDM*, pages 19–26, 2004.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, pages 129–136, 2009.

[6] W. Cheng, X. Zhang, Y. Wu, X. Yin, J. Li, D. Heckerman, and W. Wang. Inferring novel associations between snp sets and gene sets in eqtl study using sparse graphical model. In *BCB'12*, pages 466–473, 2012.

[7] H. J. Cordell. Detecting gene-gene interactions that underlie human diseases. *Nat. Rev. Genet.*, 10:392–404, 2009.

[8] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Self-taught clustering. In *ICML*, pages 200–207, 2008.

[9] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *KDD*, pages 126–135, 2006.

[10] T. Feng and X. Zhu. Genome-wide searching of rare genetic variants in WTCCC data. *Hum. Genet.*, 128:269–280, 2010.

[11] D. Fenyo, editor. *Computational Biology*. Methods in Molecular Biology. 2010.

[12] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML*, pages 36–45, 2004.

[13] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *NIPS*, pages 585–593, 2009.

[14] Q. Gu, Z. Li, and J. Han. Learning a kernel for multi-task clustering. In *AAAI*, 2011.

[15] S. Horvath and J. Dong. Geometric interpretation of gene coexpression network analysis. *PLoS Computational Biology*, 4, 2008.

[16] J. S. Hub and B. L. de Groot. Detection of functional modes in protein dynamics. *PLoS Computational Biology*, 2009.

[17] D. Kuang, H. Park, and C. H. Q. Ding. Symmetric nonnegative matrix factorization for graph clustering. In *SDM*, pages 106–117, 2012.

[18] A. Kumar and H. D. III. A co-training approach for multi-view spectral clustering. In *ICML*, pages 393–400, 2011.

[19] A. Kumar, P. Rai, and H. D. III. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

[20] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

[21] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *KDD*, pages 420–429, 2007.

[22] B. Long, P. S. Yu, and Z. M. Zhang. A general model for multiple view unsupervised learning. In *SDM*, pages 822–833, 2008.

[23] V. K. Mootha, C. M. Lindgren, K. F. Eriksson, A. Subramanian, S. Sihag, J. Lehar, P. Puigserver, E. Carlsson, M. Ridderstrale, E. Laurila, N. Houstis, M. J. Daly, N. Patterson, J. P. Mesirov, T. R. Golub, P. Tamayo, B. Spiegelman, E. S. Lander, J. N. Hirschhorn, D. Altshuler, and L. C. Groop. PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.*, 34(3):267–273, 2003.

[24] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *CVPR*, pages 888–905, 1997.

[26] H. Späth. *Cluster Dissection and Analysis. Theory, FORTRAN programs, examples*. Ellis Horwood, 1985.

[27] A. Strehl, J. Ghosh, and C. Cardie. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[28] Y. Sun and J. Han. *Mining Heterogeneous Information Networks: Principles and Methodologies*. 2012.

[29] L. Tang, X. Wang, and H. Liu. Community detection via heterogeneous interaction analysis. *Data Min. Knowl. Discov*, 25:1–33, 2012.

[30] W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In *ICDM*, pages 1016–1021, 2009.

[31] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[32] S. van Dongen. A cluster algorithm for graphs. In *Centrum voor Wiskunde en Informatica (CWI)*, page 40. 2000.

[33] X. Wang and I. Davidson. Flexible constrained spectral clustering. In *KDD*, pages 563–572, 2010.

[34] P. H. Westfall and S. S. Young. *Resampling-based Multiple Testing*. Wiley, New York, 1993.

[35] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR*, Clustering, pages 267–273, 2003.

[36] X. Zhang, S. Huang, F. Zou, and W. Wang. TEAM: Efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics*, 26(12):i217–227, 2010.

[37] D. Zhou and C. J. C. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, pages 1159–1166, 2007.