Relational Clustering by Symmetric Convex Coding

Bo Long BLONG1@BINGHAMTON.EDU
Zhongfei (Mark) Zhang ZZHANG@BINGHAMTON.EDU
Computer Science Dept., SUNY Binghamton, Binghamton, NY 13902

Xiaoyun Wu XIAOYUNW@GOOGLE.COM
Google Inc, 1600 Amphitheatre, Mountain View, CA 94043

Philip S. Yu PSYU@US.IBM.COM
IBM Watson Research Center, 19 skyline Drive, Hawthorne, NY 10532

Abstract

Relational data appear frequently in many machine learning applications. Relational data consist of the pairwise relations (similarities or dissimilarities) between each pair of implicit objects, and are usually stored in relation matrices and typically no other knowledge is available. Although relational clustering can be formulated as graph partitioning in some applications, this formulation is not adequate for general relational data. In this paper, we propose a general model for relational clustering based on symmetric convex coding. The model is applicable to all types of relational data and unifies the existing graph partitioning formulation. Under this model, we derive two alternative bound optimization algorithms to solve the symmetric convex coding under two popular distance functions, Euclidean distance and generalized I-divergence. Experimental evaluation and theoretical analysis show the effectiveness and great potential of the proposed model and algorithms.

1. Introduction

Two types of data are used in unsupervised learning, feature and relational data. Feature data are in the form of feature vectors and relational data consist of the pairwise relations (similarities or dissimilarities) between each pair of objects, and are usually stored in relation matrices and typically no other knowledge is available. Although feature data are the most common type of data, relational data have become more and more popular in many machine learning applications, such as web mining, social network analysis, bioinformatics, VLSI design, and task scheduling. Furthermore, the relational data are more general in the sense all the feature data can be transformed into relational data under a certain distance function.

The most popular way to cluster similarity-based relational data is to formulate it as the graph partitioning problem, which has been studied for decades. Graph partitioning seeks to cut a given graph into disjoint subgraphs which correspond to disjoint clusters based on a certain edge cut objective. Recently, graph partitioning with an edge cut objective has been shown to be mathematically equivalent to an appropriate weighted kernel k-means objective function (Dhillon et al., 2004; Dhillon et al., 2005). The assumption behind the graph partitioning formulation is that since the nodes within a cluster are similar to each other, they form a dense subgraph. However, in general this is not true for relational data, i.e., the clusters in relational data are not necessarily *dense* clusters consisting of strongly-related objects.

Figure 1 shows the relational data of four clusters, which are of two different types. In Figure 1, $\mathcal{C}_1 = \{v_1, v_2, v_3, v_4\}$ and $\mathcal{C}_2 = \{v_5, v_6, v_7, v_8\}$ are two traditional dense clusters within which objects are strongly related to each other. However, $\mathcal{C}_3 = \{v_9, v_{10}, v_{11}, v_{12}\}$ and $\mathcal{C}_4 = \{v_{13}, v_{14}, v_{15}, v_{16}\}$ also form two *sparse* clusters, within which the objects are not related to each other, but they are still "similar" to each other in the sense that they are related to the same set of other nodes. In Web mining, this type of cluster could be a group of music "fans" Web pages which share the same taste on the music and are linked to the same set of music Web pages but are not linked to each other (Kumar et al., 1999). Due to the importance of identifying this type of clusters (communities), it has been listed as one of the five algorithmic challenges in Web search engines (Henzinger et al., 2003). Note that the cluster structure of the relation data in Figure 1 cannot be correctly identified by graph partitioning approaches, since

Appearing in *Proceedings of the 24^{th} International Conference on Machine Learning*, Corvallis, OR, 2007.

they look for only dense clusters of strongly related objects by cutting the given graph into subgraphs; similarly, the pure bi-partite graph models cannot correctly identify this type of cluster structures. Note that re-defining the relations between the objects does not solve the problem in this situation, since there exist both dense and sparse clusters.

If the relational data are dissimilarity-based, to apply graph partitioning approaches to them, we need extra efforts on appropriately transforming them into similarity-based data and ensuring that the transformation does not change the cluster structures in the data. Hence, it is desirable for an algorithm to be able to identify the cluster structures no matter which type of relational data is given. This is even more desirable in the situation where the background knowledge about the meaning of the relations is not available, i.e., we are given only a relation matrix and do not know if the relations are similarities or dissimilarities.

In this paper, we propose a general model for relational clustering based on symmetric convex coding of the relation matrix. The proposed model is applicable to the general relational data consisting of only pairwise relations typically without other knowledge; it is capable of learning both dense and sparse clusters at the same time; it unifies the existing graph partition models to provide a generalized theoretical foundation for relational clustering. Under this model, we derive iterative bound optimization algorithms to solve the symmetric convex coding for two important distance functions, Euclidean distance and generalized I-divergence. The algorithms are applicable to general relational data and at the same time they can be easily adapted to learn a specific type of cluster structure. For example, when applied to learning only dense clusters, they provide new efficient algorithms for graph partitioning. The convergence of the algorithms is theoretically guaranteed. Experimental evaluation and theoretical analysis show the effectiveness and great potential of the proposed model and algorithms.

2. Related Work

Graph partitioning (or clustering) is a popular formulation of relational clustering, which divides the nodes of a graph into clusters by finding the best edge cuts of the graph. Several edge cut objectives, such as the average cut (Chan et al., 1993), average association (Shi & Malik, 2000), normalized cut (Shi & Malik, 2000), and min-max cut (Ding et al., 2001), have been proposed. Various spectral algorithms have been developed for these objective functions (Chan et al., 1993; Shi & Malik, 2000; Ding et al., 2001). These algorithms use the eigenvectors of a graph affinity matrix, or a matrix derived from the affinity matrix, to partition the graph.

Multilevel methods have been used extensively for graph partitioning with the Kernighan-Lin objective, which attempt to minimize the cut in the graph while maintaining equal-sized clusters (Bui & Jones, 1993; Hendrickson & Leland, 1995; Karypis & Kumar, 1998).

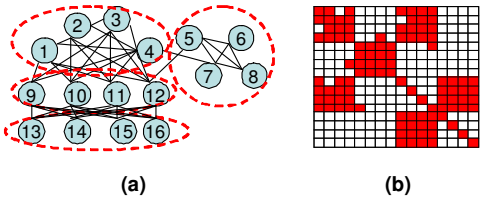

Figure 1. The graph (a) and relation matrix (b) of the relational data with different types of clusters. In (b), the dark color denotes 1 and the light color denotes 0.

Recently, graph partitioning with an edge cut objective has been shown to be mathematically equivalent to an appropriate weighted kernel k-means objective function (Dhillon et al., 2004; Dhillon et al., 2005). Based on this equivalence, the weighted kernel k-means algorithm has been proposed for graph partitioning (Dhillon et al., 2004; Dhillon et al., 2005). Yu et al. (2005) propose the graph-factorization clustering for the graph partitioning, which seeks to construct a bipartite graph to approximate a given graph. Nasraoui et al. (1999) propose the relational fuzzy maximal density estimator algorithm.

In this paper, our focus is on the homogeneous relational data, i.e., the objects in the data are of the same type. There are some efforts in the literature that can be considered as clustering heterogeneous relational data, i.e., different types of objects are related to each other. For example, co-clustering addresses clustering two types of related objects, such as documents and words, at the same time. Dhillon et al. (2003) propose a co-clustering algorithm to maximize the mutual information. A more generalized co-clustering framework is presented by Banerjee et al. (2004) wherein any Bregman divergence can be used in the objective function. Long et al. (2005), Li (2005) and Ding et al. (2006) all model the co-clustering as an optimization problem involving a triple matrix factorization.

3. Symmetric Convex Coding

In this section, we propose a general model for relational clustering. Let us first consider the relational data in Figure 1. An interesting observation is that although the different types of clusters look so different in the graph from Figure 1(a), they all demonstrate block patterns in the relation matrix of Figure 1(b) (without loss of generality, we arrange the objects from the same cluster together to make the block patterns explicit). Motivated by this observation, we propose the Symmetric Convex Coding (SCC) model to cluster relational data by learning the block pattern of a relation matrix. Since in most applications, the relations are of non-negative values and undirected, relational data can be represented as non-negative, symmetric matrices. Therefore, the definition of the SCC is given as follows.

Definition 3.1. Given a symmetric matrix $A \in \mathbb{R}_+$, a distance function $\mathfrak{D}$ and a positive number k, the symmetric

convex coding is given by the minimization,

$$\min_{\substack{C\in\mathbb{R}_+^{n\times k}, B\in\mathbb{R}_+^{k\times k}\\ C\mathbf{1}=\mathbf{1}}} \mathfrak{D}(A, CBC^T). \quad (1)$$

According to Definition 3.1, the elements of C are between 0 and 1 and the sum of the elements in each row of C equal to 1. Therefore, SCC seeks to use the convex combination of the *prototype matrix* B to approximate the original relation matrix. The factors from SCC have intuitive interpretations. The factor C is the soft membership matrix such that C_{ij} denotes the weight that the ith object associates with the jth cluster. The factor B is the prototype matrix such that B_{ii} denotes the connectivity within the ith cluster and B_{ij} denotes the connectivity between the ith cluster and the jth cluster.

SCC provides a general model to learn various cluster structures from relational data. Graph partitioning, which focuses on learning dense cluster structure, can be formulated as a special case of the SCC model. We propose the following theorem to show that the various graph partitioning objective functions are mathematically equivalent to a special case of the SCC model. Since most graph partitioning objective functions are based on the hard cluster membership, in the following theorem we modify the constraints on C as $C \in \mathbb{R}_+$ and $C^TC = I_k$ to make C to be the following cluster indicator matrix,

$$C_{ij} = \begin{cases} \frac{1}{|\pi_j|^{\frac{1}{2}}} & \text{if } v_i \in \pi_j \\ 0 & \text{otherwise} \end{cases}$$

where $|\pi_j|$ denotes the number of nodes in the jth cluster.

Theorem 3.2. *The hard version of SCC model under Euclidean distance function and $B = rI_k$ for $r > 0$, i.e.,*

$$\min_{\substack{C\in\mathbb{R}_+^{n\times k}, B\in\mathbb{R}_+^{k\times k}\\ C^TC=I_k}} ||A - C(rI_k)C^T||^2 \quad (2)$$

is equivalent to the maximization

$$\max \textit{tr}(C^TAC), \quad (3)$$

where tr denotes the trace of a matrix.

Proof. Let L denote the objective function in Eq. 2.

$$\begin{aligned} L &= ||A - rCC^T||^2 && (4)\\ &= \text{tr}((A - rCC^T)^T(A - rCC^T)) && (5)\\ &= \text{tr}(A^TA) - 2r\text{tr}(CC^TA) + r^2\text{tr}(CC^TCC^T) && (6)\\ &= \text{tr}(A^TA) - 2r\text{tr}(C^TAC) + r^2k && (7)\end{aligned}$$

The above deduction uses the property of trace $\text{tr}(XY) = \text{tr}(YX)$. Since $\text{tr}(A^TA)$, r and k are constants, the minimization of L is equivalent to the maximization of $\text{tr}(C^TAC)$. The proof is completed. □

Theorem 3.2 states that with the prototype matrix B restricted to be of the form rI_k, SCC under Euclidean distance is reduced to the trace maximization in (3). Since various graph partitioning objectives, such as ratio association (Shi & Malik, 2000), normalized cut (Shi & Malik, 2000), ratio cut (Chan et al., 1993), and Kernighan-Lin objective (Kernighan & Lin, 1970), can be formulated as the trace maximization (Dhillon et al., 2004; Dhillon et al., 2005), Theorem 3.2 establishes the connection between the SCC model and the existing graph partitioning objective functions. Based on this connection, it is clear that the existing graph partitioning models make an implicit assumption for the cluster structure of the relational data, i.e., the clusters are not related to each other (the off-diagonal elements of B are zeroes) and the nodes within clusters are related to each other in the same way (the diagonal elements of B are r). This assumption is consistent with the intuition about the graph partitioning, which seeks to "cut" the graph into k separate subgraphs corresponding to the strongly-related clusters.

With Theorem 3.2 we may put other types of structural constraints on B to derive new graph partitioning models. For example, we fix B as a general diagonal matrix instead of rI_k, i.e, the model fixes the off-diagonal elements of B as zero and learns the diagonal elements of B. This is a more flexible graph partitioning model, since it allows the connectivity within different clusters to be different. More generally, we can use B to restrict the model to learn other types of the cluster structures. For example, by fixing diagonal elements of B as zeros, the model focuses on learning only spare clusters (corresponding to bi-partite or k-partite subgraphs), which are important for Web community learning (Kumar et al., 1999; Henzinger et al., 2003). In summary, the prototype matrix B not only provides the intuition for the cluster structure of the data, but also provides a simple way to adapt the model to learn specific types of cluster structures.

4. Algorithm Derivation

In this section, we derive efficient algorithms for the SCC model under two popular distance functions, Euclidean distance and generalized I-divergence.

4.1. Algorithm for SCC under Euclidean Distance

We derive an alternative optimization algorithm for SCC under Euclidean distance, i.e., the algorithm alternatively updates B and C until convergence.

First we fix B to update C. To deal with the constraint $C\mathbf{1} = \mathbf{1}$ efficiently, we transform it to a "soft" constraint by adding a penalty term, $\alpha||C\mathbf{1} - \mathbf{1}||^2$, to the objective function, where α is a positive constant. Therefore, we obtain the following optimization.

$$\min_{C\in\mathbb{R}_+^{n\times k}} ||A - CBC^T||^2 + \alpha||C\mathbf{1} - \mathbf{1}||^2. \quad (8)$$

The objective function in (8) is quartic with respect to C. We derive an efficient updating rule for C based on the bound optimization procedure (Salakhutdinov & Roweis, 2003; D.D.Lee & H.S.Seung, 1999). 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 (D.D.Lee & H.S.Seung, 1999) are quoted as follows.

Definition 4.1. $G(S, S^t)$ is an auxiliary function for $F(S)$ if $G(S, S^t) \geq F(S)$ and $G(S, S) = F(S)$.

Lemma 4.2. *If G is an auxiliary function, then F is non-increasing under the updating rule $S^{t+1} = \arg\min_S G(S, S^t)$.*

We propose an auxiliary function for C in the following theorem.

Lemma 4.3.

$$\begin{aligned} G(C,\tilde{C}) &= \sum_{ij}(A_{ij}+\frac{\alpha}{n}-2\sum_{gh}(A_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}(1+2\log C_{jh} \\ &\quad -2\log\tilde{C}_{jh})+\frac{\alpha}{nk}\tilde{C}_{jh}(1+\log C_{jh}-\log\tilde{C}_{jh}))+ \\ &\quad \sum_{gh}([\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+ \\ &\quad \frac{\alpha}{2nk}[\tilde{C}\mathbf{1}]_j\tilde{C}_{jh}(\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+1))) \end{aligned}$$

is an auxiliary function for

$$F(C) = ||A - CBC^T||^2 + \alpha||C\mathbf{1} - \mathbf{1}||^2. \qquad (9)$$

Proof. For convenience, we let $\beta = \frac{\alpha}{nk}$.

$$\begin{aligned} F(C) &= \sum_{ij}((A_{ij}-\sum_{gh}C_{ig}B_{gh}C_{jh})^2+\beta\sum_{gh}(C_{jh}-1)^2) \\ &\leq \sum_{ij}(\sum_{gh}\frac{\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}}{[\tilde{C}B\tilde{C}^T]_{ij}}(A_{ij}-\frac{[\tilde{C}B\tilde{C}^T]_{ij}}{\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}}C_{ig}B_{gh}C_{jh})^2 \\ &\quad +\beta\sum_{gh}\frac{\tilde{C}_{jh}}{[\tilde{C}\mathbf{1}]_j}(\frac{[\tilde{C}\mathbf{1}]_j}{\tilde{C}_{jh}}C_{jh}-1)^2) \\ &= \sum_{ij}(A_{ij}-2\sum_{gh}A_{ij}C_{ig}B_{gh}C_{jh}+ \\ &\quad \sum_{gh}\frac{[\tilde{C}B\tilde{C}^T]_{ij}}{\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}}C_{ig}^2B_{gh}^2C_{jh}^2+\beta\sum_{gh}\frac{[\tilde{C}\mathbf{1}]_j}{\tilde{C}_{jh}}C_{jh}^2 \\ &\quad -2\beta\sum_{gh}C_{jh}+k\beta) \\ &= \sum_{ij}(A_{ij}+k\beta-2\sum_{gh}(A_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}\frac{C_{ig}C_{jh}}{\tilde{C}_{ig}\tilde{C}_{jh}}+ \\ &\quad \beta\tilde{C}_{jh}\frac{C_{jh}}{\tilde{C}_{jh}})+\sum_{gh}([\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}\frac{C_{ig}^2C_{jh}^2}{\tilde{C}_{ig}^2\tilde{C}_{jh}^2} \\ &\quad +\beta[\tilde{C}\mathbf{1}]_j\tilde{C}_{jh}\frac{C_{jh}^2}{\tilde{C}_{jh}^2})) \\ &\leq \sum_{ij}(A_{ij}+k\beta-2\sum_{gh}(A_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}(1+\log C_{ig} \\ &\quad +\log C_{jh}-\log\tilde{C}_{ig}-\log\tilde{C}_{jh})+\beta\tilde{C}_{jh}(1+\log C_{jh}- \\ &\quad \log\tilde{C}_{jh}))+\sum_{gh}(\frac{1}{2}[\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}(\frac{C_{ig}^4}{\tilde{C}_{ig}^4}+\frac{C_{jh}^4}{\tilde{C}_{jh}^4}) \\ &\quad +\frac{1}{2}\beta[\tilde{C}\mathbf{1}]_j\tilde{C}_{jh}(\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+1))) \\ &= \sum_{ij}(A_{ij}+k\beta-2\sum_{gh}(A_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}(1+2\log C_{jh} \\ &\quad -2\log\tilde{C}_{jh})+\beta\tilde{C}_{jh}(1+\log C_{jh}-\log\tilde{C}_{jh}))+ \\ &\quad \sum_{gh}([\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\tilde{C}_{jh}\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+ \\ &\quad \frac{1}{2}\beta[\tilde{C}\mathbf{1}]_j\tilde{C}_{jh}(\frac{C_{jh}^4}{\tilde{C}_{jh}^4}+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 C.

Theorem 4.4. *The objective function $F(C)$ in Eq.*(9) *is nonincreasing under the updating rule,*

$$C = \tilde{C} \odot (\frac{A\tilde{C}B+\frac{\alpha}{2}}{\tilde{C}B\tilde{C}^T\tilde{C}B+\frac{\alpha}{2}\tilde{C}E})^{\frac{1}{4}} \qquad (10)$$

where $\tilde{C}$ denotes the solution from the previous iteration, E denotes a $k \times k$ matrix of 1's, $\odot$ denotes entry-wise product, and the division between two matrices is entry-wise division.

Proof. Based on Lemma 4.3, take the derivative of $G(C, \tilde{C})$ w.r.t. C_{jh} to obtain

$$\begin{aligned} \frac{\partial G(C,\tilde{C})}{\partial C_{jh}} &= \sum_i\sum_{gh}(-4A_{ij}\tilde{C}_{ig}B_{gh}\frac{\tilde{C}_{jh}}{C_{jh}}-2\frac{\alpha}{nk}\frac{\tilde{C}_{jh}}{C_{jh}} \\ &\quad +4[\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}\frac{C_{jh}^3}{\tilde{C}_{jh}^3} \\ &\quad +2\frac{\alpha}{nk}[\tilde{C}\mathbf{1}]_j\frac{C_{jh}^3}{\tilde{C}_{jh}^3}). \end{aligned}$$

Solve $\frac{\partial G(C,\tilde{C})}{\partial C_{jh}} = 0$ to obtain

$$C_{jh} = \tilde{C}_{jh}(\frac{\sum_i\sum_{gh}A_{ij}\tilde{C}_{ig}B_{gh}+\frac{\alpha}{2}}{\sum_i\sum_{gh}[\tilde{C}B\tilde{C}^T]_{ij}\tilde{C}_{ig}B_{gh}+\frac{\alpha}{2}[\tilde{C}\mathbf{1}]_j})^{\frac{1}{4}}$$

Formulate the above equation into the matrix form

$$C = \tilde{C} \odot (\frac{A\tilde{C}B+\frac{\alpha}{2}}{\tilde{C}B\tilde{C}^T\tilde{C}B+\frac{\alpha}{2}\tilde{C}E})^{\frac{1}{4}}$$

By Lemma 4.2, the proof is completed. □

Similarly, we present the following theorems to derive the updating rule for B.

Algorithm 1 SCC-ED algorithm

Input: A graph affinity matrix A and a positive integer k.
Output: A community membership matrix C and a community structure matrix B.
Method:

1: Initialize B and C.
2: **repeat**
3:
$$B = B \odot \frac{C^T AC}{C^T CBC^T C}.$$
4:
$$C = C \odot (\frac{ACB + \frac{\alpha}{2}}{CBC^T CB + \frac{\alpha}{2} CE})^{\frac{1}{4}}$$
5: **until** convergence

Lemma 4.5.

$$G(B, \tilde{B}) = \sum_{ij}(A_{ij} - 2\sum_{gh} A_{ij} C_{ig} B_{gh} C_{jh} + \sum_{gh} [C\tilde{B}C]_{ij} C_{ig} C_{jh} \frac{B_{gh}^2}{\tilde{B}_{gh}})$$

is an auxiliary function for

$$F(B) = ||A - CBC^T||^2. \quad (11)$$

Theorem 4.6. *The objective function $F(B)$ in Eq.*(11) *is nonincreasing under the updating rule*

$$B = \tilde{B} \odot \frac{C^T AC}{C^T C\tilde{B}C^T C}. \quad (12)$$

Following the way to prove Lemma 4.3 and Theorem 4.4, it is not difficult to prove the above theorems. We omit details here.

We call the algorithm as the SCC-ED algorithm, which is summarized in Algorithm 1. The implementation of SCC-ED is simple and it is easy to take advantage of the distributed computation for a very large data set. The complexity of the algorithm is $O(tn^2k)$ for t iterations and it can be further reduced for sparse data. The convergence of the SCC-ED algorithm is guaranteed by Theorems 4.4 and 4.6.

If the task is to learn the dense clusters from similarity-based relational data as the graph partitioning does, SCC-ED can achieve this task simply by fixing B as the identity matrix and updating only C by (10) until convergence. In other words, updating rule (10) itself provides a new and efficient graph partitioning algorithm, which is computationally more efficient than the popular spectral graph partitioning approaches which involve expensive eigenvector computation (typically $O(n^3)$) and the extra post-processing (Yu & Shi, 2003) on eigenvectors to obtain the clustering. Compared with the multi-level approaches such as METIS (Karypis & Kumar, 1998), this new algorithm does not restrict clusters to have an equal size.

Another advantage of the SCC-ED algorithm is that it is very easy for the algorithm to incorporate constraints on B to learn a specific type of cluster structures. For example, if the task is to learn the sparse clusters by constraining the diagonal elements of B to be zero, we can enforce this constraint simply by initializing the diagonal elements of B as zeros. Then, the algorithm automatically only updates the off-diagonal elements of B and the diagonal elements of B are 'locked' to zeros.

Yet another interesting observation about SCC-ED is that if we set $\alpha = 0$ to change the updating rule for C into the following,

$$C = \tilde{C} \odot (\frac{A\tilde{C}B}{\tilde{C}B\tilde{C}^T\tilde{C}B})^{\frac{1}{4}}, \quad (13)$$

the algorithm actually provides the symmetric conic coding. This has been touched in the literature as the symmetric case of non-negative factorizaion (Catral et al., 2004; Ding et al., 2005; Long et al., 2005). Therefore, SCC-ED under $\alpha = 0$ also provides a theoretically sound solution to the symmetric nonnegative matrix factorization.

4.2. Algorithm for SCC under Generalized I-divergence

Under the generalized I-divergence, the SCC objective function is given as follows,

$$D(A||CBC^T) = \sum_{ij}(A_{ij} \log \frac{A_{ij}}{[CBC^T]_{ij}} - A_{ij} + [CBC^T]_{ij}) \quad (14)$$

Similarly, we derive an alternative bound optimization algorithm for this objective function. First, we derive the updating rule for C and our task is the following optimization.

$$\min_{C \in \mathbb{R}_+^{n \times k}} D(A||CBC^T) + \alpha||C\mathbf{1} - \mathbf{1}||^2. \quad (15)$$

Then, the following theorems provide the updating rule for C.

Lemma 4.7.

$$G(C, \tilde{C}) = \sum_{ij}(A_{ij} \log A_{ij} - A_{ij} + \frac{\alpha}{n} + A_{ij} \sum_{gh} (\frac{\tilde{C}_{ig} B_{gh} \tilde{C}_{jh}}{[\tilde{C}B\tilde{C}^T]_{ij}} \log \frac{\tilde{C}_{ig}\tilde{C}_{jh}}{[\tilde{C}B\tilde{C}^T]_{ij}}) + \sum_{gh}((\tilde{C}_{ig} B_{gh} \tilde{C}_{jh} + \frac{\alpha}{nk}[\tilde{C}\mathbf{1}]_j \tilde{C}_{jh}) \frac{C_{jh}^2}{\tilde{C}_{jh}^2}) - 2\sum_{gh}((A_{ij} \frac{\tilde{C}_{ig} B_{gh} \tilde{C}_{jh}}{[\tilde{C}B\tilde{C}^T]_{ij}} + \frac{\alpha}{nk}\tilde{C}_{jh}) \log C_{jh})$$

$$-2\sum_{gh}\frac{\alpha}{nk}\tilde{C}_{jh}(1-\log\tilde{C}_{jh}))$$

is an auxiliary function for

$$F(C)=D(A||CBC^T)+\alpha||C\mathbf{1}-\mathbf{1}||^2. \quad (16)$$

Theorem 4.8. *The objective function $F(C)$ in Eq.*(16) *is nonincreasing under the updating rule,*

$$C_{jh} = \tilde{C}_{jh}\left(\frac{\sum_i \frac{A_{ij}[\tilde{C}B]_{ih}}{[\tilde{C}B\tilde{C}^T]_{ij}}+\alpha}{\sum_i[\tilde{C}B]_{ih}+\alpha[\tilde{C}\mathbf{1}]_j}\right)^{\frac{1}{2}} \quad (17)$$

where $\tilde{C}$ denotes the solution from the previous iteration.

The following theorems provide the updating rule for B.

Lemma 4.9.

$$\begin{aligned}G(B,\tilde{B}) = &\sum_{ij}(A_{ij}\log A_{ij}-A_{ij}+\sum_{gh}C_{ig}B_{gh}C_{jh}\\ &-A_{ij}\sum_{gh}\left(\frac{C_{ig}\tilde{B}_{gh}C_{jh}}{[C\tilde{B}C^T]_{ij}}(\log C_{ig}B_{gh}C_{jh}\right.\\ &\left.-\log\frac{C_{ig}\tilde{B}_{gh}C_{jh}}{[C\tilde{B}C^T]_{ij}}\right)))\end{aligned}$$

is an auxiliary function for

$$F(B)=D(A||CBC^T). \quad (18)$$

Theorem 4.10. *The objective function $F(B)$ in Eq.*(18) *is nonincreasing under the updating rule,*

$$B_{gh} = \tilde{B}_{gh}\frac{\sum_{ij}\frac{A_{ij}C_{ig}C_{jh}}{[C\tilde{B}C^T]_{ij}}}{\sum_{ij}C_{ig}C_{jh}} \quad (19)$$

where $\tilde{B}$ denotes the solution from the previous iteration.

Due to the space limit, we omit the proofs for the above theorems. We call the algorithm based on updating rule (17) and (19) as SCC-GI, which provides another new relational clustering algorithm. Similarly, when applied to the similarity-based relational data of dense clusters, SCC-GI provides another new and efficient graph partitioning algorithm.

5. Experimental Results

This section provides empirical evidence to show the effectiveness of the SCC model and algorithms in comparison with two representative graph partitioning algorithms, a spectral approach, Normalized Cut (NC) (Shi & Malik, 2000), and a multilevel algorithm, METIS (Karypis & Kumar, 1998).

Table 1. Summary of the synthetic relational data

Graph	Parameter	n	k
syn1	$\begin{bmatrix}0.5 & 0 & 0\\ 0 & 0.5 & 0\\ 0 & 0 & 0.5\end{bmatrix}$	900	3
syn2	$1-syn1$	900	3
syn3	$\begin{bmatrix}0 & 0.1 & 0.1\\ 0.1 & 0 & 0.2\\ 0.1 & 0.2 & 0\end{bmatrix}$	900	3
syn4	$[0,1]^{10\times 10}$	5000	10

5.1. Data Sets and Parameter Setting

The data sets used in the experiments include synthetic data sets with various cluster structures and real data sets based on various text data from the 20-newsgroups (Lang, 1995), WebACE and TREC (Karypis, 2002).

First, we use synthetic binary relational data to simulate relational data with different types of clusters such as dense clusters, sparse clusters and mixed clusters. All the synthetic relational data are generated based on Bernoulli distribution. The distribution parameters to generate the graphs are listed in the second column of Table 1 as matrices (true prototype matrices for the data). In a parameter matrix P, P_{ij} denotes the probability that the nodes in the ith cluster are connected to the nodes in the jth cluster. For example, in data syn3, the nodes in cluster 2 are connected to the nodes in cluster 3 with probability 0.2 and the nodes within a cluster are connected to each other with probability 0. Syn2 is generated by using 1 minus syn1. Hence, syn1 and syn2 can be viewed as a pair of similarity/dissimilarity data. Data syn4 has ten clusters mixing with dense clusters and sparse clusters. Due to the space limit, its distribution parameters are omitted here. Totally syn4 has 5000 nodes and about 2.1 million edges.

The graphs based on the text data have been widely used to test graph partitioning algorithms (Ding et al., 2001; Dhillon, 2001; Zha et al., 2001). Note that there also exist feature-based algorithms to directly cluster documents based on word features. However, in this study our focus is clustering based on relations instead of features. Hence graph clustering algorithms are used as comparisons. We use various data sets from the 20-newsgroups (Lang, 1995), WebACE and TREC (Karypis, 2002), which cover data sets of different sizes, different balances and different levels of difficulties. We construct relational data for each text data set such that objects (documents) are related to each other with cosine similarities between the term-frequency vectors. A summary of all the data sets to construct relational data used in this paper is shown in Table 2, in which n denotes the number of objects in the relational data, k denotes the number of true clusters, and *balance* denotes the size ratio of the smallest clusters to the largest clusters.

For the number of clusters k, we simply use the number of the true clusters. Note that how to choose the optimal number of clusters is a nontrivial model selection problem and beyond the scope of this paper. For performance measure,

Table 2. Summary of relational data based on text data sets.

Name	n	k	Balance	Source
tr11	414	9	0.046	TREC
tr23	204	6	0.066	TREC
NG17-19	1600	3	0.5	20-newsgroups
NG1-20	14000	20	1.0	20-newsgroups
k1b	2340	6	0.043	WebACE
hitech	2301	6	0.192	TREC
classic3	3893	3	0.708	MEDLINE/ CISI/CRANFILD

we elect to use the Normalized Mutual Information (NMI) (Strehl & 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.

5.2. Results and Discussion

Table 3 shows the NMI scores of the four algorithms on synthetic and real relational data. Each NMI score is the average of ten test runs and the standard deviation is also reported. We observe that although there is no single winner on all the data, for most data SCC algorithms perform better than or close to NC and METIS. Especially, SCC-GI provides the best performance on eight of the eleven data sets.

For the synthetic data syn1, almost all the algorithms provide perfect NMI score, since the data are generated with very clear dense cluster structures, which can be seen from the parameter matrix in Table 1. For data syn2, the dissimilarity version of syn1, we use exactly the same set of true cluster labels as that of syn1 to measure the cluster quality; the SCC algorithms still provide almost perfect NMI score; however, the METIS totally fails on syn2, since in syn2 the clusters have the form of sparse clusters, and based on the edge cut objective, METIS looks for only dense clusters. An interesting observation is that the NC algorithm does not totally fail on syn2 and in fact it provides a satisfactory NMI score. This is due to that although the original objective of the NC algorithm focuses on dense clusters (its objective function can be formulated as the trace maximization in Eq. (3)), after relaxing C to an arbitrary orthonormal matrix, what NC actually does is to embed cluster structures into the eigen-space and to discover them by post-processing the eigenvectors. Besides the dense cluster structures, sparse cluster structures could also have a good embedding in the eigen-space under a certain condition.

In data syn3, the relations within clusters are sparser than the relations between clusters, i.e., it also has sparse clusters, but the structure is more subtle than syn2. We observe that NC does not provide a satisfactory performance and METIS totally fails; in the mean time, SCC algorithms identify the cluster structure in syn3 very well. Data syn4 is a large relational data set of ten clusters consisting of four dense clusters and six sparse clusters; we observe that the SCC algorithms perform significantly better than NC and METIS on it, since they can identify both dense clusters and sparse clusters at the same time.

For the real data based on the text data sets, our task is to find dense clusters, which is consistent with the objectives of graph partitioning approaches. Overall, the SCC algorithms perform better than NC and METIS on the real data sets. Especially, SCC-ED provides the best performance in most data sets. The possible reasons for this are discussed as follows. First, the SCC model makes use of any possible block pattern in the relation matrices; on the other hand, the edge-cut based approaches focus on diagonal block patterns. Hence, the SCC model is more robust to heavily overlapping cluster structures. For example, for the difficult NG17-19 dataset, SCC algorithms do not totally fail as NC and METIS do. Second, since the edge weights from different graphs may have very different probabilistic distributions, popular Euclidean distance function, which corresponds to normal distribution assumption, are not always appropriate. By Theorem 3.2, edge-cut based algorithms are based on Euclidean distance. On the other hand, SCC-ED is based on generalized I-divergence corresponding to Poisson distribution assumption, which is more appropriate for graphs based on text data. Note that how to choose distance functions for specific graphs is non-trivial and beyond the scope of this paper. Third, unlike METIS, the SCC algorithms do not restrict clusters to have an equal size and hence they are more robust to unbalanced clusters.

In the experiments, we observe that SCC algorithms performs stably and rarely provides unreasonable solution, though like other algorithms SCC algorithms provide local optima to the NP-hard clustering problem. In the experiments, we also observe that the order of the actual running time for the algorithms is consistent with theoretical analysis in Section 4.1, i.e., METIS$<$SCC$<$NC. For example, in a test run on NG1-20, METIS, SCC-ED, SCC-GI and NC take 8.96, 11.4, 12.1 and 35.8 seconds, respectively. METIS is the best, since it is quasi-linear.

We also run the SCC-ED algorithm on the actor/actress graph based on IMDB movie data set for a case study of social network analysis. We formulate a graph of 20000 nodes, in which each node represents an actors/actresses and the edges denote collaboration between them. The number of the cluster is set to be 200. Although there is no ground truth for the clusters, we observe that the results consist of a large number of interesting and meaningful clusters, such as clusters of actors with a similar style and tight clusters of the actors from a movie or a movie serial. For example, Table 4 shows Community 121 consisting of 21 actors/actresses, which contains the actors/actresses in movies series "The Lord of Rings".

6. Conclusions

In this paper, we propose a general model for relational clustering based on symmetric convex coding of the relation matrix. The proposed model is applicable to the gen-

Table 3. NMI comparisons of NC, METIS, SCC-ED and SCC-GI algorithms

Data	NC	METIS	SCC-ED	SCC-GI
syn1	0.9652 ± 0.031	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
syn2	0.8062 ± 0.52	0.000 ± 0.00	0.9038 ± 0.045	$\mathbf{0.9753 \pm 0.011}$
syn3	0.636 ± 0.152	0.115 ± 0.001	0.915 ± 0.145	$\mathbf{1.000 \pm 0.000}$
syn4	0.611 ± 0.032	0.638 ± 0.001	0.711 ± 0.043	$\mathbf{0.788 \pm 0.041}$
tr11	0.629 ± 0.039	0.557 ± 0.001	0.6391 ± 0.033	$\mathbf{0.661 \pm 0.019}$
tr23	0.276 ± 0.023	0.138 ± 0.004	$\mathbf{0.335 \pm 0.043}$	0.312 ± 0.099
NG17-19	0.002 ± 0.002	0.091 ± 0.004	0.1752 ± 0.156	$\mathbf{0.225 \pm 0.045}$
NG1-20	0.510 ± 0.004	$\mathbf{0.526 \pm 0.001}$	0.5041 ± 0.156	0.519 ± 0.010
k1b	0.546 ± 0.021	0.243 ± 0.000	0.537 ± 0.023	$\mathbf{0.591 \pm 0.022}$
hitech	0.302 ± 0.005	$\mathbf{0.322 \pm 0.001}$	0.319 ± 0.012	0.319 ± 0.018
classic3	0.621 ± 0.029	0.358 ± 0.000	0.642 ± 0.043	$\mathbf{0.822 \pm 0.059}$

Table 4. The members of cluster 121 in the actor graph

Cluster 121
Viggo Mortensen, Sean Bean, Miranda Otto, Ian Holm, Brad Dourif, Cate Blanchett, Ian McKellen ,Liv Tyler , David Wenham , Christopher Lee, John Rhys-Davies , Elijah Wood , Bernard Hill, Sean Astin, Dominic Monaghan, Andy Serkis, Karl Urban , Orlando Bloom , Billy Boyd ,John Noble, Sala Baker

eral relational data with various types of clusters and unifies the existing graph partitioning models. We derive iterative bound optimization algorithms to solve the symmetric convex coding for two important distance functions, Euclidean distance and generalized I-divergence. The algorithms are applicable to general relational data and at the same time they can be easily adapted to learn specific types of cluster structures. The convergence of the algorithms is theoretically guaranteed. Experimental evaluation shows the effectiveness and the great potential of the proposed model and algorithms.

Acknowledgement

We thank the anonymous reviewers for insightful comments. This work is supported in part by NSF (IIS-0535162), AFRL (FA8750-05-2-0284), and AFOSR (FA9550-06-1-0327).

References

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

Bui, T. N., & Jones, C. (1993). A heuristic for reducing fill-in in sparse matrix factorization. *PPSC* (pp. 445–452).

Catral, M., Han, L., Neumann, M., & Plemmons, R. (2004). On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices. *Linear Algebra and Its Application*.

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

D.D.Lee, & H.S.Seung (1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401*, 788–791.

Dhillon, I., Guan, Y., & 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., Guan, Y., & Kulis, B. (2005). A fast kernel-based multilevel algorithm for graph clustering. *KDD '05*.

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

Dhillon, I. S., Mallela, S., & Modha, D. S. (2003). Information-theoretic co-clustering. *KDD'03* (pp. 89–98).

Ding, C., He, X., & Simon, H. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. *SDM'05*.

Ding, C., Li, T., Peng, W., & Park, H. (2006). Orthogonal nonnegative matrix tri-factorizations for clustering. *kdd'06*.

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

Hendrickson, B., & Leland, R. (1995). A multilevel algorithm for partitioning graphs. *Supercomputing '95* (p. 28).

Henzinger, M., Motwani, R., & Silverstein, C. (2003). Challenges in web search engines. *Proc. of the 18th International Joint Conference on Artificial Intelligence* (pp. 1573–1579).

Karypis, G. (2002). A clustering toolkit.

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

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

Kumar, R., Raghavan, P., Rajagopalan, S., & Tomkins, A. (1999). Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999)*, *31*, 1481–1493.

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

Li, T. (2005). A general model for clustering binary data. *KDD'05*.

Long, B., Zhang, Z. M., & Yu, P. S. (2005). Co-clustering by block value decomposition. *KDD'05*.

Nasraoui, O., Krishnapuram, R., & Joshi, A. (1999). Relational clustering based on a new robust estimator with application to web mining. *NAFIPS 99*.

Salakhutdinov, R., & Roweis, S. (2003). Adaptive overrelaxed bound optimization methods. *ICML'03*.

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

Strehl, A., & Ghosh, J. (2002). Cluster ensembles – a knowledge reuse framework for combining partitionings. *AAAI 2002* (pp. 93–98).

Yu, K., Yu, S., & Tresp, V. (2005). Soft clustering on graphs. *NIPS'05*.

Yu, S., & Shi, J. (2003). Multiclass spectral clustering. *ICCV'03*..

Zha, H., Ding, C., Gu, M., He, X., & Simon, H. (2001). Bi-partite graph partitioning and data clustering. *ACM CIKM'01*.