Data Min Knowl Disc (2008) 16:276–312
DOI 10.1007/s10618-008-0091-4

Bipartite isoperimetric graph partitioning for data co-clustering

Manjeet Rege · Ming Dong · Farshad Fotouhi

Received: 12 February 2007 / Accepted: 23 January 2008 / Published online: 9 February 2008
Springer Science+Business Media, LLC 2008

Abstract Data co-clustering refers to the problem of simultaneous clustering of two data types. Typically, the data is stored in a contingency or co-occurrence matrix **C** where rows and columns of the matrix represent the data types to be co-clustered. An entry C_{ij} of the matrix signifies the relation between the data type represented by row i and column j. Co-clustering is the problem of deriving sub-matrices from the larger data matrix by simultaneously clustering rows and columns of the data matrix. In this paper, we present a novel graph theoretic approach to data co-clustering. The two data types are modeled as the two sets of vertices of a weighted bipartite graph. We then propose Isoperimetric Co-clustering Algorithm (ICA)—a new method for partitioning the bipartite graph. ICA requires a simple solution to a sparse system of linear equations instead of the eigenvalue or SVD problem in the popular spectral co-clustering approach. Our theoretical analysis and extensive experiments performed on publicly available datasets demonstrate the advantages of ICA over other approaches in terms of the quality, efficiency and stability in partitioning the bipartite graph.

Keywords Data mining · Clustering · Graph algorithms · Linear systems · Singular value decomposition · Eigenvalues and eigenvectors

1 Introduction

Data clustering is the classification of data objects into different groups (clusters) such that data objects in one group are similar together and dissimilar from another group.

Responsible editor: Charu Aggarwal.

M. Rege (✉) · M. Dong · F. Fotouhi
Department of Computer Science, Wayne State University, Detroit, MI 48202, USA
e-mail: rege@wayne.edu

Typically, homogenous data objects, i.e., data objects having the same data type, are grouped together using some of the well known clustering algorithms (Jain et al. 1999; Duda et al. 2000). However, many of the real world data clustering problems arising in data mining applications are pair-wise heterogenous in nature. Clustering problems of these kinds have two data types that need to be clustered together. For example, in a customer relationship management (CRM) application, it is desirable to co-cluster *customers* and *items purchased* to study items of interest for particular category of customers. Customized product promotion campaigns are then targeted at appropriate prospective customers. Collaborative information filtering applications such as movie recommender systems co-cluster the accumulated movie rating provided by viewers and the movies they have watched. A new viewer submits a movie rating for a movie he/she has liked. Using this information, the viewer is recommended other movies by classifying the rating he/she provided to a *viewer ratings-movies watched* cluster. In some of the biomedical applications, co-clustering is performed on *patient symptoms* and *medical diagnosis* for patients in the database. Computer-aided diagnosis is then achieved for a patient based on symptoms provided. From the above discussion, it is clear that the existence of two pair-wise data types is “hand-in-hand”. In other words, one data type in this scenario induces clustering of the other data type and vice-versa. Hence, applying conventional clustering algorithms separately to each of the data types cannot produce meaningful co-clustering results.

Typically, the data is stored in a contingency or co-occurrence matrix $\mathbf{C}$ where rows and columns of the matrix represent the data types to be co-clustered. An entry C_{ij} of the matrix signifies the relation between the data type represented by row i and column j. Co-clustering is the problem of deriving sub-matrices from the larger data matrix by simultaneously clustering rows and columns of the data matrix. Names such as bi-clustering, bi-dimensional clustering, and block clustering, among others, are often used in the literature to refer to the same problem formulation. In Dhillon et al. (2003), co-clustering is defined by a pair of maps from rows to row-clusters and from columns to column-clusters inducing clustered random variables. Optimal co-clustering is then derived based on the one that leads to the largest mutual information between the clustered random variables. Cai et al. (2005) have applied this algorithm to co-cluster auditory scenes and audio elements for unsupervised content discovery in audio. Long et al. (2005) factorize the data matrix into three components, the row-coefficient matrix, the block value matrix, and the column-coefficient matrix. The coefficients denote the degrees of the rows and columns associated with their clusters and the block value matrix is an explicit and compact representation of the hidden block structure of the data matrix. The minimum Bregman information principle is proposed in Banerjee et al. (2004) as a generalization of the maximum entropy principle. Based on this principle, an algorithm for the Bregman co-clustering problem is developed. George and Merugu (2005) adapted the Bregman co-clustering algorithm to a collaborative filtering framework. The key idea in this work is to simultaneously obtain user and item neighborhoods via co-clustering and generate predictions based on the average ratings of the co-clusters while taking into account the individual biases of the users and items.

A well studied problem of co-clustering in data mining literature has been that of documents and words. The goal is to cluster documents based on the common words that appear in them and to cluster words based on the common documents that they appear in. In Slonim and Tishby (2000), a joint distribution is defined over words and documents to first find word-clusters that capture most of the mutual information about the set of documents, and then find document clusters, that preserve the information about the word clusters. Mandhani et al. (2003) proposed a two-step partitional-agglomerative algorithm to hierarchically co-cluster documents and words. The partitioning step involves the identification of sub-matrices so that the respective row sets partition the row set (i.e., documents) of the original matrix. These sub-matrices form the leaf nodes of the hierarchy subsequently created in the agglomerative step. Oh et al. (2001) proposed a fuzzy co-clustering algorithm that maximizes the co-occurrence of categorical attributes (words) and the individual patterns (documents) in clusters. However, this algorithm poses certain problems when the number of documents or the number of words is very large. An enhanced version of the algorithm was proposed in Kummamuru et al. (2003) to perform fuzzy co-clustering of documents and words when dealing with large text corpora.

In this paper, we approach data co-clustering problem from a graph theoretic point of view. We model the relationship between the two data types in the co-clustering problem using a weighted bipartite graph model. The two data types represent the two kinds of vertices in the bipartite graph. Data co-clustering is achieved by partitioning the bipartite graph. Although, a graph partitioning approach has been adopted before by others, the main contribution of this work lies in a new algorithm that we propose—Isoperimetric Co-clustering Algorithm (ICA) for partitioning the bipartite graph. The proposed methodology heuristically minimizes the ratio of the perimeter of the bipartite graph partition and the area of the partition under an appropriate definition of graph-theoretic area. ICA bears resemblance to the popular spectral heuristical approach for partitioning bipartite graphs—it does not require the coordinate information of the vertices of the graphs and allows us to find partitions of an optimal cardinality instead of a predefined cardinality. However, ICA only requires a simple solution to a sparse system of linear equations instead of the eigenvalue or the singular value decomposition problem in the spectral approach. An application of this algorithm to co-cluster documents and words was first presented in a shortened form as a conference paper in Rege et al. (2006a). In this work, we have incorporated theoretical analysis and additional results that demonstrate the advantages of ICA over spectral approach in terms of the quality, efficiency and stability in partitioning the bipartite graph.

The rest of the paper is organized as follows. Section 2 begins with a short overview on graph theory. Following this, we review related work in the literature. The ICA algorithm for partitioning the bipartite graph is presented in Sect. 3. Theoretical analysis of the proposed algorithm is performed in Sect. 4. Experimental results are presented in Sect. 5 and we conclude in Sect. 6.

2 Related work

In this section, we introduce some essential background on graph theory and review related work in the literature.

2.1 Homogeneous graphs

An undirected homogeneous graph G = {V, E} consists of a set of vertices (nodes) $V = \{v_1, v_2, \ldots, v_{|V|}\}$ and a set of edges $\mathrm{E} = \{e_{ij} |$ edge between v_i and $v_j, i, j <= |V|\}$, where $|V|$ is the number of vertices. In a weighted graph, each edge e_{ij} has a positive weight denoted by $w(e_{ij})$. The weight of the edge signifies the level of association between the vertices. An edge weight of zero denotes the absence of an edge between the two respective vertices. Given a vertex numbering and the edge weights between the vertices, graphs can be represented by matrices. We begin with definitions of a few graph terminologies that play an essential role in the paper.

Definition 1 The adjacency matrix **A** of the graph is defined as,

$$A_{ij} = \begin{cases} w(e_{ij}), & \text{if } e_{ij} \text{ exists} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Definition 2 The degree of a vertex v_i denoted by d_i is defined as,

$$d_i = \sum_{e_{ij}} w(e_{ij}), \quad \forall e_{ij} \in E \tag{2}$$

Definition 3 The degree matrix **D** of the graph is a diagonal matrix as,

$$D_{ij} = \begin{cases} d_i, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Definition 4 The Laplacian matrix **L** of a graph is a symmetric matrix with one row and column for each vertex such that,

$$L_{v_i,v_j} = \begin{cases} d_i, & \text{if } i = j \\ -w(e_{ij}), & \text{if } e_{ij} \text{ exists} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

For a survey on Laplacian matrices of graphs, see Merris (1994). Some of the characteristics of **L** are as follows:

- From the definitions of **L**, **D**, and **A**, we get $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- **L** is symmetric positive semi-definite (Biggs 1974; Fiedler 1986). A graph G is connected iff 0 is a simple eigenvalue of **L** (Mohar 1991). The eigenvector for 0 is $\mathbf{e} = [1, 1, \ldots, 1]^T$.
- If we denote the second smallest eigenvalue of **L** with λ_2 then the following holds (Fiedler 1973),

$$\lambda_2 = \min \frac{\mathbf{x}^T \mathbf{L} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}, \quad \mathbf{x} \perp \mathbf{e} \tag{5}$$

Partitioning of the graph is to choose subsets of the vertex set V such that the sets share a minimal number of spanning edges while satisfying a specified cardinality constraint.

Definition 5 Suppose we bipartition set V into subsets V_1 and V_2, then the corresponding graph *cut* is defined as,

$$cut(V_1, V_2) = \sum_{i \in V_1, j \in V_2} A_{ij} \tag{6}$$

The above definition can be extended to k-partitioning of the graph. The cut in which case is defined as,

$$cut(V_1, V_2, \ldots, V_k) = \sum_{n<\theta} cut(V_n, V_\theta) \tag{7}$$

Graph partitioning methods have been applied in diverse fields such as parallel processing (Simon 1991), VLSI circuit design (Alpert and Kahng 1995), image segmentation (Shi and Malik 2000; Grady and Schwartz 2006a) and data clustering (Zha et al. 2001). A graph partitioning algorithm assigns a set of values to each vertex in the graph. We will refer to a vector consisting of the values for each of the vertices as the *indicator vector* of the graph. The *cutting* of the graph is dividing the indicator vector based on the values associated with each vertex. Of the various graph partitioning methods such as geometric partitioning (Gilbert et al. 1998), inertial partitioning (Hendrickson and Leland 1995), MCL partitioning (Dongen 2000; Enright et al. 2002) or coordinate partitioning proposed in graph theory, spectral partitioning has been the most popular and widely applied. It is based on the early works of Donath and Hoffman (1972, 1973) who proposed using the eigenvectors of adjacency matrices of the graphs to find partitions. Fiedler (1973, 1975a,b) associated λ_2 with the connectivity of the graph and suggested partitioning by dividing vertices according to their value in the corresponding eigenvector $\mathbf{u}_2 = \{u_1, u_2, \ldots, u_{|V|}\}$. Consequently, λ_2 is referred to as the *Fiedler value* and $\mathbf{u}_2$ as the *Fiedler vector*. A splitting value s partitions the vertices of the graph into the set of i such that $u_i > s$ and the set such that $u_i \leq s$.

Shi and Malik applied spectral graph partitioning to the problem of image segmentation in their Normalized Cuts paper (Shi and Malik 2000). The objective function used in this work is,

$$\min \frac{\mathbf{x}^T \mathbf{L} \mathbf{x}}{\mathbf{x}^T \mathbf{D} \mathbf{x}}, \text{ subject to } \mathbf{x}^T \mathbf{D} \mathbf{e} = 0, \mathbf{x} \neq \mathbf{0} \tag{8}$$

In Eq. 8, $\mathbf{x}$ is a column vector such that $x_i = c_1$ if $i \in V_1$ and $x_i = -c_2$ if $i \in V_2$, where c_1 and c_2 are constants derived from the degree matrix $\mathbf{D}$. By relaxing x_i from discrete to continuous, it can be shown that the solution to (8) is the eigenvector corresponding to the second smallest eigenvalue of the generalized eigenvalue problem (Chung 1997; Golub and Van-Loan 1989),

$$\mathbf{L}\mathbf{x} = \lambda \mathbf{D}\,\mathbf{x} \tag{9}$$

Partitions can then be obtained by running a clustering algorithm such as *k*-means (Duda et al. 2000) on the eigenvector **x**.

2.2 Bipartite graphs

An undirected bipartite graph $G = \{M, R, E\}$, has two sets of vertices, viz., M and R and a set of graph edges E. Let $\mathbf{B}$ be an m by n graph weight matrix, where n and m represent the number of vertices in M and R, respectively. An entry B_{ij} in this matrix is the weight of an edge appearing between a vertex $r_i \in R$ and a vertex $m_j \in M$. There are no edges between vertices of the same group (see Fig. 1). Then, the adjacency matrix of the bipartite graph is expressed as,

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \tag{10}$$

where the first m vertices index R and the last n index M.

A degree vector **d** of the bipartite graph is,

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_R \\ \mathbf{d}_M \end{bmatrix} \tag{11}$$

where $\mathbf{d}_R$ and $\mathbf{d}_M$ are vectors consisting of degree of vertices belonging to R and M, respectively. The degree matrix becomes a composite of the degree matrices of the two data types as follows,

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_M \end{bmatrix} \tag{12}$$

where $D_R(i, i) = \sum_j B_{ij}$ and $D_M(j, j) = \sum_i B_{ij}$.

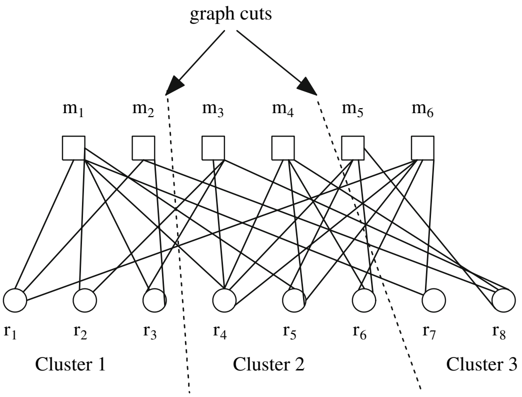


Fig. 1 The square and circular vertices (m and r, respectively) denote the two data types in the co-clustering problem that are represented by the bipartite graph. Partitioning this bipartite graph leads to co-clustering of the two data types

The bipartite Laplacian matrix is defined as,

$$\mathbf{L} = \begin{bmatrix} \mathbf{D}_R & -\mathbf{B} \\ -\mathbf{B}^T & \mathbf{D}_M \end{bmatrix} \tag{13}$$

The two data types in the co-clustering problem can be represented by the two vertices of the weighted bipartite graph. Co-clustering of the data is achieved by partitioning the bipartite graph. In Fig. 1, we show the bipartite graph partitioned using dotted lines. The three partitions obtained are viz., $\{m_1, m_2, r_1, r_2, r_3\}$, $\{m_3, m_4, r_4, r_5, r_6\}$, and $\{m_5, m_6, r_7, r_8\}$. Therefore, the objects in M are clustered into $\{m_1, m_2\}$, $\{m_3, m_4\}$, and $\{m_5, m_6\}$, while those in R are clustered into $\{r_1, r_2, r_3\}$, $\{r_4, r_5, r_6\}$, and $\{r_7, r_8\}$ simultaneously. For bipartite graph partitioning, spectral approach is the only one that has been successfully developed theoretically and widely applied for co-clustering. In order to compute these partitions, we also need to solve a generalized eigenvalue problem as in Eq. 9. However, due to the bipartite nature of the problem, the eigenvalue problem reduces to a much efficient Singular Value Decomposition (SVD) (Golub and Van-Loan 1989) problem. Dhillon (2001) and Zha et al. (2001) employed this Spectral-SVD approach to partition a bipartite graph of documents and words. Ding (2003a) performed document-word co-clustering by extending Hopfield networks (Hopfield 1982) to partition bipartite graphs and showed that the solution is the principal component analysis (PCA) (Jolliffe 2002). In Zha and Ji (2002), the two types of vertices of bipartite graph are used to represent sentences of documents of two different languages. The Spectral-SVD method is applied to identify subgraphs of the weighted bipartite graph which can be considered as corresponding to sentences that correlate well in textual contents.

This algorithm has also found application in multimedia co-clustering problems. In Rege et al. (2006b), it has been used to co-cluster a bipartite graph of user relevance feedback logs and low-level image features. Wu et al. (2005) have used it on a bipartite graph where news stories represent one type of nodes while features (textual and visual) extracted from video keyframes represent the other. In Kumar et al. (2004), two algorithms have been proposed to extract story lines from user search by constructing a word-document bipartite graph. One is a heuristic based iterative local search algorithm while the other is dynamic programming based algorithm to identify dense subgraphs. In Qiu (2004), images and their low-level features were modeled using a bipartite graph and Hopfield model based stochastic algorithm was employed for co-clustering. Bipartite graph partitioning for co-clustering have also been applied in the field of bioinformatics. In Ding (2003b), co-clustering is performed on a genes-tissues bipartite graph using the Spectral-SVD approach. In Ding et al. (2004) a bipartite graph is used to represent protein and protein complex relationship, out of which protein–protein and complex–complex interactions arise naturally. It is shown that co-clustering produces meaningful protein modules and supercomplexes.

In the next section, we derive the proposed algorithm for data co-clustering using weighted bipartite graphs and show that our algorithm requires a simple solution to a sparse system of linear equation to partition the bipartite graph.

3 Isoperimetric co-clustering algorithm (ICA)

In data co-clustering of Fig. 1, clustering of one data type viz., m induces clustering of r and vice-versa. Let us denote the m clusters as $M_1, M_2, \ldots, M_k$ and the clusters of r with $R_1, R_2, \ldots, R_k$. The basic premise of our algorithm is that, if r_i belongs to a cluster, say R_p, where $1 \leq p \leq k$, then its association with M_p is greater than its association with any other M cluster. From a graph theoretic point of view, the association between r_i and the M clusters can be expressed in terms of the sum of the edge weights. Thus,

$$R_p = \left\{ r_i \colon \sum_{j \in M_p} B_{ij} \geq \sum_{j \in M_l} B_{ij}, \forall\, l = 1, \ldots, k \right\} \tag{14}$$

where the matrix **B** is as defined in Eq. 10. Similarly, for m_j belonging to cluster M_p, the following should hold,

$$M_p = \left\{ m_j \colon \sum_{i \in R_p} B_{ij} \geq \sum_{i \in R_l} B_{ij}, \forall\, l = 1, \ldots, k \right\} \tag{15}$$

The algorithm presented here has been motivated from the combinatorial formulation of the classic isoperimetric problem (Dodziuk 1984; Dodziuk and Kendall 1986; Mohar 1989; Grady and Schwartz 2006a,b): *For a fixed area, find the shape with minimum perimeter*. We present a polynomial time heuristic for the NP-hard (Garey and Johnson 1979) problem of finding a region with minimum perimeter for a fixed area.

Definition 6 The *isoperimetric constant* ϕ of a continuous manifold is defined as (Cheeger 1970),

$$\phi = \inf_F \frac{|\triangle F|}{Vol_F}, \tag{16}$$

where F is a region in the manifold, Vol_F is the volume of F, $|\triangle F|$ is the area of the boundary of F, and ϕ is the infimum of the ratio over all possible regions F in the manifold. Also, for a compact manifold, $Vol_F \leq \frac{1}{2} Vol_{Total}$ and for a noncompact manifold $Vol_F < \infty$.

Definition 7 The *isoperimetric number* ϕ_G for a bipartite graph $G = \{M, R, E\}$ is defined as (Mohar 1989),

$$\phi(G) = \inf_F \frac{|\triangle F|}{Vol_F} \tag{17}$$

where F is a subset of the set of vertices $\{M \bigcup R\}$ of the graph and

$$Vol_F \leq \frac{1}{2} Vol_{\{M \bigcup R\}} \tag{18}$$

Since, our bipartite graph has a finite number of vertices, the infimum in Eq. 17 becomes a minimum. The boundary $\triangle F$ of the set F is defined as,

$$\triangle F = \left\{ e_{ij} |\text{ edges between a vertex in F and its complement } F^C \right\}$$

Also, since the bipartite graph is weighted, we define,

$$|\triangle F| = \sum_{e_{ij} \in \triangle F} w(e_{ij}) \tag{19}$$

The combinatorial volume (Dodziuk 1984; Dodziuk and Kendall 1986) can be defined in the following two ways:

$$Vol_F = |F| \tag{20}$$

or,

$$Vol_F = \sum_i d_i, \forall \text{ vertices } \in F \tag{21}$$

where d_i is the degree of the vertex.

The first notion defines the volume in terms of the number of vertices in the bipartite subset F while the second one defines it in terms of the sum of the edge weights incident on each of the vertices enclosed in F. For the data co-clustering problem, it is more appropriate to represent the volume in terms of the sum of the vertex degrees as it utilizes the information from the weights of the edges instead of representing the volume only in terms of the vertex cardinality which might not necessarily be informative. Consequently, for rest of the algorithm derivation we use Eq. 21 to represent volume.

Definition 8 The isoperimetric ratio ϕ_F for the bipartite subset F is defined to be the ratio of boundary area of F to the volume of F.

The isoperimetric sets for a graph G are any sets F and F^C for which $\phi_F = \phi(G)$. The specification of a set satisfying Eq. 18 together with its complement is considered as a partition. Partition with a low isoperimetric ratio is considered to be an optimal partition. An optimal partition consists of the isoperimetric sets themselves. Throughout the paper, the goal of our algorithm is to derive partitions with a low isoperimetric ratio. In other words, we want to maximize the volume Vol_F and minimize the boundary area $|\triangle F|$.

We define two indicator vectors for the two data types as follows,

$$x_i = \begin{cases} 0, & \text{if } r_i \in F \\ 1, & \text{if } r_i \notin F \end{cases} \tag{22}$$

and,

$$y_i = \begin{cases} 0, & \text{if } m_i \in F \\ 1, & \text{if } m_i \notin F \end{cases} \tag{23}$$

For simplicity, we combine the two indicator vectors into a single indicator $\mathbf{z}$ as,

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \tag{24}$$

Binary values assigned to first $|R|$ vertices of $\mathbf{z}$ indicate the partitioning of vertices in R. Partitioning of M is inferred from the next $|M|$ vertices in $\mathbf{z}$.

From Eqs. 13, 19, 22, 23 and 24, we can express $|\triangle F|$ as,

$$\begin{aligned} |\triangle F| &= \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}^T \mathbf{L} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \\ &= \mathbf{z}^T \mathbf{L} \mathbf{z} \end{aligned} \tag{25}$$

Also, using Eqs. 11, 21 and 24, we can write Vol_F as,

$$Vol_F = \mathbf{z}^T \mathbf{d} \tag{26}$$

To achieve the minimum of the ratio $\frac{|\triangle F|}{Vol_F}$, if we assume that the volume is fixed, then it is enough to minimize the numerator subject to the constraint,

$$\mathbf{z}^T \mathbf{d} = c, \tag{27}$$

where $0 < c < \frac{1}{2}\mathbf{r}^T \mathbf{d}$, c is an arbitrary constant and $\mathbf{r}$ is a vector of all ones.

We are now in a position to write the isoperimetric number of the graph as,

$$\phi(G) = \min_{\mathbf{z}} \frac{\mathbf{z}^T \mathbf{L} \mathbf{z}}{\mathbf{z}^T \mathbf{d}} \tag{28}$$

We denote the isoperimetric ratio associated with an indicator vector $\mathbf{z}$ as $\phi(\mathbf{z})$. In order to get optimal partition, we want to derive $\mathbf{z}$ such that it yields the minimum isoperimetric ratio over all values of $\mathbf{z}$.

For optimization, we relax the binary constraint on $\mathbf{z}$ (consequently, on $\mathbf{x}$ and $\mathbf{y}$) to take on real non-negative values and define a cost function $\mathcal{C}$ as,

$$\mathcal{C} = \mathbf{z}^T \mathbf{L} \mathbf{z} - \rho(\mathbf{z}^T \mathbf{d} - c) \tag{29}$$

where ρ is a Lagrange multiplier (Arfken and Weber 2000).

As $\mathbf{L}$ is positive semi-definite and $\mathbf{z}^T \mathbf{d}$ is non-negative, $\mathcal{C}$ will be at minimum at its critical point. Differentiating Eq. 29 with respect to $\mathbf{z}$, we get,

$$\frac{d\mathcal{C}}{d\mathbf{z}} = 2\mathbf{L}\mathbf{z} - \rho\mathbf{d} \tag{30}$$

Equating the above equation to zero and ignoring the constants 2 and ρ since we are not concerned in getting the actual values of **z** but only relative values, we get,

$$\mathbf{L}\mathbf{z} = \mathbf{d}$$
$$\begin{bmatrix} \mathbf{D}_R & -\mathbf{B} \\ -\mathbf{B}^T & \mathbf{D}_M \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_R \\ \mathbf{d}_M \end{bmatrix} \tag{31}$$

Equation 31 is system of linear equations with $|R| + |M|$ number of equations with as many number of variables. Also, the matrix **L** is singular, i.e. its determinant is zero. So, this system of linear equations does not have a unique solution (Golub and Van-Loan 1989).

We convert the system to non-singular system of equations by removing a single vertex from the graph and assign it to be included in F. That is, its indicator value is assigned to be zero. As will be shown in Sect. 3.1, the co-clustering results are not affected by the type of the vertex removed from the bipartite graph. A row and column in **L** and a row each in **z** and **d** are removed corresponding to the removed vertex. We write the new non-singular system of linear equations as,

$$\mathbf{L}^*\mathbf{z}^* = \mathbf{d}^*$$
$$\begin{bmatrix} \mathbf{D}_R & -\mathbf{B} \\ -\mathbf{B}^T & \mathbf{D}_M \end{bmatrix}^* \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}^* = \begin{bmatrix} \mathbf{d}_R \\ \mathbf{d}_M \end{bmatrix}^* \tag{32}$$

Solving Eq. 32 for **x** and **y** results in a real valued solution. In order to get partitions, this solution needs to be *cut* using a *splitting value* (as explained in Sect. 2) to convert it to a binary vector as per Eqs. 22 and 23. Amongst the common methods for cutting the indicator vector are the median cut and the ratio cut (Hagen and Kahng 1992). Median cut uses the median of the indicator vector **z** as the *splitting value* to produce equally sized partitions while ratio cut chooses one such that the resulting partitions have the lowest isoperimetric ratio. As our goal is to produce optimal co-clustering of the two data types and not necessarily equally sized clusters, we employ the ratio cut to get the partitions. To perform k-partitioning, we apply the algorithm recursively until the isoperimetric ratio obtained after every partition fails to meet a pre-determined threshold called the k-parameter. During recursion, it is checked if the partition ratio is less than the k-parameter. If so, the recursion is continued; else it is stopped.

3.1 Vertex removal

In this section, we discuss the vertex removal strategy employed in the algorithm to solve the system of linear equations. For a homogeneous graph, the spectral radius of **L** is $\leq$ twice the maximum degree of the graph (Anderson and Morley 1985) suggesting that removing a vertex with the maximum degree is the appropriate choice for us. However, in a bipartite graph it is not clear if the vertex removed should belong to any particular data type. That is, if we go with the heuristic of removing the maximum degree vertex then should we calculate the maximum across both sets of vertices together or should we remove a maximum degree vertex from within one of the two

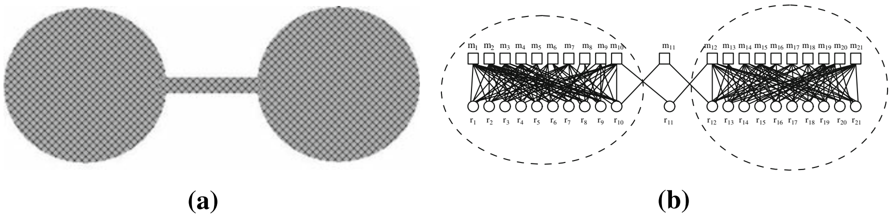

m_1 m_2 m_3 m_4 m_5 m_6 m_7 m_8 m_9 m_{10}
m_{11}
m_{12} m_{13} m_{14} m_{15} m_{16} m_{17} m_{18} m_{19} m_{20} m_{21}
r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10}
r_{11}
r_{12} r_{13} r_{14} r_{15} r_{16} r_{17} r_{18} r_{19} r_{20} r_{21}
(a)
(b)

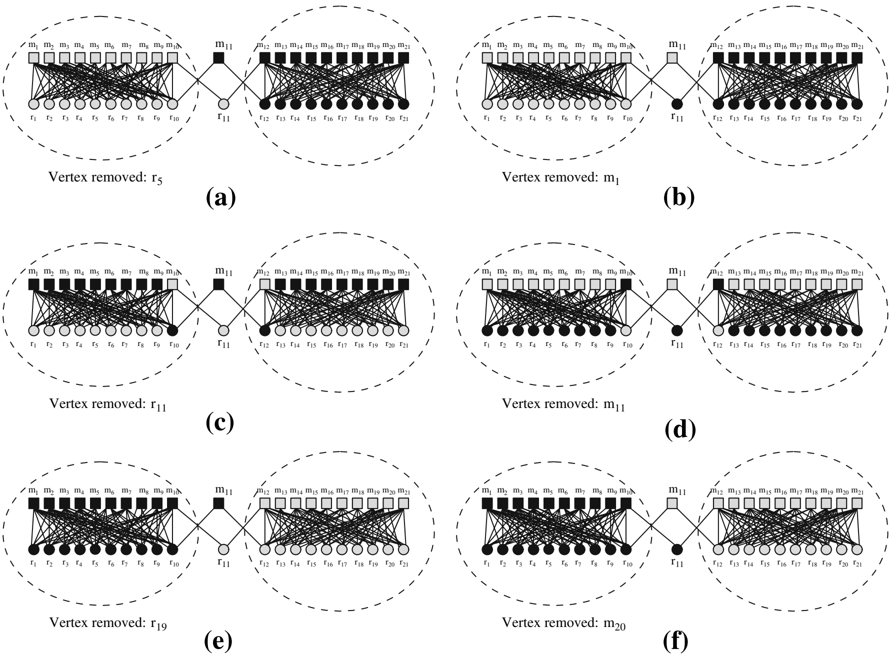
Fig. 2 (**a**) Cheeger's dumbbell-shaped homogeneous graph has vertices in the two lobes densely connected while the two lobes are sparsely connected to each other. (**b**) Bipartite version of the dumbbell graph consisting of the two data types to be co-clustered

vertex sets? We analyze this using the dumbbell shaped graph (Fig. 2a) which was discussed in Cheeger (1970) on the relationship of the isoperimetric constant and the eigenvalues of the Laplacian on continuous manifolds. We constructed a bipartite dumbbell graph (Fig. 2b) with uniform weights and 42 vertices in total with 21 vertices for each of the two data types. In the dumbbell graph, vertices in each of the two lobes are densely connected while the two lobes are sparsely connected to each other. An optimal partitioning for this graph should result in cutting of the graph into the two lobes. Figure 3 shows the partitioning achieved after different vertices were removed from the bipartite dumbbell graph to solve the system of linear equations. We removed vertices that were densely connected (i.e., high degree) from each of the two data types (represented with m and r) and vertices that were sparsely connected (like m_{11} and r_{11}). From the six cases, we observe that it does not matter which vertex data type is removed in the algorithm as long as the vertex is densely connected in the graph (Fig. 3a, b, e, f). However, if a vertex along the ideal cut is removed, that is sparsely connected vertex to the graph (low degree), then the algorithm produces imbalanced partitions (Fig. 3c, d). In the experiments (Sect. 5), we removed vertex with maximum degree, minimum degree and also randomly chose a vertex to be removed. Our results show that ICA produces optimal partitions with low isoperimetric ratio when the removed vertex is the one with maximum degree.

3.2 Algorithm summary

The main steps of ICA can be summarized as follows:

(1) Given the data matrix **B**, construct **L** and **d** using Eqs. 13 and 11, respectively.
(2) Find the vertex with the maximum degree in the bipartite graph, and construct $\mathbf{L}^*$ and $\mathbf{d}^*$ by removing the row and column from **L** and a row from **d** corresponding to the maximum degree vertex.
(3) Solve the system of linear equations in Eq. 32 for the indicator vector **z** defined by Eq. 24.
(4) Bipartition **z** using ratio cut to get two clusters.
(5) If more than two partitions (clusters) are desired, i.e., to perform k-partitioning, apply the algorithm recursively to each partition until the isoperimetric ratio of the sub-partitions is larger than the k-parameter.


m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
m11
m12 m13 m14 m15 m16 m17 m18 m19 m20 m21
r1 r2 r3 r4 r5 r6 r7 r8 r9 r10
r11
r12 r13 r14 r15 r16 r17 r18 r19 r20 r21
(a)
(b)
(c)
(d)
(e)
(f)

Vertex removed: r_5

Vertex removed: m_1

Vertex removed: r_{11}

Vertex removed: m_{11}

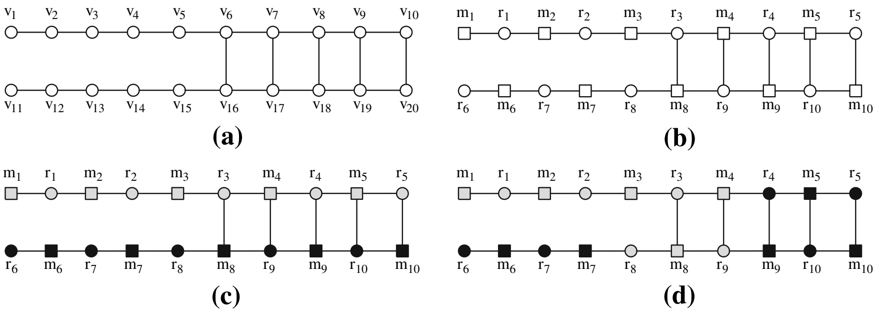
Vertex removed: r_{19}

Vertex removed: m_{20}

Fig. 3 The six cases show the partitioning achieved when different vertices from the bipartite graph are removed to solve the system of linear equations

3.3 Advantages of ICA over spectral approach

Although spectral graph partitioning has been popular and successfully applied to diverse research problems, it does suffer from some significant drawbacks. In Guattery and Miller (1998), families of graphs have been proposed for which spectral partitioning fails to produce the best partition. For example, the *roach* graphs that have an approximate shape of a cockroach consist of two path graphs, each on $2k$ vertices. The “body” section of the graph consists of edges between the upper and lower paths while the “antennae” section has no edges between the two path graphs. These graphs will always be sub-optimally partitioned into two symmetrical halves by the Spectral method (using the median cut) relative to the minimum isoperimetric ratio criterion. A *roach* graph for $k = 5$ is shown in Fig. 4a. We constructed a bipartite roach graph for $k = 5$, shown in Fig. 4b, where the two kinds of vertices are indexed by the alphabets m and r. In Fig. 4c and d, we show the results for bipartitioning the bipartite graph with the Spectral-SVD algorithm and ICA, respectively. We can see that, spectral approach has undesirably partitioned the bipartite graph into the two symmetrical halves with a much higher isoperimetric ratio. These results for bipartite graph are in agreement with the ones demonstrated in Guattery and Miller (1998) for homogeneous graphs. With this example, we have been able to show that ICA is able to perform well on a category of graphs that Spectral methods are not able to partition efficiently.



v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10}
v_{11} v_{12} v_{13} v_{14} v_{15} v_{16} v_{17} v_{18} v_{19} v_{20}
(a)
m_1 r_1 m_2 r_2 m_3 r_3 m_4 r_4 m_5 r_5
r_6 m_6 r_7 m_7 r_8 m_8 r_9 m_9 r_{10} m_{10}
(b)
m_1 r_1 m_2 r_2 m_3 r_3 m_4 r_4 m_5 r_5
r_6 m_6 r_7 m_7 r_8 m_8 r_9 m_9 r_{10} m_{10}
(c)
m_1 r_1 m_2 r_2 m_3 r_3 m_4 r_4 m_5 r_5
r_6 m_6 r_7 m_7 r_8 m_8 r_9 m_9 r_{10} m_{10}
(d)

Fig. 4 *Roach* graphs have been shown to produce partitioning with low isoperimetric ratio. (**a**) Roach graph for $k = 5$. (**b**) Bipartite roach graph for $k = 5$. (**c**) Solution with Spectral-SVD algorithm. Ratio = 0.2174. (**d**) Solution with ICA. Ratio = 0.1304

ICA requires solution to a system of linear equations which in general is computationally efficient over solving an eigenvalue problem or performing SVD as in the spectral approach (Dhillon 2001; Zha et al. 2001). The Lanczos algorithm (Golub and Van-Loan 1989; Demmel 1997) is a popular method to efficiently compute approximations of eigenvalues of large symmetric matrices. However, recently some concerns have been raised about this method in approximating eigenvalues (Kuijlaars 2001). While outliers in the eigenvalue spectrum are approximated well, eigenvalues in the bulk of the spectrum are typically harder to approximate with this method. Also, solution to the eigenvector problem (and consequently SVD) is less stable to minor perturbations of the matrix than the solution to a system of linear equations if the desired eigenvector corresponds to an eigenvalue that is very close to other eigenvalues of the matrix (Golub and Van-Loan 1989). In Sect. 4.2, we have also compared the sensitivity of ICA and Spectral-SVD algorithms with respect to edge weights of the bipartite graph.

4 Theoretical analysis of ICA

4.1 Time complexity

Computational time of the proposed algorithm depends on the solution to Eq. 32. In particular, the time complexity is dependent on the number of non-zero entries in **L**, which asymptotically is $O(|E|)$. We can solve Eq. 32 using either a direct method such as Gaussian elimination or an iterative approach like the popular conjugate gradient method. Iterative methods have been popular due to their computational efficiency. Another advantage in favor of iterative methods is that a partial answer may be obtained at intermediate stages of the solution by specifying a fixed number of iterations. If we adopt the conjugate gradient method then the complexity of Eq. 32 is $O(|E|)$. Note that, this only measures the time complexity to compute the indicator vector. We also need to include the time complexity to employ the ratio cut which is of the order of $O(h\,logh)$ where $h = |R| + |M|$. Factoring this in, time complexity of the proposed algorithm is $O(|E| + h\,logh)$. Further, if a constant number of recursions are performed, then the time complexity reduces to $O(h\,logh)$. Empirical results on time complexity are presented in Sect. 5.5.

4.2 Sensitivity analysis

We examine the sensitivity of ICA and the Spectral-SVD algorithm to the edge weights. In the following, we first perform the sensitivity analysis with respect to a general parameter p.

4.2.1 *ICA*

Recall from Eq. 32, ICA requires a solution to a non-singular sparse system of linear equations represented by,

$$\begin{bmatrix} \mathbf{D}_R & -\mathbf{B} \\ -\mathbf{B}^T & \mathbf{D}_M \end{bmatrix}^* \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}^* = \begin{bmatrix} \mathbf{d}_R \\ \mathbf{d}_M \end{bmatrix}^* \tag{33}$$

where the three bipartite matrices are the Laplacian, indicator vector and the degree vector after vertex removal.

$$\mathbf{L}^*\mathbf{z}^* = \mathbf{d}^* \tag{34}$$

Differentiating this equation with respect to p,

$$\mathbf{L}^*\frac{\delta \mathbf{z}^*}{\delta p} = -\mathbf{z}^*\frac{\delta \mathbf{L}^*}{\delta p} + \frac{\delta \mathbf{d}^*}{\delta p} \tag{35}$$

For a given solution to Eq. 34, $\mathbf{L}^*$ and $\mathbf{z}^*$ are known and $\frac{\delta \mathbf{L}^*}{\delta p}$ can be determined analytically. In order to determine the derivative at point $\mathbf{z}^* = \begin{bmatrix} x^* \\ y^* \end{bmatrix}$, $\frac{\delta \mathbf{z}^*}{\delta p}$ can be solved for as a system of linear equations.

4.2.2 *Spectral-SVD*

The Spectral-SVD algorithm (Dhillon 2001; Zha et al. 2001) performs SVD of the matrix $\mathbf{D}_R^{-1/2}\mathbf{B}\mathbf{D}_M^{-1/2}$.

$$\mathbf{D}_R^{-1/2}\mathbf{B}\mathbf{D}_M^{-1/2}\mathbf{v} = (1-\lambda_2)\mathbf{u} \tag{36}$$

$$\mathbf{D}_M^{-1/2}\mathbf{B}^T\mathbf{D}_R^{-1/2}\mathbf{u} = (1-\lambda_2)\mathbf{v} \tag{37}$$

where $\mathbf{u}$ and $\mathbf{v}$ are the left and right singular vectors corresponding to the second (largest) singular value viz., $(1-\lambda_2)$. Letting $\mathbf{u} = \mathbf{D}_R^{1/2}\mathbf{x}$ and $\mathbf{v} = \mathbf{D}_M^{1/2}\mathbf{y}$ and with some algebraic manipulations we get,

$$\begin{bmatrix} \mathbf{D}_R & -\mathbf{B} \\ -\mathbf{B}^T & \mathbf{D}_M \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \lambda_2 \begin{bmatrix} \mathbf{D}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_M \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$
$$\mathbf{Lz} = \lambda_2 \mathbf{Dz} \tag{38}$$

where λ_2 is the Fiedler value of the bipartite **L**.

As before, we differentiate Eq. 38 with respect to p,

$$\mathbf{z}\frac{\delta \mathbf{L}}{\delta p} + \mathbf{L}\frac{\delta \mathbf{z}}{\delta p} = \mathbf{Dz}\frac{\delta \lambda_2}{\delta p} + \lambda_2 \mathbf{z}\frac{\delta \mathbf{D}}{\delta p} + \lambda_2 \mathbf{D}\frac{\delta \mathbf{z}}{\delta p} \tag{39}$$

Using Rayleigh quotient and the Chain rule, it is possible to calculate $\frac{\delta \lambda_2}{\delta p}$. The Rayleigh quotient is,

$$\lambda = \frac{\mathbf{z}^T \mathbf{Lz}}{\mathbf{z}^T \mathbf{z}} \tag{40}$$

Applying Chain rule,

$$\frac{\delta \lambda_2}{\delta p} = \frac{\delta \lambda_2}{\delta \mathbf{z}} \frac{\delta \mathbf{z}}{\delta p} \tag{41}$$

In the above equation, $\frac{\delta \lambda_2}{\delta \mathbf{z}}$ can be calculated from Eq. 40 as,

$$\frac{\delta \lambda_2}{\delta \mathbf{z}} = 2\mathbf{Lz}(\mathbf{z}^T \mathbf{z})^{-1} - 2\mathbf{z}^T \mathbf{Lz}(\mathbf{z}^T \mathbf{z})^{-2}\mathbf{z} \tag{42}$$

From Eq. 39, since all the terms are either known or can be calculated analytically, we get a system of linear equations which may be solved for $\frac{\delta \mathbf{z}}{\delta p}$.

4.2.3 Sensitivity to edge weight

We can analyze the effect of a specific parameter by finding $\frac{\delta L}{\delta p}$ and $\frac{\delta d}{\delta p}$ for the specific parameter. We can determine $\frac{\delta L^*}{\delta p}$ from $\frac{\delta L}{\delta p}$ by deleting the row and column corresponding to the removed vertex from $\frac{\delta L}{\delta p}$. For an edge weight $w(e_{ij})$,

$$\left(\frac{\delta d}{\delta w(e_{ij})}\right)_{r_i} = \begin{cases} 1, & \text{if } e_{ij} \text{ is incident on } r_i \\ 0, & \text{otherwise} \end{cases} \tag{43}$$

and,

$$\left(\frac{\delta d}{\delta w(e_{ij})}\right)_{m_j} = \begin{cases} 1 & \text{if } e_{ij} \text{ is incident on } m_j \\ 0, & \text{otherwise} \end{cases} \tag{44}$$

The matrix $\frac{\delta L}{\delta w(e_{ij})}$ is the **L** matrix with edge set reduced to $E = \{e_{ij}\}$.

Equations 35 and 39, show that the derivative of the isoperimetric solution is not degenerate. On the other hand, the spectral solution may be degenerate depending on the state of Fiedler vector and its value.

5 Experiments and results

In this section, we compare ICA with MCL (Dongen 2000; Enright et al. 2002) and the Spectral-SVD (Dhillon 2001; Zha et al. 2001) graph partitioning approach for data co-clustering. Section 5.1 discusses the evaluation methods used to report the experimental results. In Sect. 5.2, we present the results on co-clustering text documents and words using some of the publicly available datasets. We demonstrate the stability of ICA in solving the system of equations compared to performing SVD in the spectral approach by evaluating the performance in the presence of noise in Sect. 5.3. Section 5.4 is devoted to experiments we performed on co-clustering multimedia documents (images) and visual keywords (low-level image features). We also compare the computational speed of ICA, Spectral-SVD, and MCL in Sect. 5.5.

5.1 Evaluation methodology

In this section, we discuss the evaluation methods used to report the results on co-clustering. Traditional clustering reporting technique such as a confusion matrix has been used earlier to present co-clustering results, specially document-word co-clustering in Dhillon (2001), Zha et al. (2001) and Dhillon et al. (2003). In these works, a confusion matrix was used to demonstrate document clustering while top words from each of the clusters are displayed to show word clustering. Although somewhat helpful, this method does not give a complete picture of the co-clustering achieved. This is because, as discussed in Sects. 1 and 3, clustering of one data type induces clustering of the other and vice-versa. The goal of co-clustering is NOT to achieve perfect clustering of one data type but to achieve optimal co-clustering of the two data types together. So, a document confusion matrix might signify an optimal clustering on the documents but does not demonstrate the optimality of the document-word clustering by showing the top few words from every cluster. Due to the constraints enforced on the documents by word clustering, it should be clear that sub-optimal document confusion matrix can still result in an optimal document-word cluster. Similarly, an optimal document confusion matrix does not necessarily translate to an optimal document-word co-clustering. Hence, in addition to document confusion matrices and top words from every cluster, we also report the quality of documents and words co-clustering in terms of isoperimetric ratios as explained below.

The relationship between the Fiedler value of a graph and the isoperimetric constant has been demonstrated in some of the classic papers in graph theory papers such as Fiedler (1973, 1975a,b); Alon and Milman (1985); Alon (1986) and Cheeger (1970). In fact the goals of graph partitioning algorithms in general is to minimize the isoperimetric ratio. Hence, isoperimetric ratio can naturally be used to evaluate the goodness of co-clustering. ICA, Spectral-SVD, and MCL algorithms are compared in terms of

Table 1 Summary of the datasets used for *k*-partioning documents and words

Dataset	No. of clusters	No. of words	No. of docs
oh0	10	3182	1003
oh5	10	3012	918
oh10	10	3238	1050
oh15	10	3100	913
re0	13	2886	1504
re1	25	3758	1657
wap	20	8460	1560
tr11	9	6429	414
tr12	8	5804	313
tr21	6	7902	336
tr23	6	5832	204
tr31	7	10128	927
tr41	10	7454	878
tr45	10	8261	690

the isoperimetric ratio by employing the ratio cut to find the optimal partition (i.e., not necessarily partitions of equal size).

5.2 Documents and words co-clustering

For the experiments on documents and words co-clustering, we have used some of the publicly available datasets:

(1) We used the Medline and Cranfield dataset available from Cornell University.[1] Medline consists of abstracts in biomedicine received from the National Library of Medicine. Cranfield is a collection of abstracts in aeronautics and related areas originally used for tests at the Cranfield Institute of Technology in Bedford, England.

(2) We have also utilized the dataset used in Han and Karypis (2000).[2] Summary of this dataset is shown in Table 1. Data sets oh0, oh5, oh10 and oh15 are from OHSUMED collection (Hersh et al. 1994). Data sets re0 and re1 are from Reuters-21578 text categorization test collection Distribution 1.0 (Lewis 1999). Data set wap is from the WebACE project (WAP) (Boley et al. 1999). Each document corresponds to a web page listed in the subject hierarchy of Yahoo! Data sets tr11, tr12, tr21, tr23, tr31, tr41 and tr45 are derived from TREC-5 , TREC-6, and TREC-7 collections (TREC 1996, 1997, 1998).

For bipartitioning tests, we mixed some of the datasets mentioned above. Table 2 shows the bipartitioning datasets. These datasets were created as follows:

(1) Med-Cran dataset is Medline and Cranfield datasets mixed together.

(2) ArachidonicAcids-Hematocrit was derived from oh10 using Arachidonic Acids and Hematocrit classes.

[1] ftp://ftp.cs.cornell.edu/pub/smart.

[2] http://www.cs.umn.edu/~han/data/tmdata.tar.gz.

Table 2 Summary of the datasets used for bipartitioning documents and words

Dataset	No. of words	No. of docs
Med-Cran	9181	2431
ArachidonicAcids-Hematocrit	2353	274
Interest-Trade	2682	538
Sugar-Ship	2348	243
Mexico-Uric	1650	107

(3) Interest-Trade was formed by mixing Interest and Trade classes of re0 dataset.
(4) 2 classes from re1, viz. Sugar and Ship were combined to form the Sugar-Ship dataset.
(5) Classes Mexico and Uric-Acid from oh0 dataset were mixed to create Mexico-Uric.

The stop-words from all the datasets have been removed and words stemmed using Porter's suffix-stripping algorithm (Porter 1980).

In Sect. 3.1, we discussed that removing the vertex with maximum degree from the bipartite graph is the best choice to solve Eq. 31. We now show this empirically by comparing three strategies of removing the maximum degree, minimum degree and a random vertex for bipartitioning. For random vertex removal, three randomly chosen vertices were removed and the vertex that gave the best isoperimetric ratio was reported. The results on all the 5 bipartitioning datasets are shown in Table 3. Maximum vertex removal is denoted by ICA-MaxVR, minimum by ICA-MinVR, random by ICA-RanVR and the Spectral-SVD by Spec-SVD. The last column denotes the performance of the MCL algorithm. As can be seen, ICA-MinVR yields partitions with comparatively high isoperimetric ratios than ICA-MaxVR. Moreover, mininum degree vertex tends to lie along the ideal cut for the bipartite graph and removing such a vertex can be disastrous as was demonstrated in the bipartite dumbbell example earlier (Fig. 3c, d). ICA-RanVR can sometimes slightly outperform ICA-MaxVR (Med-Cran, Interest-Trade). However, due to the randomness associated with it, there is always a possibility that it can actually end up being ICA-MinVR and perform poorly at times. As a result, ICA-RanVR lacks in consistency in terms of guaranteed optimal partitioning of the bipartite graph. Moreover, the difference in ratios of ICA-MaxVR and ICA-RanVR when ICA-RanVR does outperform is very negligible. For the rest of the experiments, we have employed ICA-MaxVR and is referred to as ICA from now on.

Table 3 Isoperimetric Ratio of ICA (max, min & random vertex removal), Spec-SVD, and MCL for bipartitioning all the datasets

Dataset	ICA-MaxVR	ICA-MinVR	ICA-RanVR	Spec-SVD	MCL
Med-Cran	0.1485	0.1647	0.1415	0.2281	0.1915
ArachidonicAcids-Hematocrit	0.2455	0.2611	0.2704	0.4142	0.3211
Interest-Trade	0.2872	0.3971	0.2868	0.4652	0.3566
Sugar-Ship	0.2036	0.2593	0.2910	0.3648	0.2410
Mexico-Uric	0.2666	0.2825	0.2964	0.3345	0.2989

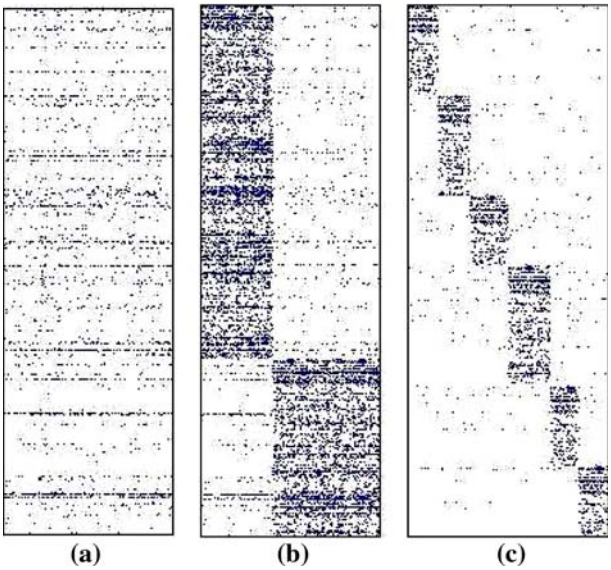


Fig. 5 (**a**) Sparseness of typical word-document matrix (shown here *Med-Cran*) before co-clustering (**b**) *Med-Cran* matrix after bipartitioning (**c**) *tr21* matrix after *k*-partitioning

In Fig. 5a, the sparsity pattern of a typical word-document matrix is shown (*Med-Cran* in the figure). Co-clustering this dataset (i.e., bipartitioning the bipartite graph) essentially leads to re-ordering of the rows and columns such that words and documents are co-clustered together. This is denoted by the two dense sub-matrices in Fig. 5b.

In Table 4, we present the results of the three algorithms on the bipartitioning datasets in terms of the confusion matrices and the top words from every cluster. From the confusion matrices, we can infer that the algorithms obtain similar results on Med-Cran, ArachidonicAcids-Hematocrit, and Interest-Trade datasets. The inability of Spec-SVD and MCL to partition complex datasets is evident on Sugar-Ship and Mexico-Uric that consist of significant number of overlapping words appearing between the two clusters. While ICA is able to effectively achieve co-clustering, the other two algorithms classify all the documents in both these datasets into one cluster. Based on the document cluster label, we examined the top words and observed that the ICA top words consist of popular words from that class. As expected, this is true for Spec-SVD and MCL top words as well but only when the algorithm does not misclassify all the documents.

We performed *k*-partitioning on the datasets mentioned in Table 1 by recursively applying ICA, Spec-SVD, and MCL algorithms. Tables 5–16, report the document confusion matrices and top words from the clusters on some of the *k*-partitioning datasets. We can see that ICA clustering is quite balanced across clusters. On the other hand, Spec-SVD and MCL, at times have a tendency to classify large number of documents from different classes into a single cluster. We also compare the performance of the three algorithms for *k*-partitioning in terms of Mean Isoperimetric Ratio, which

Table 4 Confusion matrix of documents and top words from every cluster on the bipartitioning datasets

Datasets		D_1	D_2	Top words
Med-Cran	ICA	1012	21	W_1: tumeur, trillat, perineum, osteosarcom, osseus, kugl, joss
		0	1398	W_2: random, havelock, fatigu, stratagem, calculus, open, cx
	Spec-SVD	1033	0	W_1: joss, osseus, kugl, osteosarcom, trillat, analag, perineum
		0	1398	W_2: institut, transit, brit, rapid, show, list, coh
	MCL	1029	4	W_1: kugl, osteosarcom, ablat, abscess, abnorm, perineum, capsul
		8	1390	W_2: midget, coh, team, transit, dict, rapid, institut
ArachidonicAcids-Hematocrit	ICA	17	109	W_1: congenit, seroneg, cmv, suscept, whenev, cytomegaloviru, centre
		128	20	W_2: colon, experiment, plac, latter, injuri, heate, vasodil
	Spec-SVD	27	99	W_1: whenev, cmv, congenit, suscept, cytomegaloviru, seroneg, help
		147	1	W_2: condens, evid, conserv, modifi, anti, turnov, kidnei
	MCL	24	102	W_1: blunt, intak, preferenti, regul, assess, sd, packag
		148	0	W_2: ascertain, methodologi, condens, releas, nutrition, potenc, thyrotropin
Interest-Trade	ICA	218	1	W_1: mercantil, bancorp, chicago, marin, quote, st, johnson
		51	268	W_2: acquir, farm, revalu, difficult, joint, jean, slap
	Spec-SVD	212	7	W_1: mercantil, chicago, bancorp, st, marin, quote, fhlb
		74	245	W_2: easili, british, fundament, yen, posit, seen, applic
	MCL	219	0	W_1: bancorp, depart, mercantil, sloane, sudden, chicago, imbal
		68	251	W_2: subsidis, signific, applic, penetr, announc, fairli, seen
Sugar-Ship	ICA	2	104	W_1: uae, saudi, arab, ra, cold, command, lull
		136	1	W_2: advanc, system, held, prospect, offici, deliv, nov
	Spec-SVD	1	105	W_1: bolivia, el, gabon, uruguai, trinidad, tobago, haiti
		0	137	W_2: jamaica, argentina, cumul, salvador, denatur, colombia, authoris
	MCL	6	100	W_1: variou, jacque, previous, individu, push, nacion, lo
		0	137	W_2: polit, lai, argentina, breed, situat, jamaica, themselv
Mexico-Uric	ICA	40	11	W_1: client, cit, anonym, charge, abort, percept, reason
		10	46	W_2: combin, nor, sleep, common, hg, hemoglobin, investig

Table 4 continued

Datasets		D_1	D_2	Top words
	Spec-SVD	50	1	W_1: constitu, phe, ident, peptid, lipoprotein, insensit, densiti
		56	0	W_2: equival, cost, situat, qualiti, econom, servic, termin
	MCL	42	9	W_1: microgram, constitu, convers, referr, consist, lipoprotein, elderli
		53	3	W_2: incur, radio, solution, equival, viii, emphas, diffus

Table 5 Confusion matrix of documents and top words from every cluster on the tr12 dataset using ICA

Class	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
54	1	1	0	0	30	0	0	2
58	0	0	24	5	0	0	0	0
77	0	0	0	9	0	0	0	0
78	22	8	0	0	0	0	0	0
82	0	0	0	0	0	20	0	15
94	0	20	0	0	0	0	9	0
95	0	0	2	0	4	0	0	48
100	15	0	0	0	0	18	60	0

W_1: citi, ecolog, ecologi, organ, inform, archiv, source
W_2: feder, russia, law, econom, parti, region, reform
W_3: strike, union, worker, govern, railroad, seoul, header
W_4: border, fish, tiger, guard, water, japanes, russian
W_5: satellit, space, launch, develop, commun, system, rocket
W_6: develop, product, research, materi, industri, amp, genet
W_7: export, design, produc, foreign, equip, nuclear, special
W_8: percent, bank, compani, czech, dec, million, servic

is the mean of the isoperimetric ratios from recursive application of the algorithms. These results are shown in Table 17. As is evident, ICA consistently outperforms Spec-SVD and MCL on all the datasets. Figure 5c shows the *tr21* word-document matrix after co-clustering (i.e., k-partitioning with $k = 6$).

5.3 Stability of ICA in solving system of equations vs. performing SVD in the spectral approach

Unlike ICA, that solves a sparse system of linear equations, the spectral approach requires solution to an eigenvalue or singular value decomposition problem. With reference to the discussion in Sect. 3.3, Spec-SVD might not be stable in the presence of noise. Secondly, it would be interesting to see whether ICA outperforms Spec-SVD in the presence of different kinds of noises. Moreover, real world text corpora consist of a lot of noise which can be difficult to eliminate in spite of extensive data preprocessing. We have compared the performance of ICA and Spec-SVD in the presence of Gaussian additive and multiplicative noise. Additive noise had zero mean with

Table 6 Confusion matrix of documents and top words from every cluster on the tr12 dataset using Spec-SVD

Class	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
54	20	7	0	0	4	2	0	1
58	0	0	0	0	29	0	0	0
77	0	0	0	0	0	2	0	7
78	0	1	0	23	0	6	0	0
82	0	0	8	0	0	3	22	2
94	0	24	0	0	0	5	0	0
95	0	0	33	2	0	19	0	0
100	2	0	3	1	22	65	0	0

W_1: station, research, orbit, space, project, arian, earth
W_2: author, constitut, nation, power, polit, elect, protect
W_3: secur, gmt, slovak, million, econom, tax, industri
W_4: address, tel, club, natur, people, ecolog, oblast
W_5: seoul, trade, engin, oper, subwai, solidar, report
W_6: item, bank, compani, invest, materi, accord, govern
W_7: plant, cell, vol, equip, applic, institut, activ
W_8: ship, phrase, kuril, sakhalin, poacher, vessel, territori

Table 7 Confusion matrix of documents and top words from every cluster on the tr12 dataset using MCL

Class	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
54	15	12	0	0	0	1	1	5
58	1	1	0	0	27	0	0	0
77	0	0	0	0	0	9	0	0
78	0	1	0	4	19	6	0	0
82	0	15	15	0	0	0	5	0
94	0	19	0	0	0	2	0	8
95	26	0	0	24	1	1	1	1
100	3	40	23	10	0	17	0	0

W_1: pulp, forth, interconnect, reloc, affirm, false, municip
W_2: aziz, unusu, align, popul, scale, acquir, ot
W_3: outcom, superconductor, resum, compat, tongxun, benz, ball
W_4: technologi, electromechan, enabl, friendli, thank, strength, creation
W_5: willing, vaccinia, petroleum, ore, nucleu, plenti, introductori
W_6: cadmium, enlist, match, campo, speed, qualiti, effici
W_7: elev, keizai, toward, admir, nato, incomplet, death
W_8: review, vancouv, encroach, exactli, drink, vol, pardon

variance increasing from 1 to the maximum value in the original data. Multiplicative noise had mean of 1 with its variance going from 1 to a maximum of 5. Figures 6 and 7 compare the performance for bipartitioning in the presence of additive and multiplicative noise. We followed the same procedure to observe the effect of additive and multiplicative noise on k-partitioning. We present these results for 6 of the datasets shown in Table 1. These results are shown in Figs. 8 and 9, respectively. We see that inspite of the varying amounts and kinds of noise in the data, ICA is able to perform optimal partitioning indicated by its low isoperimetric ratio. Second noticeable fact is in regards to stability. Rising ratios as the variance increases indicates that the performance of algorithm is gradually decreasing. However, fluctuating ratios indicates

Table 8 Confusion matrix of documents and top words from every cluster on the tr21 dataset using ICA

Class	D_1	D_2	D_3	D_4	D_5	D_6
251	0	201	0	0	2	28
252	13	0	0	0	3	0
254	0	0	0	2	7	0
255	5	0	0	1	0	35
257	0	0	25	0	2	8
259	0	0	0	4	0	0

W_1: crime, program, unit, bill, title, law, grant
W_2: section, amend, agreem, act, servic, tax, subsect
W_3: cigarett, tobacco, smok, market, itag, cent, hyph
W_4: hoover, summer, center, fbi, book, crime, kennedi
W_5: compani, technologi, drug, develop, heart, center, health
W_6: environment, countri, page, govern, mexico, environ, industri

Table 9 Confusion matrix of documents and top words from every cluster on the tr21 dataset using Spec-SVD

Class	D_1	D_2	D_3	D_4	D_5	D_6
251	3	173	7	0	0	48
252	8	0	2	0	0	6
254	0	0	0	2	0	7
255	34	1	1	5	0	0
257	0	4	0	1	23	7
259	0	0	0	2	0	2

W_1: center, people, pollution, waste, time, trade, kolbe
W_2: bill, unit, amend, paragraph, mexico, american, nafta
W_3: subtitl, victim, nation, violenc, follow, servic, senat
W_4: organ, evid, time, investig, date, presid, law
W_5: page, frnewlin, philip, countri, morri, text, advertis
W_6: cost, research, medic, page, care, angioplasti, nasa

Table 10 Confusion matrix of documents and top words from every cluster on the tr21 dataset using MCL

Class	D_1	D_2	D_3	D_4	D_5	D_6
251	1	142	37	0	51	0
252	10	1	4	0	0	1
254	0	0	0	0	4	5
255	0	1	1	0	0	39
257	10	20	2	3	0	0
259	0	0	4	0	0	0

W_1: percent, crewmemb, illinoi, specia, overal, exacerb, unravel
W_2: electron, tight, disintegr, practition, clean, texa, we
W_3: balleng, glue, manzullo, emphas, destini, god, deck
W_4: stamped, baesler, adjourn, vehicl, woodwork, lobbyist, racism
W_5: applianc, stori, interconnect, evapor, instantli, alabama, ceas
W_6: put, declin, slack, ei, copper, salvador, graham

Table 11 Confusion matrix of documents and top words from every cluster on the tr23 dataset using ICA

Class	D_1	D_2	D_3	D_4	D_5	D_6
271	6	25	0	0	0	5
272	0	1	0	0	1	9
273	0	15	3	2	71	0
274	29	9	2	0	5	0
277	1	1	0	0	13	0
280	0	0	0	4	2	0

W_1: senat, house, car, vehicl, propos, fuel, develop
W_2: pjg, frnewlin, itag, energi, solar, system, electr
W_3: dornan, hunter, mine, war, people, time, page
W_4: ivori, eleph, ban, african, africa, anim, wildlif
W_5: chairman, fund, amend, gentleman, program, rep, budget
W_6: care, hospit, center, patient, health, rule, project

Table 12 Confusion matrix of documents and top words from every cluster on the tr23 dataset using Spec-SVD

Class	D_1	D_2	D_3	D_4	D_5	D_6
271	6	0	0	20	10	0
272	9	0	0	2	0	0
273	35	1	53	0	2	0
274	1	0	0	31	8	5
277	4	0	0	11	0	0
280	0	5	0	1	0	0

W_1: cost, speaker, servic, provid, research, time, medic
W_2: project, follow, fund, manag, agree, motor, appropri
W_3: amend, rep, nafta, gentleman, budget, program, center
W_4: power, water, qtag, electr, text, page, renew
W_5: command, somalia, countri, kill, rep, armi, date
W_6: poacher, profil, speci, pub, text, people, kenya

instability and inconsistency to partition optimally. In some of these figures, ratios of both the algorithms have fluctuated in limits. However, in quite a few of these experiments, Spec-SVD demonstrates instability. Some examples of these are Fig. 6 (Interest-Trade, Sugar-Ship, Mexico-Uric), Fig. 7 (Sugar-Ship, Mexico-Uric), Fig. 8 (re0, wap) and in Fig. 9 (oh15, re0, tr11). To demonstrate the stability of both the algorithms, we calculated the mean of standard deviation of the isoperimetric ratios of both the algorithms for bipartitioning and k-partitioning in the presence of the noise. These results are shown in Table 18. Higher standard deviation of Spec-SVD indicates instability to partition the noisy datasets.

5.4 Co-clustering image features and relevance feedback logs

In this section, we perform experiments to co-cluster multimedia data. Specifically, we perform co-clustering of low-level image features and accumulated user relevance

Table 13 Confusion matrix of documents and top words from every cluster on the tr23 dataset using MCL

Class	D_1	D_2	D_3	D_4	D_5	D_6
271	25	10	0	1	0	0
272	0	3	0	0	0	8
273	29	40	0	0	22	0
274	9	0	0	15	0	21
277	0	3	12	0	0	0
280	1	1	0	4	0	0

W_1: vast, hilliard, siemen, chile, mobil, lightli, propuls
W_2: appear, strenuous, trillion, coloc, rise, care, beache
W_3: nec, hang, depriv, skelton, cover, mi, impress
W_4: gasolin, children, sake, inelig, thrown, content, pretti
W_5: harden, formul, coffee, refuge, vii, school, deplet
W_6: excit, classroom, double, powell, fresh, correctli, lincoln

Table 14 Confusion matrix of documents and top words from every cluster on the tr31 dataset using ICA

Class	D_1	D_2	D_3	D_4	D_5	D_6	D_7
301	40	0	296	0	0	9	7
302	43	1	17	2	0	0	0
304	0	0	10	101	0	35	5
305	0	1	0	1	0	1	18
306	0	186	31	10	0	0	0
307	12	0	0	0	20	79	0
310	0	0	0	0	2	0	0

W_1: health, chairman, vaccin, percent, center, program, million
W_2: force, presid, people, govern, nation, somalia, africa
W_3: drug, section, amend, defens, author, unit, secretari
W_4: itag, frnewlin, land, blank, forest, nation, bill
W_5: cellular, school, phone, telephon, compani, cancer, week
W_6: percent, project, million, invest, industri, develop, fund
W_7: car, vehicl, driver, accid, firefight, truck, date

Table 15 Confusion matrix of documents and top words from every cluster on the tr31 dataset using Spec-SVD

Class	D_1	D_2	D_3	D_4	D_5	D_6	D_7
301	0	0	0	0	344	1	7
302	0	0	3	3	48	3	6
304	4	2	26	0	83	4	32
305	0	0	0	0	0	21	0
306	0	150	72	0	0	0	5
307	59	6	12	32	0	1	1
310	2	0	0	0	2	0	0

W_1: text, headlin, fire, station, section, length, docid
W_2: speci, amend, chairman, protect, servic, center, endang
W_3: amend, children, vaccin, health, gentleman, time, act
W_4: troop, africa, kill, countri, somalia, report, war
W_5: program, nation, offic, fund, report, provision, force
W_6: frnewlin, itag, product, foreign, increas, billion, bank
W_7: cent, scare, brain, commun, radio, dollar, date

Table 16 Confusion matrix of documents and top words from every cluster on the tr31 dataset using MCL

Class	D_1	D_2	D_3	D_4	D_5	D_6	D_7
301	196	89	0	0	0	0	67
302	39	0	0	9	15	0	0
304	2	1	44	99	0	0	5
305	0	19	0	0	0	0	2
306	101	11	2	0	105	0	8
307	1	4	3	66	0	30	7
310	1	1	0	0	1	0	1

W_1: car, forai, metropol, airfield, ineffici, flaw, adjac
W_2: entrench, eye, unknown, edward, road, filter, dramat
W_3: achiev, lapse, sarawak, wenzhou, asahi, celebr, vista
W_4: intent, auster, arrang, underscor, merdeka, extrem, pjg
W_5: thin, oklahoma, exist, olsen, afflict, mo, shore
W_6: diet, jefferson, ran, shin, merced, thoi, minimum
W_7: engin, parti, unit, reforest, loose, turtle, imprison

Table 17 Mean isoperimetric ratios of ICA, Spec-SVD, and MCL in k-partitioning all the datasets

Dataset	ICA	Spec-SVD	MCL
oh0	0.1867	0.3247	0.2954
oh5	0.1870	0.4418	0.3268
oh10	0.2255	0.3818	0.2861
oh15	0.1816	0.4359	0.2259
re0	0.2015	0.4271	0.3242
re1	0.1516	0.3509	0.2175
wap	0.2804	0.4210	0.3956
tr11	0.2817	0.4605	0.3750
tr12	0.2558	0.4508	0.3179
tr21	0.3356	0.5248	0.4054
tr23	0.2769	0.4832	0.4265
tr31	0.2622	0.4386	0.3247
tr41	0.2756	0.46	0.3969
tr45	0.2189	0.4759	0.2985

feedbacks in a content-based image retrieval (CBIR) system (Smeulders et al. 2000). In the following, we explain the data preparation.

CBIR systems typically represent images using a vector consisting of low-level features extracted from the images. The features could be values derived from certain image characteristics such as color and texture or it could be raw pixel intensities (Gonzalez and Woods 2002). Usually, a number of features chosen for image representation, say k, are specific to a system. We define a k by n Feature-Image matrix, $\mathbf{FI}$, where column vectors correspond to the feature vectors for the n images while rows denote the k chosen low-level features used to represent images. In a query by example scenario in CBIR retrieval process, users query the system by providing an example image. The system retrieves images from the database and presents them to the user. User then provides relevance feedback wherein relevant images (or irrelevant) to the users query are marked positive (or negative) (Rui et al. 1997). By keeping the users

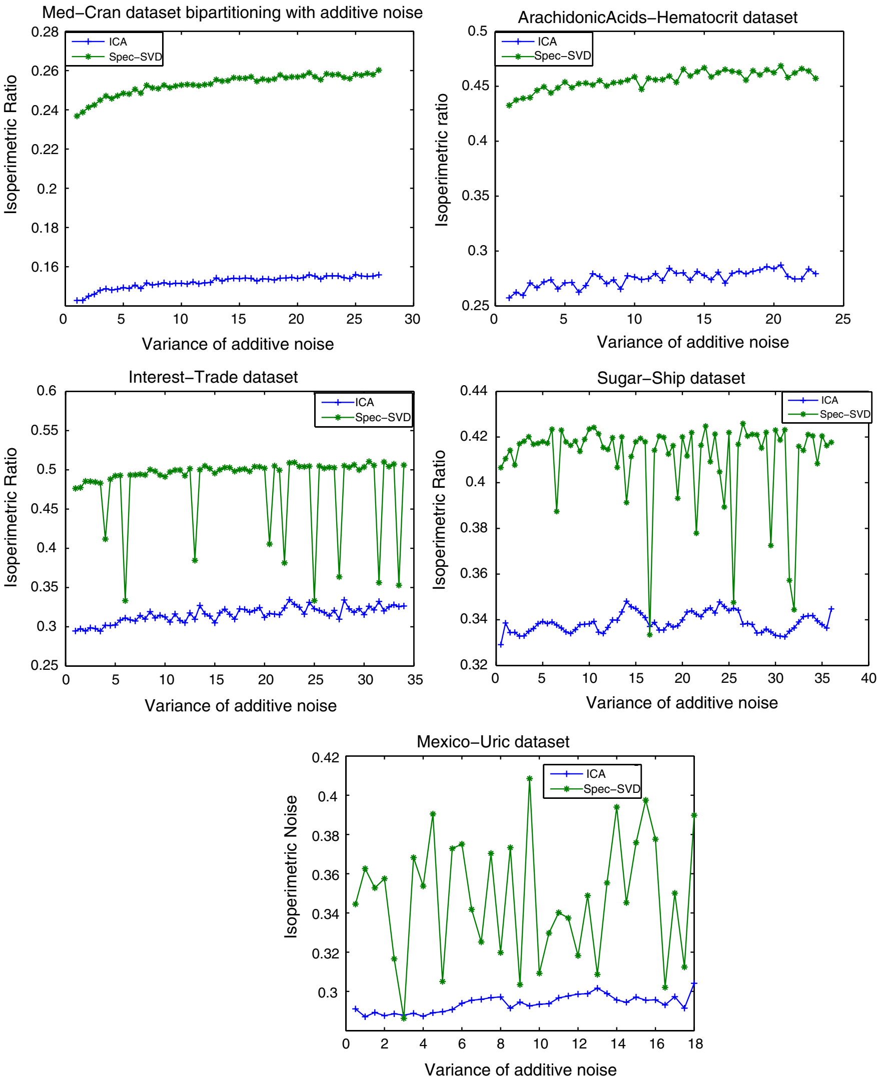


Fig. 6 Results for bipartitioning all the datasets in the presence of Gaussian additive Noise with 0 mean and variance increasing from 1 to maximum value in the data matrix

in the loop during retrieval, a CBIR system is able to adjust query at every iteration for subjectivity in judgement by learning from the feedback received. Feedback logs accumulated from user interactions can be used to create an m by n binary matrix **LI**, which we call the Log-Image matrix in which columns represent the n images and rows represent m user feedback logs collected. For an entry LI_{ij} in this matrix,

$$LI_{ij} = \begin{cases} 1, & \text{if image } I_j \text{ was marked positive in log } L_i \\ 0, & \text{otherwise} \end{cases} \tag{45}$$

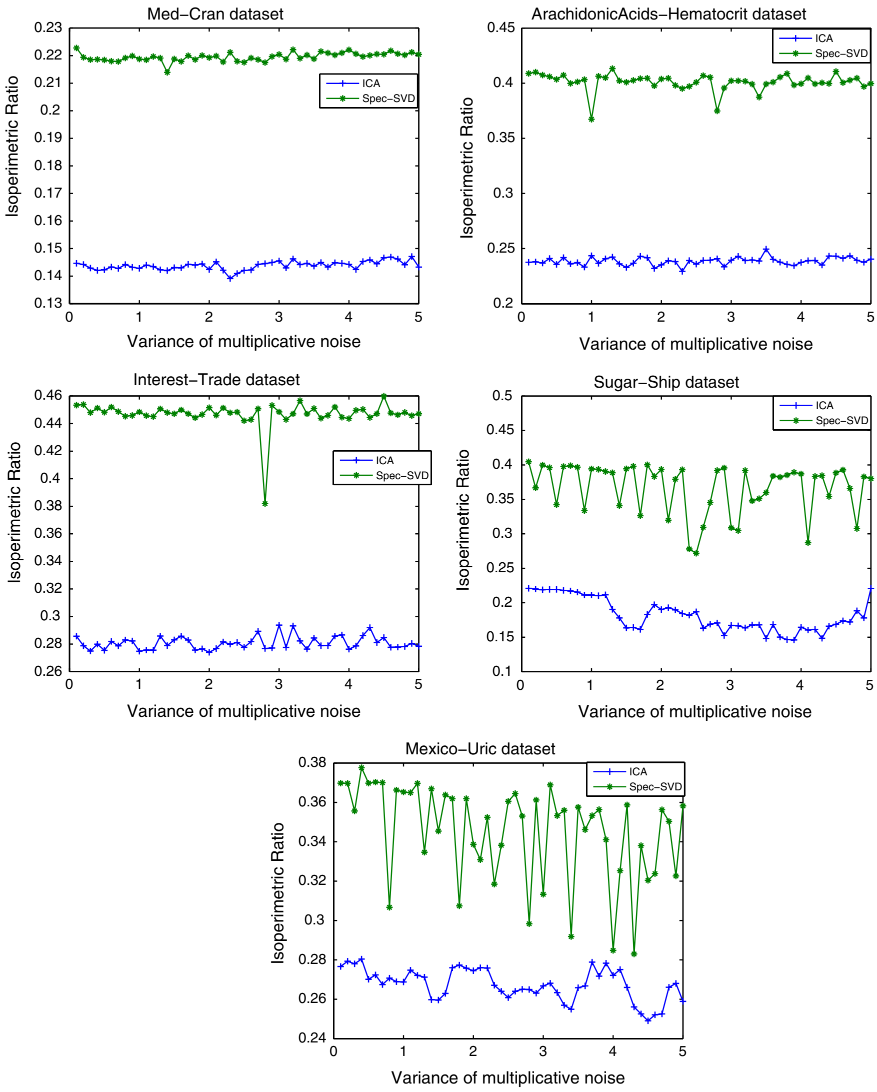


Fig. 7 Results for bipartitioning all the datasets in the presence of Gaussian multiplicative Noise with unit mean and variance increasing from 1 to 5

In Rege et al. (2006b), we have proposed integrating the low-level feature information in **LI** and the high-level semantic information from the users together by defining a Log-Feature matrix **LF** as

$$LF_{ij} = \mathbf{l}_i \cdot \mathbf{f}_j \tag{46}$$

where $\mathbf{l}_i$ and $\mathbf{f}_j$ are the corresponding row vectors from the **LI** and **FI** matrices, respectively. The **LF** matrix can be represented using a bipartite graph where the two kind of

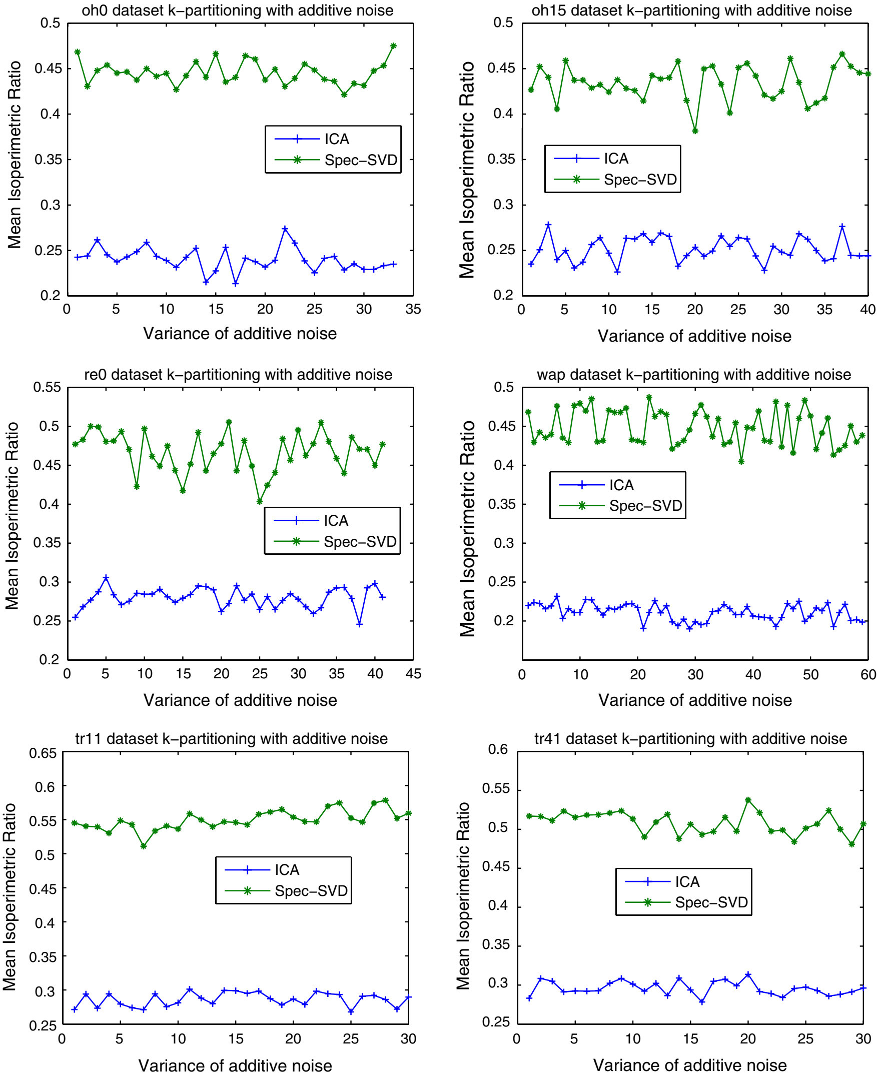


Fig. 8 Results on 6 of the datasets for k-partitioning in the presence of Gaussian additive noise as before

vertices are the logs and the features. Co-clustering these two together, we can identify features relevant to a semantic concept and hence narrow the semantic gap (Zhao and Grosky 2002).

We generated 1,250 user feedback logs on 5 image categories from the Corel image database. Each image category had a total of 250 user feedback logs. In a typical CBIR relevance feedback, a user marks a few positive images from the ones presented to him/her. In the user feedback logs generated, we randomly had 5 images selected from that category as positive while the rest were considered to be negative. The image categories used in our experiments were *Arabian Horses, Playing Cards,*

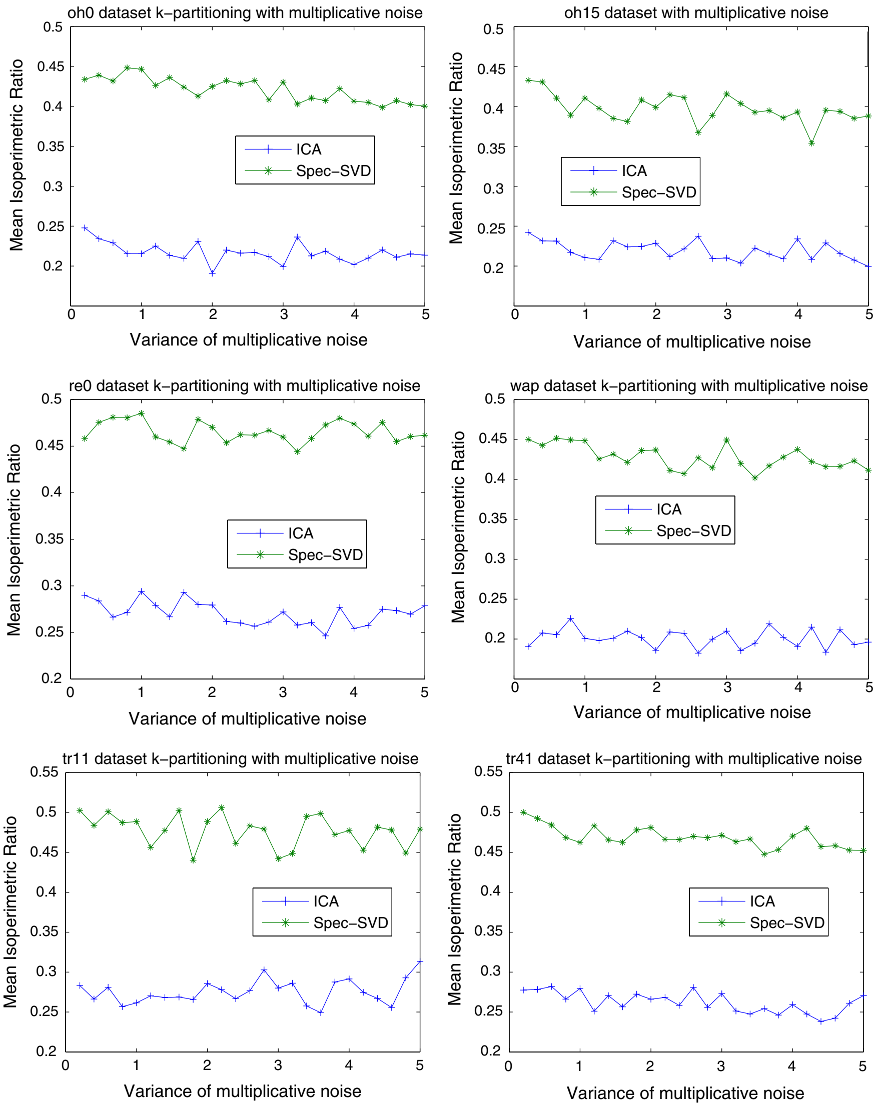


Fig. 9 Results on 6 of the datasets for *k*-partitioning in the presence of Gaussian multiplicative noise as before

Table 18 Mean standard deviation of the isoperimetric ratios of ICA and Spec-SVD over all the noisy datasets

Experiment	ICA	Spec-SVD
Bipartitioning with additive noise	0.57×10^{-2}	2.21×10^{-2}
Bipartitioning with multiplicative noise	0.84×10^{-2}	1.6×10^{-2}
k-partitioning with additive noise	1.1×10^{-2}	1.88×10^{-2}
k-partitioning with multiplicative noise	1.27×10^{-2}	1.53×10^{-2}

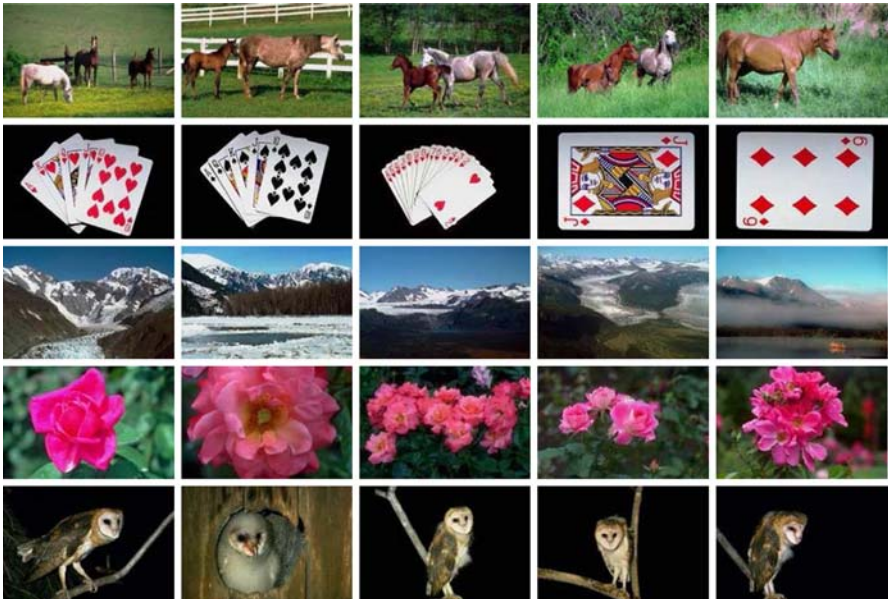


Fig. 10 A few sample images from each of the image categories used in our experiments

Glaciers & Mountains, Roses and Owls. Each category consists of 100 images. Some of the sample images from these categories are shown in Fig. 10. For the image features, we adopted the HSV space and performed principal component analysis (PCA) along H, S and V dimensions separately. The image was then projected in the eigen vector space to get weights along the principal components. A feature vector for each image was formed by concatenating weights along the three dimensions (Rege et al. 2006b).

Table 19 Mean isoperimetric ratios of partitioning 3, 4 & all 5 image categories mixed together

Dataset	ICA	Spec-SVD	MCL
Horses-Cards-Mountains	0.7737	0.8137	0.7803
Horses-Cards-Roses	0.8482	0.8739	0.8536
Horse-Cards-Owls	0.7546	0.8486	0.7639
Horses-Mountains-Roses	0.8042	0.8890	0.8243
Horses-Mountains-Owls	0.8092	0.8445	0.8462
Horses-Roses-Owls	0.7829	0.8914	0.8846
Cards-Mountains-Roses	0.8062	0.8510	0.8727
Cards-Mountains-Owls	0.7518	0.8427	0.8514
Mountains-Roses-Owls	0.7691	0.8769	0.7964
Horses-Cards-Mountains-Roses	0.8575	0.8769	0.9128
Horses-Cards-Mountains-Owls	0.7965	0.8442	0.8547
Cards-Mountains-Roses-Owls	0.8452	0.8774	0.8682
Horses-Mountains-Roses-Owls	0.8183	0.8798	0.9125
Horses-Cards-Roses-Owls	0.8775	0.8919	0.8911
Horses-Cards-Mountains-Roses-Owls	0.7523	0.8834	0.8351

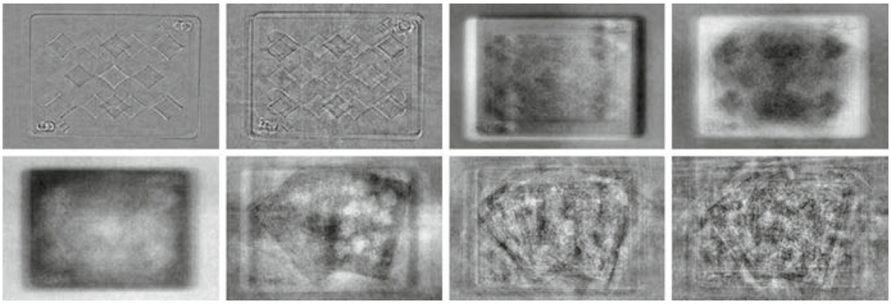


Fig. 11 Principal Component images of *Playing Cards* category obtained after the partitioning of the 3 categories *Horses-Cards-Mountains*

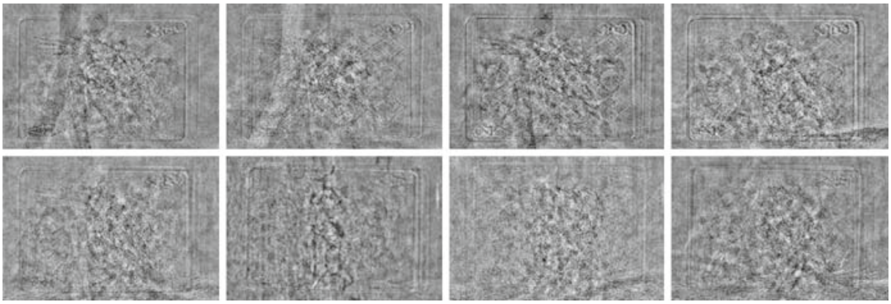


Fig. 12 Principal Component images of *Owls* category obtained after the partitioning of the 3 categories *Horse-Cards-Owls*

To evaluate the partitioning of this dataset, we mixed some of these categories. In Table 19, we report the results for partitioning by mixing 3, 4 and all 5 image categories. As before, Mean Isoperimetric Ratio was calculated. Ratios obtained with ICA are lower than the ones obtained with Spec-SVD and MCL. Also, to demonstrate that ICA performs optimal partitioning on this dataset, we performed an additional experiment. After co-clustering of logs and features, every cluster consists of semantic information in the form of logs and the corresponding image features representative of the semantic information. Since, image features used in our experiments are the weights for the principal components, we can display the principal component images corresponding to the V space from a cluster. In Fig. 11, we display the principal component images from one of the clusters after the k-partitioning of *Horses-Cards-Mountains*. Figure 12 shows the principal component images from one cluster in the partitioning of *Horse-Cards-Owls*. From Fig. 11, we can see that the images representing the feature components are indeed *Playing Cards* like. Also in Fig. 12, in the center of the images we can see *Owls* like structure distinctly. That is, the features corresponding to this cluster are able to represent the underlying semantic concept. We get similar feature component images for other clusters. This experiment visually demonstrates that ICA has been able to optimally co-cluster logs and features and that the algorithm works well in practice.

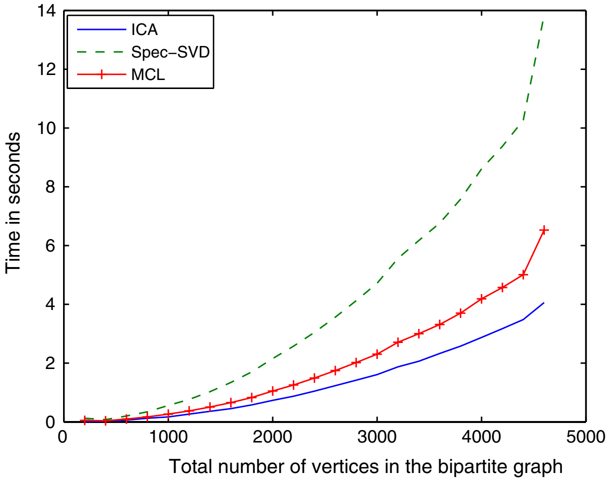


Fig. 13 Computational speed comparison for ICA, Spec-SVD, and MCL. The time required by each of the algorithms to compute the indicator vectors are displayed for varying number of nodes in the bipartite graph

5.5 Computational speed comparison

Analysis of time complexity of ICA was done in Sect. 4.1. We now compare the computational speed of ICA with Spec-SVD and MCL. The time complexity for the three algorithms is dependent on the sparseness of the data matrix. In other words, it takes more time to partition a densely connected bipartite graph compared to a sparsely connected one. For this reason, we considered the worst case scenario of a fully connected bipartite graph (with uniform weights) where every vertex of one data type is connected with all the vertices of the other data types. Since, the time required to cut the indicator vector is same for all the algorithms, we compare on the basis of the time required to calculate the indicator vector. The experiment was performed on a machine with a 3 GHz Intel Pentium 4 processor with 1 GB RAM. In Fig. 13, we plot the time required by the three algorithms as the number of vertices in the fully connected bipartite graph increase. We can see that, Spec-SVD time increases very rapidly compared to ICA and MCL that both rise steadily. Amongst the three algorithms, ICA proves to be the quickest.

6 Conclusions

We treated the data co-clustering problem as a weighted bipartite graph partitioning problem. The two data types in data co-clustering were modeled as the two sets of vertices of a bipartite graph. In order to obtain co-clustering, we proposed the Isoperimetric Co-clustering Algorithm—a new method for partitioning the bipartite graph. The proposed algorithm requires a simple solution to a sparse system of linear equations. Experiments performed on text as well as multimedia datasets demonstrate the advantages of our approach over other approaches in terms of the quality, efficiency and stability in partitioning the bipartite graph.

Acknowledgements This research was partially funded by the 21st Century Jobs Fund Award, State of Michigan, under grant: 06-1-P1-0193, and by National Science Foundation, under grant: IIS-0713315.

References

Alon N (1986) Eigenvalues and expanders. Combinatorica 6(2):83–96

Alon N, Milman VD (1985) λ_1 isoperimetric inequalities for graphs, and superconcentrators. J Comb Theory Ser B 38:73–88

Alpert CJ, Kahng AB (1995) Recent directions in netlist partitioning: a survey. Integr VLSI J 19(12):1–81

Anderson WN, Morley TD (1985) Eigenvalues of the laplacian of a graph. Linear Multilinear Algebra 18: 141–145

Arfken GB, Weber HJ (2000) Mathematical methods for physicists, 5th edn. Academic Press

Banerjee A, Dhillon IS, Ghosh J, Merugu S, Modha DS (2004) A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining (KDD '04), pp 509–514

Biggs N (1974) Algebraic graph theory. Cambridge University Press

Boley D, Gini M, Gross R, Han E-H, Hastings K, Karypis G, Kumar V, Mobasher B, Moore J (1999) Document categorization and query generation on the world wide web using webace. AI Rev 11:365–391

Cai R, Lu L, Hanjalic A (2005) Unsupervised content discovery in composite audio. In: Proceedings of the 13th annual ACM international conference on Multimedia (MM '05), pp 628–637

Cheeger J (1970) A lower bound for the smallest eigenvalue of the laplacian. In: Gunning RC (ed) Problems in Analysis. Princeton Univ. Press, pp 195–199

Chung FRK (1997) Spectral graph theory. American Mathematical Society

Demmel JW (1997) Applied numerical linear algebra. SIAM

Dhillon IS (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining (KDD)

Dhillon IS, Mallela S, Modha DS (2003) Information-theoretic co-clustering. In: Proceedings of ninth ACM SIGKDD international conference on knowledge discovery and data mining (KDD '03), pp 89–98

Ding CHQ (2003a) Document retrieval and clustering: from principal component analysis to self-aggregation networks. In: Proceedings of int'l parallel and distributed processing symposium proceedings of 9th int'l workshop on artificial intelligence and statistics

Ding CHQ (2003b) Unsupervised feature selection via two-way ordering in gene expression analysis. Bioinformatics 19:1259–1266

Ding CHQ, He X, Meraz RF, Holbrook SR (2004) A unified representation of multiprotein complex data for modeling interaction networks. Proteins: Struct Func Bioinform 57(1):99–108

Dodziuk J (1984) Difference equations, isoperimetric inequality and the transience of certain random walks. Trans Am Math Soc 284:787–794

Dodziuk J, Kendall WS (1986) Combinatorial laplacians and isoperimetric inequality. In: From local times to global geometry, control and physics. Pitman Research Notes in Mathematics Series 150:68–74, [Longman Scientific and Techical]

Donath WE, Hoffman AJ (1972) Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices. IBM Tehn Disclosure Bull 15:938–944

Donath WE, Hoffman AJ (1973) Lower bounds for the partitioning of graphs. IBM J Res Dev 17:420–425

Dongen SV (2000) Graph clustering by flow simulation. PhD thesis, University of Utrecht

Duda RO, Hart PE, Stork DG (2000) Pattern classification. Wiley

Enright AJ, Dongen SV, Ouzounis CA (2002) An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res 30(7):1575–1584

Fiedler M (1973) Algebraic connectivity of graphs. Czech Math J 23:298–305

Fiedler M (1975a) Eigenvectors of acyclic matrices. Czech Math J 25:607–618

Fiedler M (1975b) A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. Czech Math J 25:619–633

Fiedler M (1986) Special matrices and their applications in numerical mathematics. Martinus Nijhoff Publishers

Garey MR, Johnson DS (1979) Computers and intractability; a guide to the theory of NP-completeness. W. H. Freeman and Company

George T, Merugu S (2005) A scalable collaborative filtering framework based on co-clustering. In: Proceedings of the fifth IEEE international conference on data mining (ICDM '05)

Gilbert JR, Miller GL, Teng SH (1998) Geometric mesh partitioning: implementation and experiments. SIAM J Sci Comput 19(6):2091–2110

Golub GH, Van-Loan CF (1989) Matrix computations. John Hopkins Press

Gonzalez RC, Woods RE (2002) Digital image processing. Prentice Hall, Upper Saddle River

Grady L, Schwartz EL (2006a) Isoperimetric graph partitioning for image segmentation. IEEE Trans Pattern Anal Mach Intell 28(3):469–475

Grady L, Schwartz EL (2006b) Isoperimetric partitioning: A new algorithm for graph partitioning. SIAM J Sci Comput 27(6):1844–1866

Guattery S, Miller GL (1998) On the quality of spectral separators. SIAM J Matrix Anal Appl 19(3): 701–719

Hagen L, Kahng AB (1992) New spectral methods for ratio cut partitioning and clustering. IEEE Trans Comput Aid Design Integr Circuits Sys 11(9):1074–1085

Han E-H, Karypis G (2000) Centroid-based document classification: analysis and experimental results. In: Proceedings of 4th European conference on principles and practice of knowledge discovery in databases (PKDD '00), pp 424–431

Hendrickson B, Leland R (1995) The chaco user's guide. Technical Report SAND95-2344, Sandia National Laboratories, Albuquerque

Hersh W, Buckley C, Leone TJ, Hickam D (1994) Ohsumed: an interactive retrieval evaluation and new large test collection for research. In: Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR '94), pp 192–201

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci USA 79:2554–2558

Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323

Jolliffe IT (2002) Principal component analysis, 2nd edn. Springer, New York

Kuijlaars ABJ (2001) Which eigenvalues are found by the Lanczos method. SIAM J Matrix Anal Appl 22(1):306–321

Kumar R, Mahadevan U, Sivakumar D (2004) A graph-theoretic approach to extract storylines from search results. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining (KDD '04), pp 216–225

Kummamuru K, Dhawale A, Krishnapuram R (2003) Fuzzy co-clustering of documents and keywords. In: Proceedings of The 12th IEEE international conference on fuzzy systems (FUZZ '03), pp 772–777

Lewis DD (1999) Reuters-21578 text categorization test collection distribution 1.0, http://www.daviddlewis.com/resources/testcollections/reuters21578/

Long B, Zhang Z, Yu PS (2005) Co-clustering by block value decomposition. In: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining (KDD '05), pp 635–640

Mandhani B, Joshi S, Kummamuru K (2003) A matrix density based algorithm to hierarchically co-cluster documents and words. In: Proceedings of the 12th international conference on World Wide Web (WWW '03), pp 511–518

Merris R (1994) Laplacian matrices of graphs: a survey. Linear Algebra Appl 197:143–176

Mohar B (1989) Isoperimetric numbers of graphs. J Comb Theory Ser B 47:274–291

Mohar B (1991) The Laplacian spectrum of graphs. Graph Theory Comb Appl 2:871–898

Oh C-H, Honda K, Ichihashi H (2001) Fuzzy clustering for categorical multivariate data. In: Proceedings of joint 9th IFSA world congress and 20th NAFIPS international conference, pp 2154–2159

Porter MF (1980) An algorithm for suffix stripping. Program 14(3):130–137

Qiu G (2004) Image and feature co-clustering. In: Proceedings of IEEE ICPR

Rege M, Dong M, Fotouhi F (2006a) Co-clustering documents and words using bipartite isoperimetric graph partitioning. In: Proceedings of the 6th IEEE international conference on data mining (ICDM)

Rege M, Dong M, Fotouhi F (2006b) Co-clustering image features and semantic concepts. In: Proceedings of IEEE international conference on image processing

Rui Y, Huang TS, Mehrotra S (1997) Content-based image retrieval with relevance feedback in mars. In: Proceedins of IEEE International conference on image processing

Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8): 888–905

Simon HD (1991) Partitioning of unstructured problems for parallel processing. Comput Syst Eng 2:135–148

Slonim N, Tishby N (2000) Document clustering using word clusters via the information bottleneck method. In: Research and development in information retrieval, pp 208–215

Smeulders AWM, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380

TREC (1996, 1997, 1998) Text retrieval conference, http://trec.nist.gov

Wu X, Ngo CW, Li Q (2005) Co-clustering of time-evolving news story with transcript and keyframe. In: Proceedings of IEEE international conference on multimedia and expo (ICME ’05), pp 117–120

Zha H, He X, Ding CHQ, Simon H, Gu M (2001) Bipartite graph partitioning and data clustering. In: Proceedings of the tenth international conference on information and knowledge management (CIKM)

Zha H, Ji X (2002) Correlating multilingual documents via bipartite graph modeling. In: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR ’02)

Zhao R, Grosky WI (2002) Narrowing the semantic gap-improved text-based web document retrieval using visual features. IEEE Trans Multimedia 4(2):189–200