Clustering on Complex Graphs

Bo Long and **Mark (Zhongfei) Zhang**
Computer Science Dept., SUNY Binghamton
Binghamton, NY 13902
{blong1, zzhang}@binghamton.edu

Philip S. Yu
Dept. of Computer Science, UIC at Chicago
Chicago, IL, 60607
psyu@cs.uic.edu

Tianbing Xu
Computer Science Dept., SUNY Binghamton
Binghamton, NY 13902
txu@cs.binghamton.edu

Abstract

Complex graphs, in which multi-type nodes are linked to each other, frequently arise in many important applications, such as Web mining, information retrieval, bioinformatics, and epidemiology. In this study, We propose a general framework for clustering on complex graphs. Under this framework, we derive a family of clustering algorithms including both hard and soft versions, which are capable of learning cluster patterns from complex graphs with various structures and statistical properties. We also establish the connections between the proposed framework and the traditional graph partitioning approaches. The experimental evaluation provides encouraging results to validate the proposed framework and algorithms.

Introduction

Graph clustering has traditionally focused on homogeneous graphs consisting of nodes of a single type. However, many examples of real-world graphs involve multi-type nodes linking to each other (we refer them as complex graphs in this study). There are two types of links in a complex graph: homogeneous links within the same type of nodes and heterogeneous links between two different types of nodes. For example, in a complex graph consisting of paper nodes and author nodes from a scientific publication domain, papers are linked to each other by citations (homogeneous links) and papers and authors are linked to each other by authorship (heterogeneous links); in a complex graph from Web search system, Web pages are linked to each other by homogeneous links and there are heterogeneous links between Web pages, search queries, and Web users; a collection of documents has been formulated as a bi-partite complex graphs with heterogeneous links between document nodes and word nodes (Dhillon 2001).

In general, it is very difficult for traditional graph clustering approaches to learn hidden cluster patterns from a complex graph, since they simply treat all the nodes as the same type. For example, when applying a traditional graph partitioning approach to a complex graph, we face two problems: how to cut different types of links to get meaningful partitions; how to interpret the resulting partitions consisting of different types of nodes. Note that different types of nodes have different cluster structures; partitioning them into the same groups cannot tell this difference.

Another intuitive thought for complex graph learning is to transform a complex graph into a set of small graphs of single-type nodes and learn from each small graph individually. However, it is very difficult, if not impossible, to do the transformation without information loss. Furthermore, learning the hidden structures for each type of nodes individually cannot provide useful global structures of a complex graph. For example, for a complex graph of document and word nodes, besides the cluster structures for documents and words, respectively, the relations between document clusters and word clusters are also very useful.

Hence, complex graphs have presented a great challenge to graph clustering. In this paper, we propose a general framework for clustering on complex graphs. By representing a complex graph as a set of related matrices, clustering on a complex graph is formulated as the problem of related matrix approximation under an certain distance function. Under this framework, we consider both hard and soft clustering. A hard clustering algorithm is derived and generalized to various Bregman divergences. The soft clustering algorithms are derived based on the bound optimization procedure (Salakhutdinov and Roweis 2003). We also investigate the relation between the proposed framework and the edge cut objectives to provide a unified view to traditional graph partitioning approaches.

Problem Formulation

Suppose that we are given an undirected complex graph with non-negative edge weights, $G = (\{\mathcal{V}_i\}_{i=1}^m, \mathcal{E})$, where $\{\mathcal{V}_i\}_{i=1}^m$ denotes m sets of different types of nodes, and $\mathcal{E}$ denotes the edges in the graph. Rather than using one affinity matrix to represent the whole graph, we represent a complex graph as a set of related matrices $\{\{\mathbf{S}^{(i)} \in \mathbb{R}_+^{n_i \times n_i}\}_{i=1}^m, \{\mathbf{A}^{(ij)} \in \mathbb{R}_+^{n_i \times n_j}\}_{i,j=1}^m\}$, where $\mathbf{S}^{(i)}$ represents the edge weights for the homogeneous links within $\mathcal{V}_i$; $\mathbf{A}^{(ij)}$ represents the edge weights for the heterogeneous links between $\mathcal{V}_i$ and $\mathcal{V}_j$; n_i denotes the number of nodes in $\mathcal{V}_i$. For convenience, here we assume that every type of nodes has homogeneous links within them and heterogeneous links to all other types of nodes.

We let $\mathbf{C}^i \in \mathbb{R}_+^{n_i \times k_i}$ denote the cluster membership matrix for each type of nodes $\mathcal{V}_i$ such that $\mathbf{C}_{pq}^{(i)}$ denote the weight that the pth node in $\mathcal{V}_i$ is associated with the qth cluster. The intra-type cluster pattern matrix $\mathbf{D}^{(i)} \in \mathbb{R}^{k_i \times k_i}$ denotes the link patterns within the same type of nodes such that $\mathbf{D}_{pq}^{(i)}$ denotes the link strength between the pth cluster and the qth cluster of $\mathcal{V}_i$. The inter-type pattern matrix $\mathbf{B}^{(ij)} \in \mathbb{R}^{k_i \times k_j}$ denotes the link patterns between the different types of nodes such that $\mathbf{B}_{pq}^{(ij)}$ denotes the link strength between the pth cluster of $\mathcal{V}_i$ and the qth cluster of $\mathcal{V}_j$. In the above formulation, k_i is a given number of clusters. Then, we formulate the objective function of clustering on a complex graph as following,

$$\begin{aligned} L &= \sum_{i=1}^{m} w^{(i)} \mathfrak{D}(\mathbf{S}^{(i)}, \mathbf{C}^{(i)} \mathbf{D}^{(i)} (\mathbf{C}^{(i)})^T) \\ &\quad + \sum_{i,j=1}^{m} w^{(ij)} \mathfrak{D}(\mathbf{A}^{(ij)}, \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^T), \end{aligned} \tag{1}$$

where $\mathfrak{D}$ denotes a given distance function, w^i and w^{ij} denote the given weight coefficients for users to express the importance for different sets of links.

In this study, we examine both hard and soft clustering tasks, which are formulated as follows. In the following formulation, $\mathbf{1}$ denotes a vector consisting 1's and $\mathbf{I}$ denotes an identity matrix.

- **Hard clustering** $\mathbf{C}^{(i)}$ is an cluster indicator matrix.

  $$\arg\min_{\mathbf{C}^{(i)} \in \{0,1\}^{n_i \times k_i}, \mathbf{C}^{(i)}\mathbf{1}=\mathbf{1}, \mathbf{D}^{(i)} \in \mathbb{R}^{k_i \times k_i}, \mathbf{B}^{(ij)} \in \mathbb{R}^{k_i \times k_j}} L \tag{2}$$

- **Soft clustering** $\mathbf{C}^{(i)}$ is non-negative and with appropriate normalization it can be interpreted as the soft weights for the cluster membership.

  $$\arg\min_{\mathbf{C}^{(i)} \in \mathbb{R}_+^{n_i \times k_i}, \mathbf{D}^{(i)} \in \mathbb{R}_+^{k_i \times k_i}, \mathbf{B}^{(ij)} \in \mathbb{R}_+^{k_i \times k_j}} L \tag{3}$$

The proposed framework has the following advantages. First, for a large complex graph, the framework avoids the computation difficulty of working on one huge matrix representing the whole graph. Second, the framework is flexible to make use of different types of link information to identify the cluster structures for each type of nodes. Third, the framework allows the cluster structures of different types of nodes to interact with each other, i.e., clustering one type of nodes can be viewed as a dynamic dimension reduction for other types of nodes. Fourth, cluster pattern matrices $\mathbf{D}^{(i)}$ and $\mathbf{B}^{(ij)}$ provide an intuitive summary of link patterns in a complex graph.

To avoid the clutter, in the following algorithm derivation, we work on a basic type of complex graph, $G = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$, in which there exist homogeneous links within $\mathcal{V}_1$ and heterogeneous links between $\mathcal{V}_1$ and $\mathcal{V}_2$. However, all the discussions in the rest of the paper can be easily extended to a general complex graph. The basic type graph can be represented as $\{\mathbf{S} \in \mathbb{R}_+^{n_1 \times n_1}, \mathbf{A} \in \mathbb{R}_+^{n_1 \times n_2}\}$. Then, the objective function is simplified as follows (for convenience, we omit weight coefficients),

$$L = \mathfrak{D}(\mathbf{S}, \mathbf{C}^{(1)}\mathbf{D}(\mathbf{C}^{(1)})^T) + \mathfrak{D}(A, \mathbf{C}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T), \tag{4}$$

Hard clustering on complex graphs

First we derive a clustering algorithm for complex graphs based on the most popular distance function, Euclidean distance function. Under Euclidean distance function, our task is

$$\min_{\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \mathbf{D}, \mathbf{B}} ||\mathbf{S} - \mathbf{C}^{(1)}\mathbf{D}(\mathbf{C}^{(1)})^T||^2 + ||A - \mathbf{C}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T||^2. \tag{5}$$

We present the following theorem which is the basis of our algorithm.

Theorem 1. *If* $\mathbf{C}^{(1)} \in \{0,1\}^{n_1 \times k_1}$, $\mathbf{C}^{(2)} \in \{0,1\}^{n_2 \times k_2}$, $\mathbf{D} \in \mathbb{R}^{k_1 \times k_1}$ *and* $\mathbf{B} \in \mathbb{R}^{k_1 \times k_2}$ *is the optimal solution to the minimization in* (5)*, then*

$$\mathbf{D} = ((\mathbf{C}^{(1)})^T\mathbf{C}^{(1)})^{-1}(\mathbf{C}^{(1)})^T\mathbf{S}\mathbf{C}^{(1)}((\mathbf{C}^{(1)})^T\mathbf{C}^{(1)})^{-1} \tag{6}$$

$$\mathbf{B} = ((\mathbf{C}^{(1)})^T\mathbf{C}^{(1)})^{-1}(\mathbf{C}^{(1)})^T\mathbf{A}\mathbf{C}^{(2)}((\mathbf{C}^{(2)})^T\mathbf{C}^{(2)})^{-1} \tag{7}$$

Proof. Let L denote the objective function in (5). L can be expanded as follows.

$$\begin{aligned} L &= \operatorname{tr}((\mathbf{S} - \mathbf{C}^{(1)}\mathbf{D}(\mathbf{C}^{(1)})^T)^T(\mathbf{S} - \mathbf{C}^{(1)}\mathbf{D}(\mathbf{C}^{(1)})^T)) \\ &\quad + \operatorname{tr}((\mathbf{A} - \mathbf{C}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T)^T(\mathbf{A} - \mathbf{C}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T)) \\ &= \operatorname{tr}(\mathbf{S}^T\mathbf{S}) - 2\operatorname{tr}(\mathbf{C}^{(1)}\mathbf{D}^T(\mathbf{C}^{(1)})^T\mathbf{S}) + \\ &\quad \operatorname{tr}(\mathbf{C}^{(1)}\mathbf{D}^T(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}\mathbf{D}(\mathbf{C}^{(1)})^T) + \operatorname{tr}(\mathbf{A}^T\mathbf{A}) \\ &\quad - 2\operatorname{tr}(\mathbf{C}^{(2)}\mathbf{B}^T(\mathbf{C}^{(1)})^T\mathbf{A}) \\ &\quad + \operatorname{tr}(\mathbf{C}^{(2)}\mathbf{B}^T(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T). \end{aligned}$$

Take the derivative with respect to $\mathbf{D}$ and $\mathbf{B}$, we obtain

$$\frac{\partial L}{\partial \mathbf{D}} = -2(\mathbf{C}^{(1)})^T\mathbf{S}\mathbf{C}^{(1)} + 2(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}D(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}. \tag{8}$$

$$\frac{\partial L}{\partial \mathbf{B}} = -2(\mathbf{C}^{(1)})^T\mathbf{A}\mathbf{C}^{(3)} + 2(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T\mathbf{C}^{(2)}. \tag{9}$$

According to KKT conditions, we solve $\frac{\partial L}{\partial \mathbf{D}} = 0$ and $\frac{\partial L}{\partial \mathbf{B}} = 0$ to obtain Eq.(6) and Eq.(7). This completes the proof of the theorem. □

Based on Theorem 1, we propose an alternative optimization algorithm, which alternatively updates $\mathbf{D}$, $\mathbf{B}$, $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ until convergence. Eqs (6) and (7) provide the updating rules for $\mathbf{D}$ and $\mathbf{B}$, respectively, while $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ are fixed. The updating rules (6) and (7) can be implemented more efficiently than it appears. First, they do not really involve computing inverse matrices, since $(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}$ and

$(\mathbf{C}^{(2)})^T\mathbf{C}^{(2)}$ are special diagonal matrices with the size of each cluster on its diagonal; second, the product of C^TAC can be calculated without matrix multiplication, since $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ are indicator matrices.

Then, we fix $\mathbf{D}$, $\mathbf{B}$ and $\mathbf{C}^{(2)}$ to update $\mathbf{C}^{(1)}$. Since each row of $\mathbf{C}^{(1)}$ is an indicator vector with only one element equal to 1, we adopt the re-assignment procedure to update $\mathbf{C}^{(1)}$ row by row. To update hth row of $\mathbf{C}^{(1)}$, we have

$$\mathbf{C}^{(1)}_{hp^*} = 1 \text{ for } p^* = \arg\min_p L_p \tag{10}$$

where L_p denotes the objective function when the hth node is assigned to the pth cluster. The updating rule re-assigns each node to the cluster such that the objective function is minimized at the current step. Note that the necessary computation does not involve the whole objective function. Similarly, we have the updating rule for $\mathbf{C}^{(2)}$,

$$\mathbf{C}^{(2)}_{hp^*} = 1 \text{ for } p^* = \arg\min_p L_p. \tag{11}$$

Since different complex graphs from different applications may have different statistical properties, Euclidean distance, which corresponds to normal distribution assumption, is not appropriate for all graphs. A large number of useful distance functions, such as Euclidean distance, generalized I-divergence, and KL divergence, can be generalized as the Bregman divergences (S.D.Pietra 2001; Banerjee et al. 2004b), which correspond to a large number of exponential family distributions. Due to the following nice properties of Bregman divergences, Theorem 1 holds true for all Bregman divergences.

Theorem 2. *Let X be a random variable taking values in $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset S \subseteq \mathbb{R}^d$ following distribution v. Given a Bregman divergence $D_\phi : S \times \mathit{int}(S) \mapsto [0, \infty)$, the problem*

$$\min_{s \in S} E_v[D_\phi(X, s)] \tag{12}$$

has a unique minimizer given by $s^* = E_v[X]$.

The proof of Theorem 2 is omitted (please refer to (S.D.Pietra 2001; Banerjee et al. 2004b)). Theorem 2 states the optimal estimation of a Bregman representative is always the mean of a sample. In the objective function (4) under a Bregman divergence, $\mathbf{D}$ and $\mathbf{B}$ contains Bregman representatives of edge weights between or within clusters of nodes. When cluster memberships $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ are given, according to Theorem 2, the optimal $\mathbf{B}_{pq}$ is obtained as the mean edge weights between the pth cluster of $\mathcal{V}_1$ and the qth cluster of $\mathcal{V}_2$

$$\mathbf{B}_{pq} = \frac{1}{|\pi^{(1)}_p||\pi^{(2)}_q|} \sum_{i \in \pi^{(1)}_p, j \in \pi^{(2)}_q} \mathbf{A}_{ij}, \tag{13}$$

where $\pi^{(1)}_p$ and $\pi^{(2)}_q$ denote the pth cluster of $\mathcal{V}_1$and the qth cluster of $\mathcal{V}_2$, respectively. If we write Eq (13) in a matrix form by using the cluster indicator matrices, we obtain Eq. (7), i.e., Eq. (7) holds true for all Bregman divergences. Similarly, Eq. (6) holds true for all Bregman divergences. As for updating rules (10) and (11), they work well for any distance function as long as the objective function is calculated based on the corresponding distance function.

Hence, we obtain a general clustering algorithm for complex graphs under various Bregman divergences, which iteratively updates cluster indicator matrices and cluster pattern matrices by using the updating rules (6), (7), (10) and (11) (similar updating rules for a general complex graph also hold true) until convergence. The computational complexity of the algorithm is $O(tmn^2k)$ for a general complex graph, where t denotes the number of iterations, $n = \max\{n_i\}_{i=1}^m$ and $k = \max\{k_i\}_{i=1}^m$. Based on Theorem 1, Theorem 2, and the criteria for reassignment in (10) and (11), the objective function is non-increasing under these updating rules. Therefore, the convergence of the algorithm is guaranteed.

Soft clustering on complex graphs

In this section, we derive alternative optimization algorithms for soft clustering on complex graphs based on the ideas of the bound optimization procedure (Salakhutdinov and Roweis 2003; Lee and Seung 2000). The basic idea is to construct an auxiliary function which is a convex upper bound for the original objective function based on the solution obtained from the previous iteration. Then, a new solution to the current iteration is obtained by minimizing this upper bound. The definition of the auxiliary function and a useful lemma (Lee and Seung 2000) are given as follows.

Definition 3. *$G(S, S^t)$ is an auxiliary function for $F(S)$ if $G(S, S^t) \geq F(S)$ and $G(S, S) = F(S)$.*

Lemma 4. *If G is an auxiliary function, then F is non-increasing under the updating rule $S^{t+1} = \arg\min_S G(S, S^t)$.*

First, we derive an algorithm under Euclidean distance function. The most difficult part is the updating rule for $\mathbf{C}^{(1)}$, since the objective function involves both $\mathbf{C}^{(1)}$ and $(\mathbf{C}^{(1)})^T$. We propose an auxiliary function for $\mathbf{C}^{(1)}$ in the following theorem.

Lemma 5.

$$\begin{aligned} G(\mathbf{C}^{(1)}, \tilde{\mathbf{C}}^{(1)}) \quad = \quad & \sum_{i,j}(\mathbf{S}^2_{ij} + \sum_{g,h}([\tilde{\mathbf{C}}^{(1)}\mathbf{D}(\tilde{\mathbf{C}}^{(1)})^T]_{ij}\tilde{\mathbf{C}}^{(1)}_{ig}\mathbf{D}_{gh}\tilde{\mathbf{C}}^{(1)}_{jh} \\ & \frac{(\mathbf{C}^{(1)}_{ig})^4}{(\tilde{\mathbf{C}}^{(1)}_{ig})^4} - 2\mathbf{S}_{ij}\tilde{\mathbf{C}}^{(1)}_{ig}\mathbf{D}_{gh}\tilde{\mathbf{C}}^{(1)}_{jh} \\ & (1 + 2\log\mathbf{C}^{(1)}_{ig} - 2\log\tilde{\mathbf{C}}^{(1)}_{ig}))) + \sum_{i,l}(\mathbf{A}^2_{il} \\ & + \sum_g(\frac{1}{2}[\tilde{\mathbf{C}}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T]_{il}\tilde{\mathbf{C}}^{(1)}_{ig}[\mathbf{B}(\mathbf{C}^{(2)})^T]_{gl} \\ & (\frac{(\mathbf{C}^{(1)}_{ig})^4}{(\tilde{\mathbf{C}}^{(1)}_{ig})^4} + 1) - 2\mathbf{A}_{il}\tilde{\mathbf{C}}^{(1)}_{ig}[\mathbf{B}(\mathbf{C}^{(2)})^T]_{gl} \\ & (1 + \log\mathbf{C}^{(1)}_{ig} - \log\tilde{\mathbf{C}}^{(1)}_{ig}))) \end{aligned} \tag{14}$$

is an auxiliary function for

$$F(\mathbf{C}^{(1)}) = ||\mathbf{S} - \mathbf{C}^{(1)}\mathbf{D}(\mathbf{C}^{(1)})^T||^2 + ||A - \mathbf{C}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T||^2. \tag{15}$$

Proof.

$$F(\mathbf{C}^{(1)}) \quad = \quad \sum_{i,j}(\mathbf{S}_{ij} - [\mathbf{C}^{(1)}\mathbf{D}(\mathbf{C}^{(1)})^T]_{ij})^2$$

$$
\begin{aligned}
& + \sum_{i,l} (\mathbf{A}_{il} - [\mathbf{C}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T]_{il})^2 \\
\leq \quad & \sum_{i,j} \sum_{g,h} \frac{\tilde{\mathbf{C}}^{(1)}_{ig}\mathbf{D}_{gh}\tilde{\mathbf{C}}^{(1)}_{jh}}{[\tilde{\mathbf{C}}^{(1)}\mathbf{D}(\tilde{\mathbf{C}}^{(1)})^T]_{ij}} (\mathbf{S}_{ij} - \frac{[\tilde{\mathbf{C}}^{(1)}\mathbf{D}(\tilde{\mathbf{C}}^{(1)})^T]_{ij}}{\tilde{\mathbf{C}}^{(1)}_{ig}\mathbf{D}_{gh}\tilde{\mathbf{C}}^{(1)}_{jh}} \\
& \mathbf{C}^{(1)}_{ig}\mathbf{D}_{gh}\mathbf{C}^{(1)}_{jh})^2 + \sum_{i,l}\sum_{g} \frac{\tilde{\mathbf{C}}^{(1)}_{ig}[\mathbf{B}(\mathbf{C}^{(2)})^T]_{gl}}{[\tilde{\mathbf{C}}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T]_{il}} (\mathbf{A}_{il} - \\
& \frac{[\tilde{\mathbf{C}}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T]_{il}}{\tilde{\mathbf{C}}^{(1)}_{ig}[\mathbf{B}(\mathbf{C}^{(2)})^T]_{gl}} [\mathbf{C}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T]_{il})^2 \\
= \quad & \sum_{i,j} (\mathbf{S}^2_{ij} + \sum_{g,h} ([\tilde{\mathbf{C}}^{(1)}\mathbf{D}(\tilde{\mathbf{C}}^{(1)})^T]_{ij}\tilde{\mathbf{C}}^{(1)}_{ig}\mathbf{D}_{gh}\tilde{\mathbf{C}}^{(1)}_{jh} \\
& \frac{(\mathbf{C}^{(1)}_{ig})^2(\mathbf{C}^{(1)}_{jh})^2}{(\tilde{\mathbf{C}}^{(1)}_{ig})^2(\tilde{\mathbf{C}}^{(1)}_{jh})^2} - 2\mathbf{S}_{ij}\mathbf{C}^{(1)}_{ig}\mathbf{D}_{gh}\mathbf{C}^{(1)}_{jh})) + \sum_{i,l} (\mathbf{A}^2_{il} \\
& + \sum_{g} ([\tilde{\mathbf{C}}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T]_{il}\tilde{\mathbf{C}}^{(1)}_{ig}[\mathbf{B}(\mathbf{C}^{(2)})^T]_{gl} \\
& \frac{(\mathbf{C}^{(2)}_{ig})^2}{(\tilde{\mathbf{C}}^{(2)}_{ig})^2} - 2\mathbf{A}_{il}\mathbf{C}^{(1)}_{ig}[\mathbf{B}(\mathbf{C}^{(2)})^T]_{gl})) \\
\leq \quad & \sum_{i,j} (\mathbf{S}^2_{ij} + \sum_{g,h} (\frac{1}{2}[\tilde{\mathbf{C}}^{(1)}\mathbf{D}(\tilde{\mathbf{C}}^{(1)})^T]_{ij}\tilde{\mathbf{C}}^{(1)}_{ig}\mathbf{D}_{gh}\tilde{\mathbf{C}}^{(1)}_{jh} \\
& (\frac{(\mathbf{C}^{(1)}_{ig})^4}{(\tilde{\mathbf{C}}^{(1)}_{ig})^4} + \frac{(\mathbf{C}^{(1)}_{jh})^4}{(\tilde{\mathbf{C}}^{(1)}_{jh})^4}) - 2\mathbf{S}_{ij}\tilde{\mathbf{C}}^{(1)}_{ig}\mathbf{D}_{gh}\tilde{\mathbf{C}}^{(1)}_{jh}(1 + \log \mathbf{C}^{(1)}_{ig} \\
& + \log \mathbf{C}^{(1)}_{jh} - \log \tilde{\mathbf{C}}^{(1)}_{ig} - \log \tilde{\mathbf{C}}^{(1)}_{jh}))) + \sum_{i,l} (\mathbf{A}^2_{il} + \\
& \sum_{g} (\frac{1}{2}[\tilde{\mathbf{C}}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T]_{il}\tilde{\mathbf{C}}^{(1)}_{ig}[\mathbf{B}(\mathbf{C}^{(2)})^T]_{gl}(\frac{(\mathbf{C}^{(2)}_{ig})^4}{(\tilde{\mathbf{C}}^{(2)}_{ig})^4} + 1) \\
& -2\mathbf{A}_{il}\tilde{\mathbf{C}}^{(1)}_{ig}[\mathbf{B}(\mathbf{C}^{(2)})^T]_{gl}(1 + \log \mathbf{C}^{(1)}_{ig} - \log \tilde{\mathbf{C}}^{(1)}_{ig}))) \\
= \quad & G(\mathbf{C}^{(1)}, \tilde{\mathbf{C}}^{(1)})
\end{aligned}
$$

During the above deduction, we uses Jensen's inequality, convexity of the quadratic function and inequalities, $x^2 + y^2 \geq 2xy$ and $x \geq 1 + \log x$. □

The following theorem provides the updating rule for $\mathbf{C}^{(1)}$.

Theorem 6. *The objective function $F(\mathbf{C}^{(1)})$ in Eq.*(15) *is nonincreasing under the updating rule,*

$$\mathbf{C}^{(1)} = \tilde{\mathbf{C}}^{(1)} \odot (\frac{\mathbf{S}\tilde{\mathbf{C}}^{(1)}\mathbf{D} + \frac{1}{2}\mathbf{A}\mathbf{C}^{(2)}\mathbf{B}^T}{\tilde{\mathbf{C}}^{(1)}\mathbf{D}(\tilde{\mathbf{C}}^{(1)})^T\tilde{\mathbf{C}}^{(1)}\mathbf{D} + \frac{1}{2}\tilde{\mathbf{C}}^{(1)}\mathbf{B}(\mathbf{C}^{(2)})^T\mathbf{C}^{(2)}\mathbf{B}^T})^{\frac{1}{4}}, \tag{16}$$

where $\tilde{\mathbf{C}}^{(1)}$ denotes the solution from the previous iteration, $\odot$ denotes entry-wise product, and the division between two matrices is entry-wise division.

Proof. Solve $\frac{\partial G(\mathbf{C}^{(1)}, \tilde{\mathbf{C}}^{(1)})}{\partial \mathbf{C}^{(1)}_{ig}} = 0$ and write the solution into the matrix form to obtain (16). Then, by Lemma (4), The objective function in (15) is nonincreasing under this updating rule. □

Similarly, by designing appropriate auxiliary functions (details are omitted due to the space limit), we obtain the updating rules for $\mathbf{C}^{(2)}$, $\mathbf{D}$ and $\mathbf{B}$,

$$\mathbf{C}^{(2)} = \tilde{\mathbf{C}}^{(2)} \odot (\frac{A^T\mathbf{C}^{(1)}\mathbf{B}}{\tilde{\mathbf{C}}^{(2)}\mathbf{B}^T(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}\mathbf{B}}) \tag{17}$$

$$\mathbf{D} = \tilde{\mathbf{D}} \odot (\frac{(\mathbf{C}^{(1)})^T\mathbf{S}\mathbf{C}^{(1)}}{(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}\tilde{\mathbf{D}}(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}}) \tag{18}$$

$$\mathbf{B} = \tilde{\mathbf{B}} \odot (\frac{(\mathbf{C}^{(1)})^T\mathbf{A}\mathbf{C}^{(2)}}{(\mathbf{C}^{(1)})^T\mathbf{C}^{(1)}\tilde{\mathbf{B}}(\mathbf{C}^{(2)})^T\mathbf{C}^{(2)}}) \tag{19}$$

Unfortunately, unlike the hard clustering, the above updating rules do not hold true for all Bregman divergences. For different divergences, we need to design different auxiliary functions to derive updating rules.

A Unified View to Graph Partitioning

The traditional graphs of single-type nodes can be viewed as a special case of complex graphs, i.e., we have only one graph affinity matrix $\mathbf{S}$. When applied to this special case, the proposed framework actually provides a family of new graph partition algorithms with both hard and soft versions.

Existing graph partitioning approaches are mainly based on the edge cut objectives, such as ratio association (Shi and Malik 2000), ratio cut (Chan, Schlag, and Zien 1993), Kernighan-Lin objective (Kernighan and Lin 1970), and normalized cut (Shi and Malik 2000), which can be formulated as the following trace maximization (Dhillon, Guan, and Kulis 2004), i.e., $\max \text{tr}(\mathbf{C}^T S\mathbf{C})$, Typically, $\mathbf{C}$ is formulated as a weighted indicator matrix such that $\mathbf{C}^T\mathbf{C} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix.

We propose the following theorem to show that the edge cut objectives are mathematically equivalent to a special case of the proposed framework. To be consistent with the tradition in edge cut objects, in the following theorem, we assume that $\mathbf{C}$ is a weighted indicator matrix such that $\mathbf{C}^T\mathbf{C} = \mathbf{I}$.

Theorem 7. *The minimization*

$$\min_{\mathbf{C}^T\mathbf{C}=\mathbf{I}, \mathbf{D}=r\mathbf{I}} ||\mathbf{S} - \mathbf{C}\mathbf{D}\mathbf{C}^T||^2 \tag{20}$$

is equivalent to the maximization

$$\max_{\mathbf{C}^T\mathbf{C}=\mathbf{I}} tr(\mathbf{C}^T\mathbf{S}\mathbf{C}), \tag{21}$$

where r is a positive constant and tr denotes the trace of a matrix.

Proof. Let L denote the objective function in Eq. 20.

$$
\begin{aligned}
L &= ||\mathbf{S} - \mathbf{C}(r\mathbf{I})\mathbf{C}^T||^2 \\
&= \text{tr}(\mathbf{S}^T\mathbf{S}) - 2r\text{tr}(\mathbf{C}\mathbf{C}^T\mathbf{S}) + r^2\text{tr}(\mathbf{C}\mathbf{C}^T\mathbf{C}\mathbf{C}^T) \\
&= \text{tr}(\mathbf{S}^T\mathbf{S}) - 2r\text{tr}(\mathbf{C}^T\mathbf{S}\mathbf{C}) + r^2 k
\end{aligned}
$$

Since $\text{tr}(\mathbf{S}^T\mathbf{S})$, r and k are constants, the minimization of L is equivalent to the maximization of $\text{tr}(\mathbf{C}^T\mathbf{S}\mathbf{C})$. The proof is completed. □

Theorem 7 states that the traditional edge cut objectives are equivalent to a special case of the complex graph clustering, clustering on a homogeneous graph under Euclidean distance with cluster pattern matrix $\mathbf{D}$ restricted to be of form $r\mathbf{I}$, i.e., a diagonal matrix. Based on this connection, edge cut objectives make two implicit assumptions for a graph. First, Euclidean distance in Theorem 7 implies normal distribution assumption for the edge weights in a graph. Second, the diagonal constraint on $\mathbf{D}$ assumes that a cluster of nodes always forms a strongly intra-connected subgraph.

These two assumptions are not always true in real applications. The distribution of edge weights could deviate from normal distribution; a cluster of nodes could weakly intra-connected but have the same link pattern, i.e., they all are strongly linked to another set of nodes (Long et al. 2007). On the other hand, the proposed framework and algorithms are flexible to avoid these assumptions when necessary.

Related work

Traditional graph clustering mainly focus on homogeneous graphs of single-type nodes. Graph partitioning divides a graph into subgraphs by finding the best edge cuts of the graph. Several edge cut objectives, such as the average cut (Chan, Schlag, and Zien 1993), average association (Shi and Malik 2000), normalized cut (Shi and Malik 2000), and min-max cut (Ding et al. 2001), have been proposed. Various spectral algorithms have been developed for these objective functions (Chan, Schlag, and Zien 1993; Shi and Malik 2000; Ding et al. 2001). Multilevel methods have been used extensively for graph partitioning with the Kernighan-Lin objective, which attempts to minimize the cut in the graph while maintaining equal-sized clusters (Hendrickson and Leland 1995; Karypis and Kumar 1998). (Yu, Yu, and Tresp 2005) proposes the graph-factorization clustering for soft graph partitioning, which seeks to construct a bipartite graph to approximate a given graph. (van Dongen 2000) proposes a graph clustering algorithm based on flow simulation. (Shental et al. 2004) formulates graph clustering as inferences in an undirected graphical model.

Clustering on a special case of a complex graphs, a bi-partite graph consisting of two types of nodes, has been touched in the literature as co-clustering. (Dhillon 2001) proposes a spectral approach on a bi-partite graph. A generalized co-clustering framework is presented by (Banerjee et al. 2004a) wherein any Bregman divergence can be used in the objective function. Another special case of a complex graph is a k-partite graph consisting of multi-type nodes. (Long et al. 2006) proposes a framework of relation summary network to cluster k-partite graphs. (Gao et al. 2005) proposes an algorithm based on semi-definite programming for clustering star-structured k-partite graphs. Note that these approaches focus on special complex graphs with only heterogeneous links and cannot make use of homogeneous link information.

Experiments

We consider the task of learning cluster structures from a collection of linked documents, such as a collection of cited papers or a collection of linked Web pages. The linked documents can be formulated as a complex graph of document nodes and word nodes, in which there are homogeneous links within document nodes and heterogeneous links between document and word nodes.

In this study, we use benchmark document data sets from the 20-newsgroups (Lang 1995), WebACE and TREC (tre), which cover data sets of different sizes, different balances and different levels of difficulties. The documents are pre-processed by removing the stop words. We construct complex graphs as follows. The link weights between document nodes and word nodes are TF-IDF weights; since the explicit link information is not available, the links between two documents are simulated based on Bernoulli distribution such that the documents from the same cluster link to each other with probability 0.2 and the documents form the different clusters link to each other with probability 0.1. Under this construction, both homogeneous and heterogeneous links provide information for the cluster structures. A summary of all the data sets to construct complex graphs used in this paper is shown in Table 1, in which n denotes the number of documents, k denotes the number of true document clusters, and *balance* denotes the size ratio of the smallest clusters to the largest clusters.

Table 1: Summary of data sets for constructing complex graphs.

Name	n	k	Balance	Source
tr11	414	9	0.046	TREC
tr45	690	10	0.0856	TREC
NG1-20	14000	20	1.0	20-newsgroups
k1b	2340	6	0.043	WebACE

We compare our Hard Complex Graph Clustering (HCGC) and Soft Complex Graph Clustering (SCGC) under Euclidean distance with three representative algorithms. The first one is a classic graph partitioning algorithm, Normalized Cut Graph Partitioning (NCGP), which is capable of using homogeneous links; the second is Spectral Bi-partite Graph Clustering (SBGC), which is capable of using heterogeneous links in a bi-partite graph. To our best knowledge, there are no clustering algorithms for general complex graphs in the literature. However, (Neville, Adler, and Jensen 2003) proposes an approach, Spectral Graph Clustering using both Links and Attributes (SGCLA), which is able to use both link and word information by treating word information as attributes. We use SGCLA as the third comparison algorithm.

For the number of clusters k, we simply use the number of the true document clusters. For performance measure, we elect to use the Normalized Mutual Information (NMI) (Strehl and Ghosh 2002) between the resulting cluster labels and the true cluster labels, which is a standard way to measure the cluster quality. The final performance score is the average of ten runs.

Since NCGP makes use of only homogeneous links and SBGC makes use of only heterogeneous links, we actually have three types of graphs in the experiments, complex graphs of both homogeneous and heterogeneous links (the results are reported with suffix ”-all”), homogeneous graphs of only homogeneous links (with suffix ”-ho”); bi-partite complex graphs of only heterogeneous links (with suffix ”-he”). The NMI scores are reported in Table 2. First, we observe that CGC algorithms always provide the best performances when using both homogeneous and heterogeneous links. For example, for the difficult tr23 data, compared with SGCLA, HCGC increases performance about 40%. Second, for the bipartite graphs of only heterogeneous links, CGC

Table 2: NMI scores comparison

Algorithm	tr23	tr45	k1b	NG1-20
NCGP-ho	0.239 ± 0.019	0.528 ± 0.022	0.399 ± 0.009	0.296 ± 0.020
SBGC-he	0.272 ± 0.034	0.539 ± 0.029	0.493 ± 0.016	0.380 ± 0.030
SGCLA-all	0.358 ± 0.006	0.701 ± 0.012	0.554 ± 0.016	0.538 ± 0.011
HCGC-ho	0.321 ± 0.030	0.653 ± 0.033	0.481 ± 0.029	0.331 ± 0.035
SCGC-ho	0.330 ± 0.021	0.642 ± 0.035	0.493 ± 0.019	0.403 ± 0.019
HCGC-he	0.313 ± 0.033	0.563 ± 0.029	0.553 ± 0.017	0.402 ± 0.017
SCGC-he	0.307 ± 0.026	0.570 ± 0.034	0.573 ± 0.021	0.413 ± 0.023
HCGC-all	$\mathbf{0.501 \pm 0.027}$	$\mathbf{0.801 \pm 0.035}$	0.613 ± 0.012	0.540 ± 0.031
SCGC-all	0.490 ± 0.024	0.793 ± 0.028	$\mathbf{0.640 \pm 0.007}$	$\mathbf{0.590 \pm 0.018}$

algorithms perform slightly better than SBGC. This verifies the effectiveness of CGC algorithms on the bi-partite complex graphs. Third, We observe that CGC algorithms perform better than NCGP on the homogeneous graphs of only homogeneous links. This demonstrates the great potential of CGC algorithms as new graph partitioning algorithms. Besides the document clusters, the CGC algorithms also provide the word clusters and the relations between document clusters and word clusters, which are very useful to identify the topics of the document clusters. Finally, it is worth to note that although the Euclidean distance function is most popular, CGC algorithms under other Bregman divergences may be more desirable in specific complex graph clustering applications depending on the statistical properties of the graphs.

Conclusions

We have proposed a general framework and a family of algorithms to cluster different types of nodes in a complex graph by using both homogeneous and heterogeneous links. Experiments show encouraging results. Future work may include automatic learning the number of clusters and a through investigation on more complex graphs from various applications.

Acknowledgement

This work is supported in part by NSF (IIS-0535162) and AFRL (FA8750-05-2-0284). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

Banerjee, A.; Dhillon, I. S.; Ghosh, J.; Merugu, S.; and Modha, D. S. 2004a. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *KDD*, 509–514.

Banerjee, A.; Merugu, S.; Dhillon, I. S.; and Ghosh, J. 2004b. Clustering with bregman divergences. In *SDM*.

Chan, P. K.; Schlag, M. D. F.; and Zien, J. Y. 1993. Spectral k-way ratio-cut partitioning and clustering. In *DAC '93*, 749–754.

Dhillon, I.; Guan, Y.; and Kulis, B. 2004. A unified view of kernel k-means, spectral clustering and graph cuts. Technical Report TR-04-25, University of Texas at Austin.

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

Ding, C. H. Q.; He, X.; Zha, H.; Gu, M.; and Simon, H. D. 2001. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of ICDM 2001*, 107–114.

Gao, B.; Liu, T.-Y.; Zheng, X.; Cheng, Q.-S.; and Ma, W.-Y. 2005. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *KDD '05*, 41–50.

Hendrickson, B., and Leland, R. 1995. A multilevel algorithm for partitioning graphs. In *Supercomputing '95*, 28.

Karypis, G., and Kumar, V. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* 20(1):359–392.

Kernighan, B., and Lin, S. 1970. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal* 49(2):291–307.

Lang, K. 1995. News weeder: Learning to filter netnews. In *ICML*.

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

Long, B.; Wu, X.; Zhang, Z. M.; and Yu, P. S. 2006. Unsupervised learning on k-partite graphs. In *KDD'06*.

Long, B.; Xu, X.; Zhang, Z. M.; and Yu, P. S. 2007. Community learning by graph approximation. In *ICDM*, 232–241.

Neville, J.; Adler, M.; and Jensen, D. 2003. Clustering relational data using attribute and link information. In *Proceedings of the Text Mining and Link Analysis Workshop, IJCAI'03*.

Salakhutdinov, R., and Roweis, S. 2003. Adaptive overrelaxed bound optimization methods. In *ICML'03*.

S.D.Pietra, V.D.Pietera, J. 2001. Duality and auxiliary functions for bregman distances. Technical Report CMU-CS-01-109, Carnegie Mellon University.

Shental, N.; Zomet, A.; Hertz, T.; and Weiss, Y. 2004. Pairwise clustering and graphical models. In Thrun, S.; Saul, L.; and Schölkopf, B., eds., *NIPS'03*. Cambridge, MA: MIT Press.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8):888–905.

Strehl, A., and Ghosh, J. 2002. Cluster ensembles – a knowledge reuse framework for combining partitionings. In *AAAI 2002*, 93–98.

http://trec.nist.gov/.

van Dongen, S. M. 2000. *Graph Clustering by Flow Simulation*. Ph.D. Dissertation, University of Utrecht.

Yu, K.; Yu, S.; and Tresp, V. 2005. Soft clustering on graphs. In *NIPS'05*.