Co-clustering documents and words using Bipartite Isoperimetric Graph Partitioning

Manjeet Rege Ming Dong
Machine Vision and Pattern Recognition Lab

Farshad Fotouhi
Database and Multimedia Systems Group

Department of Computer Science, Wayne State University
Detroit, MI 48202, USA

Abstract

In this paper, we present a novel graph theoretic approach to the problem of document-word co-clustering. In our approach, documents and words are modeled as the two vertices of a bipartite graph. We then propose Isoperimetric Co-clustering Algorithm (ICA) - a new method for partitioning the document-word 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 extensive experiments performed on publicly available datasets demonstrate the advantages of ICA over spectral approach in terms of the quality, efficiency and stability in partitioning the document-word bipartite graph.

1 Introduction

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. The entire data set is typically represented by a *word-document* matrix **B** where rows of the matrix denote the words in the collection while columns represent the documents. The words are treated as features and documents are represented as a vector such that an entry B_{ij} in the matrix signifies the relevance of word i for document j. Co-clustering of documents and words is performed by deriving sub-matrices by simultaneously clustering rows and columns of **B**.

Although much of the earlier research was focussed on performing document and word clustering separately, co-clustering of documents and words has been a topic of much interest in recent years because of its applications to problems arising in text, Web and multimedia documents. Dhillon et al. [1] defined co-clustering of documents and words by a pair of maps from rows to row-clusters (words) and from columns to column-clusters (documents) 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. Long et al. [2] factorize the word-document matrix **B** 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 **B**. In [3], 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. [4] 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.

In this paper, we model document-word co-clustering as a bipartite graph partitioning problem. Although similar approach has been adopted before by others, the main contribution of this work lies in a new algorithm that we propose - **I**soperimetric **C**o-clustering **A**lgorithm (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. Our work bears resemblance to the popular spectral heuristical approach for partitioning bipartite graphs - in the sense that 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, the proposed algorithm requires a simple solution to a sparse system of linear equations instead of the eigenvalue or SVD problem in spectral co-clustering. Our extensive experiments performed on publicly available datasets demon-

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

strate the advantages of our approach over spectral approach in terms of the quality, efficiency and stability in partitioning the document-word bipartite graph.

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 V=$\{v_1, v_2, ..., v_{|V|}\}$ and a set of edges 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,*

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

DEFINITION 2 *The degree of a vertex v_i denoted by d_i is,*

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

DEFINITION 3 *The degree matrix* **D** *of the graph is a diagonal matrix such that,*

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

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

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

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,*

$$cut(V_1, V_2) = \sum_{i \in V_1, j \in V_2} A_{ij} \tag{5}$$

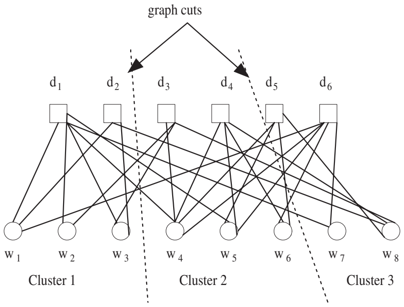

Figure 1. The square and circular vertices denote the two kinds of vertices in the bipartite graph that are used to represent words and documents. Co-clustering of documents and words is achieved by partitioning this bipartite graph.

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

$$cut(V_1, V_2, ..., V_k) = \sum_{n<\theta} cut(V_n, V_\theta) \tag{6}$$

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. Spectral graph partitioning uses eigenvectors of **L** to compute the indicator vector and has been one of the most popular and widely applied methods. It is based on the early work of Fiedler [5] who associated λ_2, the second smallest eigenvalue of **L** with connectivity of graph and suggested partitioning by dividing vertices according to their value in the corresponding eigenvector $\mathbf{u}_2 = \{u_1, u_2, ..., u_{|V|}\}$. 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 vertices i such that $u_i > s$ and the set of vertices such that $u_i \leq s$. Shi and Malik applied spectral graph partitioning to image segmentation in [6] by using the eigenvector corresponding to the second smallest eigenvalue of the generalized eigenvalue problem [7],

$$\mathbf{Lx} = \lambda \mathbf{D}\ \mathbf{x} \tag{7}$$

2.2 Bipartite graphs

An undirected bipartite graph *G* =$\{$*D*, *W*, *E*$\}$, has two sets of vertices, viz., *D* and *W* and a set of graph edges *E*. Let **B** be an m by n graph weight matrix, where n and m represent the number of vertices in *D* and *W*, respectively. An entry B_{ij} in this matrix is the weight of an edge appearing between a vertex $w_i \in$ *W* and a vertex $d_j \in$ *D*. There are no edges between vertices of the same group. Then, the adjacency matrix **A** of the bipartite graph is expressed as,

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \tag{8}$$

where the first m vertices index *W* and the last n index *D*. A degree vector **d** of the bipartite graph is,

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_W \\ \mathbf{d}_D \end{bmatrix} \tag{9}$$

where $\mathbf{d}_W$ and $\mathbf{d}_D$ are vectors consisting of degrees of *W* and *D* vertices, respectively. The degree matrix **D** and Laplacian **L** are,

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_W & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_D \end{bmatrix} \tag{10}$$

$$\mathbf{L} = \begin{bmatrix} \mathbf{D}_W & \textbf{-B} \\ \textbf{-B}^T & \mathbf{D}_D \end{bmatrix} \tag{11}$$

where $D_W(i,i) = \sum_j B_{ij}$ and $D_D(j,j) = \sum_i B_{ij}$

If we let *W* be the set of words and *D* be the set of documents then we can represent the documents and words by the two vertices of the weighted bipartite graph. Co-clustering is then achieved by partitioning the bipartite graph. In Figure 1, we show the bipartite graph partitioned using dotted lines. In order to compute these partitions, we also need to solve a generalized eigenvalue problem as in Equation (7). However, due to the bipartite nature of the problem, the eigenvalue problem reduces to a much efficient Singular Value Decomposition (SVD) [7] problem and has been widely applied in many co-clustering papers. Dhillon [8] and Zha et al., [9] employed this Spectral-SVD approach to partition a bipartite graph of documents and words. In [10], the two types of vertices of bipartite graph are used to represent sentences of documents of two different languages. The Spectral-SVD method is then 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 co-clustering multimedia documents (images/videos) and visual keywords (features). In [11], it has been used to co-cluster a bipartite graph of user relevance feedback logs and low-level image features. Wu et al. [12] use the same algorithm 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 the next section, we derive the proposed algorithm for co-clustering documents and words using weighted bipartite graphs and show that our algorithm requires a simple solution to a sparse system of linear equations to partition the bipartite graph.

3 Isoperimetric Co-clustering Algorithm to co-cluster documents and words

In the co-clustering of documents and words (Fig 1), clustering of documents induces clustering of words and vice-versa. Let us denote the document clusters as $D_1, D_2, ..., D_k$ and the clusters of words with $W_1, W_2, ..., W_k$. The basic premise of our algorithm is that, if w_i belongs to a cluster, say W_p, where $1 \leq p \leq k$, then its association with D_p is greater than its association with any other D cluster. From a graph theoretic point of view, the association between w_i and the D clusters can be expressed in terms of the sum of the edge weights. Thus,

$$W_p = \{w_i : \sum_{j \in D_p} B_{ij} \geq \sum_{j \in D_l} B_{ij}, \forall\ l = 1, ..., k\} \tag{12}$$

where the matrix **B** is as defined in Equation (8). Similarly, for d_j belonging to cluster D_p, the following should hold,

$$D_p = \{d_j : \sum_{i \in W_p} B_{ij} \geq \sum_{i \in W_l} B_{ij}, \forall\ l = 1, ..., k\} \tag{13}$$

The algorithm presented here has been motivated from the combinatorial formulation of the classic isoperimetric problem [13–16]: *For a fixed area, find the shape with minimum perimeter*. We present a polynomial time heuristic for the NP-hard 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 [17],*

$$\phi = \inf_F \frac{|\triangle F|}{Vol_F} \ , \tag{14}$$

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 ={D, W, E} is defined as [15],*

$$\phi(G) = \inf_F \frac{|\triangle F|}{Vol_F} \tag{15}$$

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

$$Vol_F \leq \frac{1}{2} Vol_{\{D \bigcup W\}} \tag{16}$$

Since, the number of documents and words are finite, the bipartite graph has finite number of vertices. Hence, the infimum in Equation (15) becomes a minimum. The boundary $\triangle F$ of the set F can be expressed as, $\triangle F = \{e_{ij}|$ edges between a vertex in F and its complement $F^C\}$

Since the bipartite graph of documents and words is weighted, we define,

$$|\triangle F| = \sum_{e_{ij} \in \triangle F} w(e_{ij}) \tag{17}$$

The combinatorial volume [13, 14] can be defined in the following two ways:

$$Vol_F = |F| \tag{18}$$

or,

$$Vol_F = \sum_i d_i, \forall \ vertices \in F \tag{19}$$

Equation (18) defines the volume in terms of the number of vertices in the bipartite subset *F* while Equation (19) defines it in terms of the sum of the edge weights incident on each of the vertices enclosed in *F*. To co-cluster documents and words, 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 Equation (19) 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 Equation (16) 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 an indicator vector **x** for words as follows,

$$x_i = \begin{cases} 0, & \text{if } w_i \in F \\ 1, & \text{if } w_i \notin F \end{cases} \tag{20}$$

and an indicator vector **y** for documents as,

$$y_i = \begin{cases} 0, & \text{if } d_i \in F \\ 1, & \text{if } d_i \notin F \end{cases} \tag{21}$$

For simplicity, we combine the two indicator vectors into a single indicator **z** as,

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \tag{22}$$

Binary values assigned to first m vertices of **z** indicate the partitioning of words. Similarly, partitioning of documents is inferred from the next n vertices in **z**.

From Equations (11), (17), (20), (21) and (22), we can express $|\triangle F|$ as,

$$\begin{aligned} |\triangle F| &= \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}^T \mathbf{L} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \\ &= \mathbf{z}^T\mathbf{L}\mathbf{z} \end{aligned} \tag{23}$$

Also, using Equations (9), (19) and (22), we can write Vol_F as,

$$Vol_F = \mathbf{z}^T\mathbf{d} \tag{24}$$

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 on the denominator as,

$$\mathbf{z}^T\mathbf{d} = c, \tag{25}$$

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

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

$$\phi(G) = \min_{\mathbf{z}} \frac{\mathbf{z}^T\mathbf{L}\mathbf{z}}{\mathbf{z}^T\mathbf{d}} \tag{26}$$

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

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

$$\mathcal{C} = \mathbf{z}^T\mathbf{L}\mathbf{z} - \rho(\mathbf{z}^T\mathbf{d} - c) \tag{27}$$

where ρ is a Lagrange multiplier.

As $\mathbf{L}$ is positive semi-definite [18] and $\mathbf{z}^T\mathbf{d}$ is non-negative, $\mathcal{C}$ will be at minimum at its critical point. Differentiating Equation (27) with respect to **z**, we get,

$$\frac{d\mathcal{C}}{d\mathbf{z}} = 2\mathbf{L}\mathbf{z} - \rho\,\mathbf{d} \tag{28}$$

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,

$$\begin{aligned} \mathbf{L}\mathbf{z} &= \mathbf{d} \\ \begin{bmatrix} \mathbf{D}_W & \textbf{-B} \\ \textbf{-B}^T & \mathbf{D}_D \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} &= \begin{bmatrix} \mathbf{d}_W \\ \mathbf{d}_D \end{bmatrix} \end{aligned} \tag{29}$$

Equation (29) is system of linear equations with $m + n$ number of equations with as many number of variables. Also, the matrix **L** is singular, i.e. its determinant is zero. So, this is a singular system of linear equations and hence does not have a unique solution [7].

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. We illustrate in Section 3.1 that it does not matter whether a document or a word vertex is removed, as long as the removed vertex is densely connected to 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,

$$\begin{aligned} \mathbf{L}^*\mathbf{z}^* &= \mathbf{d}^* \\ \begin{bmatrix} \mathbf{D}_W & \textbf{-B} \\ \textbf{-B}^T & \mathbf{D}_D \end{bmatrix}^* \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}^* &= \begin{bmatrix} \mathbf{d}_W \\ \mathbf{d}_D \end{bmatrix}^* \end{aligned} \tag{30}$$

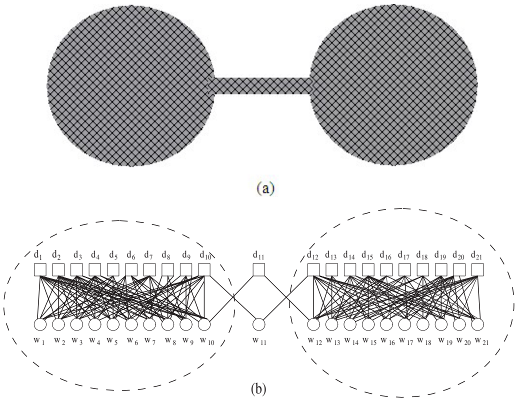
(a)
d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10}
d_{11}
d_{12} d_{13} d_{14} d_{15} d_{16} d_{17} d_{18} d_{19} d_{20} d_{21}
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10}
w_{11}
w_{12} w_{13} w_{14} w_{15} w_{16} w_{17} w_{18} w_{19} w_{20} w_{21}
(b)

Figure 2. (a) Cheeger's dumbbell-shaped 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 document and word vertices

Solving Equation (30) 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 Section 2) to convert it to a binary vector as per Equations (20) and (21). Amongst the common methods for cutting the indicator vector are the median cut and the ratio cut [19]. 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 partitions to gain document-word clusters 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 [20] suggesting that vertex with the maximum degree should be removed. However, in the document-word bipartite graph it is not clear whether we should remove the document vertex or the word vertex or is it that it does not matter which one we remove. That is, if we go with the heuristic of removing the maximum degree vertex then should we calculate the maximum across both document and word vertices together or should we remove a maximum degree vertex from within one of the two sets of vertices? We analyze this using the dumbbell shaped graph (Figure 2a) which was discussed in [17] on the relationship of the isoperimetric constant and the eigenvalues of the Laplacian on continuous manifolds. We constructed a bipartite dumbbell graph (Figure 2b) consisting of 21 document and word vertices each with uniform weights. 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. In Figure 3, we show 6 cases where different document and word vertices are removed to solve the system of linear equations and the corresponding partitioning achieved. We removed document and word vertices that were densely connected (i.e. high degree) and vertices that were sparsely connected (like d_{11} and w_{11}). 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 (Figure 3 a, 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 (Figure 3 c, d). In the experiments (Section 4), we removed vertex with maximum degree, minimum degree and also randomly chose a vertex to be removed. Our results show that the algorithm 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 word-document matrix **B**, construct **d** and **L** using Equations (9) 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 Equation (30) for the indicator vector **z** defined by Equation (22).
4. Bipartition **z** using ratio cut to get two document-word 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.

3.3 Advantages over Spectral approach

Although spectral methods in graph partitioning have been popular and successfully applied to diverse research problems, it does suffer from some significant drawbacks. In [21], 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

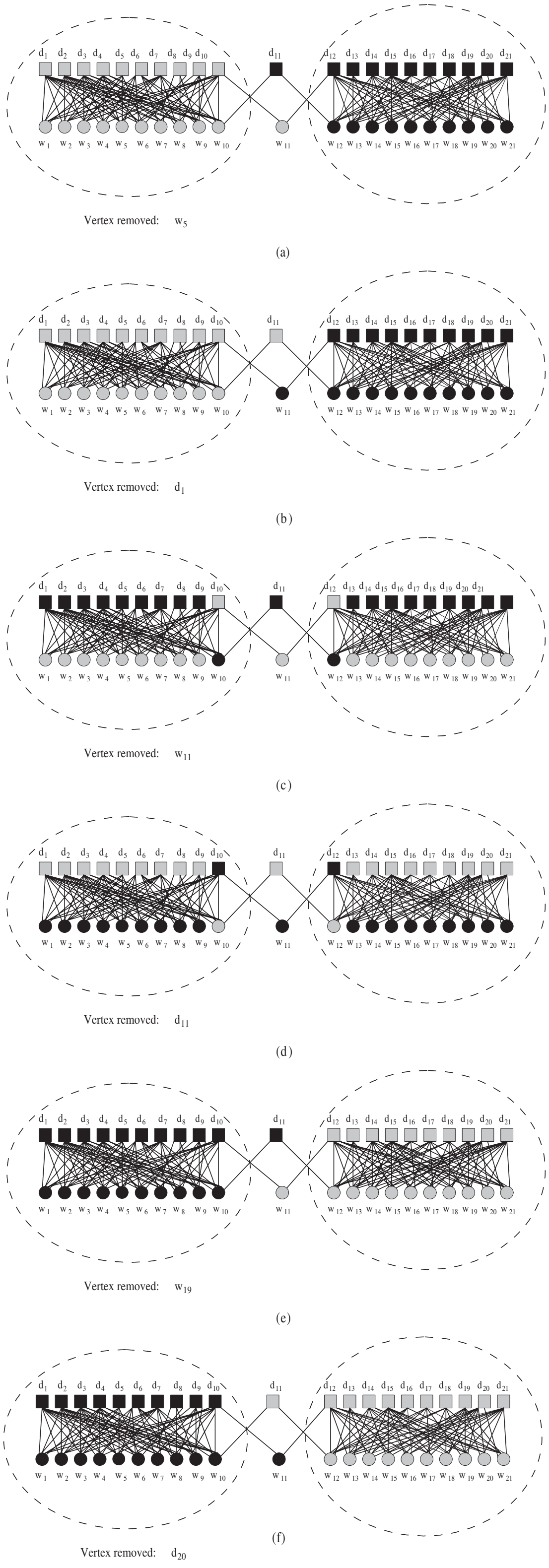
Vertex removed: w_5
(a)
Vertex removed: d_1
(b)
Vertex removed: w_{11}
(c)
Vertex removed: d_{11}
(d)
Vertex removed: w_{19}
(e)
Vertex removed: d_{20}
(f)
d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10} d_{11} d_{12} d_{13} d_{14} d_{15} d_{16} d_{17} d_{18} d_{19} d_{20} d_{21}
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10} w_{11} w_{12} w_{13} w_{14} w_{15} w_{16} w_{17} w_{18} w_{19} w_{20} w_{21}

Figure 3. The 6 cases show the partitioning achieved when different document and word vertices from the dumbbell graph are removed to solve the system of linear equations

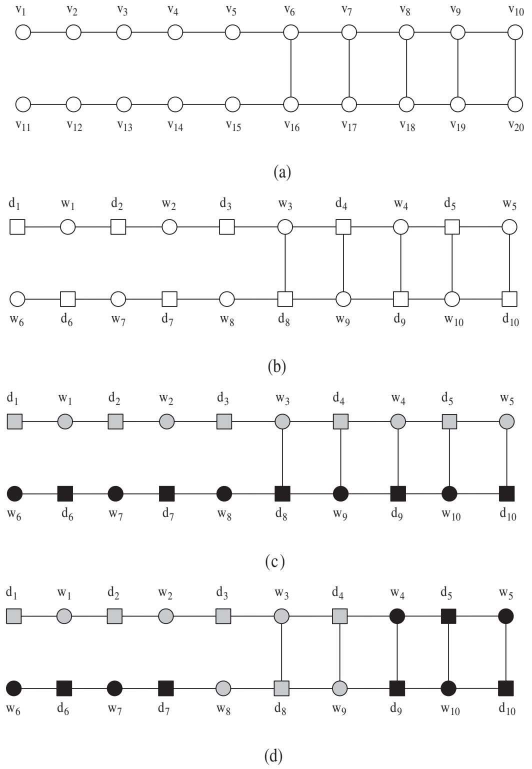
v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10}
v_{11} v_{12} v_{13} v_{14} v_{15} v_{16} v_{17} v_{18} v_{19} v_{20}
(a)
d_1 w_1 d_2 w_2 d_3 w_3 d_4 w_4 d_5 w_5
w_6 d_6 w_7 d_7 w_8 d_8 w_9 d_9 w_{10} d_{10}
(b)
(c)
(d)

Figure 4. (a) *Roach* graph for $k = 5$. (b) Bipartite roach graph for $k = 5$ with documents and words vertices. (c) Solution with Spectral-SVD algorithm. Isoperimetric ratio = 0.2174. (d) Solution with ICA. Isoperimetric ratio = 0.1304.

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 Figure 4a. We constructed a bipartite roach graph of document and word vertices for $k = 5$, shown in Figure 4b. In Figures 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 [21] for homogeneous graphs. With this example, we have been able to show that our algorithm is able to perform well on a category of graphs that spectral methods are not able to partition efficiently.

The proposed algorithm 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 [8, 9]. The Lanczos algorithm [7] 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 [22]. 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 [7].

3.4 Time Complexity

Computational time of the proposed algorithm depends on solution to Equation (30). In particular, the time complexity is dependent on the number of non-zero entries in **L**, which asymptotically is $O(|E|)$. We can solve Equation (30) 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 Equation (30) 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 $h = m + n$. 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 $O(h\,logh)$.

4 Experiments and Results

In this Section, we empirically compare ICA with the popular Spectral-SVD bipartite graph partitioning approach [8, 9]. Section 4.1 discusses the evaluation method used to report the experimental results. In Section 4.2, we present the results on co-clustering documents and words using some of the publicly available datasets. We also compare the computational speed of ICA and Specral-SVD in Section 4.3.

4.1 Evaluation Methodology

Traditional clustering reporting technique such as a confusion matrix has been used earlier to present co-clustering results, specially document-word co-clustering in [1, 8, 9]. 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 Sections 1 and 3, clustering of words induces clustering of the documents and vice-versa. The goal of co-clustering is NOT to achieve perfect clustering of one data type but to achieve the 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.

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 [5, 17]. Infact the goals of both ICA and the Spectral-SVD algorithms is to minimize the isoperimetric ratio. ICA achieves this by solving a system of linear equations while Spectral-SVD minimizes the ratio by solving the eigenvalue or the SVD problem. Hence, isoperimetric ratio can naturally be used to evaluate the goodness of co-clustering. ICA and Spectral-SVD algorithms are compared in terms of the isoperimetric ratio by employing the ratio cut to find the optimal partition (i.e. not necessarily partitions of equal size).

4.2 Documents and words co-clustering

We have primarily utilized the dataset used in [23][1]. Summary of this dataset is shown in Table II. Data sets oh0, oh5, oh10 and oh15 are from OHSUMED collection [24]. Data sets re0 and re1 are from Reuters- 21578 text categorization test collection Distribution 1.0 [25]. Data set wap is from the WebACE project (WAP) [26]. 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 [2]. We also used the Medline (1033 medical abstracts),and Cranfield (1400 aeronautical systems abstracts) dataset[3].

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

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

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

[2] http://trec.nist.gov

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

Table I. Summary of the datasets used for bipartitioning documents and words

Dataset	No. of words	No. of docs
Med-Cran	9181	2431
Eng-Heart	2504	375
Graft-Phos	2432	293
ArachidonicAcids-Hematocrit	2353	274
Enzyme-Infections	2528	311
Interest-Trade	2682	538

2. Classes England and Heart-Valve-Prosthesis from oh0 dataset were mixed to create Eng-Heart.
3. Graft-Survival and Phospholipids from oh5 were mixed to form Graft-Phos.
4. ArachidonicAcids-Hematocrit was derived from oh10 using Arachidonic Acids and Hematocrit classes.
5. 2 classes from oh15, viz. Enzyme Activation and Staphylococcal Infections were combined to form the Enzyme-Infections dataset.
6. Interest-Trade was formed by mixing Interest and Trade classes of re0 dataset.

Table II. 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

Table III. Isoperimetric Ratio of ICA (max,min & random vertex removal) and SpecSVD for bipartitioning all the datasets.

Dataset	ICAMaxVR	ICAMinVR	ICARanVR	SpecSVD
M-C	0.1485	0.1647	0.1415	0.2281
E-H	0.3036	0.3316	0.2907	0.4142
G-P	0.2650	0.3541	0.2948	0.4033
A-H	0.2455	0.2611	0.3158	0.4221
E-I	0.1974	0.2821	0.3340	0.2891
I-T	0.2872	0.3971	0.2768	0.4652

Table IV. Mean Isoperimetric Ratios of ICA and SpecSVD in k-partitioning all the datasets.

Dataset	ICA	SpecSVD
oh0	0.1867	0.3247
oh5	0.1870	0.4418
oh10	0.2255	0.3818
oh15	0.1816	0.4359
re0	0.2015	0.4271
re1	0.1516	0.3509
wap	0.2804	0.4210
tr11	0.2817	0.4605
tr12	0.2558	0.4508
tr21	0.3356	0.5248
tr23	0.2769	0.4832
tr31	0.2622	0.4386
tr41	0.2756	0.46
tr45	0.2189	0.4759

In Section 3.1, we discussed that removing the vertex with maximum degree from the bipartite graph is the best choice to solve Equation (29). We now show this empirically by comparing three strategies of removing the maximum degree, minimum degree and a random vertex from the bipartite graph. 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 6 bipartitioning datasets are shown in Table III. We have abbreviated dataset names due to space constraints. Maximum vertex removal is denoted by ICAMaxVR, minimum by ICAMinVR, random by ICARanVR and the Spectral-SVD by SpecSVD. As can be seen, ICAMinVR yields partitions with comparatively high isoperimetric ratios than the other cases. 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 graph example (Figure 3 c,d). ICARanVR can sometimes slightly outperform ICAMaxVR (*Med-Cran*, *Eng-Heart*, *Interest-Trade*). However, due to the randomness associated with it, it can actually end up being ICAMinVR and perform poorly at times (*Graft-Phos, ArachidonicAcids-Hematocrit, Enzyme-Infections*). As a result, ICARanVR lacks in consistency in terms of guranteed optimal partitioning of the bipartite graph. Moreover, the difference in ratios of ICAMaxVR and ICARanVR when ICARanVR does outperform is very negligible. For the rest of the experiments, we have employed ICAMaxVR and is referred to as ICA from now on.

In Figure 5a, the sparsity pattern of a typical word-document matrix (*Med-Cran* in the figure) is shown. 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

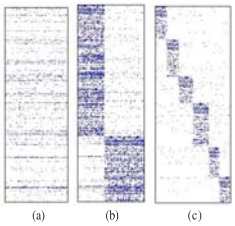

Figure 5. (a) Sparseness of typical word-document matrix (shown here *Med-Cran*) before co-clustering (b) *Med-Cran* matrix after bipartitioning (c) *tr21* matrix after k-partitioning

Figure 5b.

We performed k-partitioning on the datasets mentioned in Table II by recursively applying ICA and SpecSVD algorithms. Since, we get an isoperimetric ratio for every bipartition, for k-partitioning comparison, we calculated the Mean Isoperimetric Ratio. These results are shown in Table IV. As is evident, ICA consistently outperforms SpecSVD on all the datasets. Figure 5c shows the *tr21* word-document matrix after co-clustering (i.e. k-partitioning with k=6).

Table V. Mean Standard deviation of the Isoperimetric Ratios of ICA and SpecSVD over all the noisy datasets.

Experiment	ICA	SpecSVD
Bipart. with add. noise	0.668×10^{-2}	1.41×10^{-2}
Bipart. with mult. noise	0.43×10^{-2}	0.534×10^{-2}
k-part. with add. noise	1.1×10^{-2}	1.88×10^{-2}
k-part. with mult. noise	1.27×10^{-2}	1.53×10^{-2}

Performance in the presence of Noise: Unlike ICA, that solves a sparse system of linear equations, the spectral approach requires solution to an eigenvalue or SVD problem. With reference to the discussion in Section 3.3, SpecSVD might not be stable in the presence of noise. Also, it would be interesting to see whether ICA outperforms SpecSVD in the presence of different kinds of noises. To evaluate this, we compared the performance of the two algorithms in the presence of Gaussian additive and multiplicative noise. Additive noise had zero mean with 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. We have shown representative behavior of the two algorithms in Figure 6. First two plots are bipartitioning in the presence of additive noise on *Interest-Trade* and multiplicative noise on *ArachidonicAcids-Hematocrit*. Similarly, next two are for k-partitioning with additive noise on *wap* and multiplicative on *re0* datasets. From these results, we can 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 instability and inconsistency to partition optimally. To demonstrate the stability of both the algorithms empirically, 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 V. Higher standard deviation of SpecSVD indicates instability in partitioning the noisy datasets.

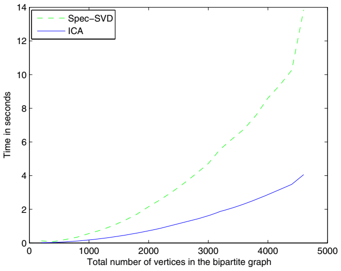

Figure 7. Computational speed comparison for ICA and SpecSVD. The time required by each of the algorithms to compute the indicator vectors are displayed for varying number of vertices in the bipartite graph.

4.3 Computational speed comparison

We now compare the computational speed of ICA with SpecSVD. The time complexity for both the 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 where every vertex of one type is connected with all the vertices of the other type. Since, the time required to cut the indicator vector is same for both the algorithms, we compare on the basis of the time required to calculate the indicator vector. This experiment was performed on a machine with a 3 GHz Intel Pentium 4 processor with 1 GB RAM. In Figure 7, we plot the time required by both the algorithms as the number of vertices in the fully connected bipartite graph increase. Time for ICA gradually increases as the number of vertices increase. However, SpecSVD time increases more rapidly comparatively and hence is clearly outperformed by ICA.

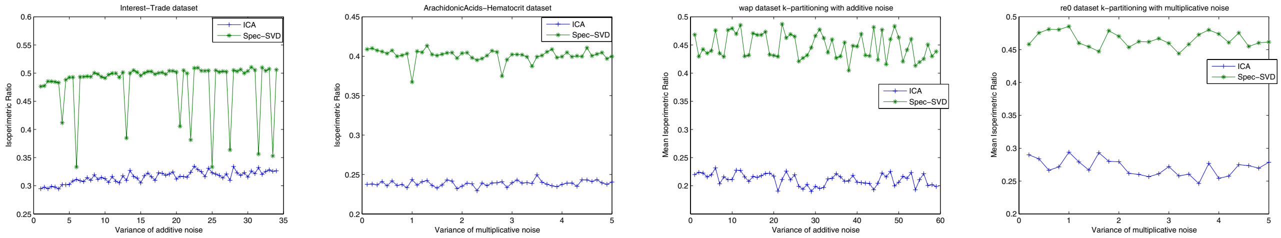

Figure 6. First two figures are bipartitioning with additive noise (*Interest-Trade*) and multiplicative noise (*ArachidonicAcids-Hematocrit*). Next two are *k*-partitioning with additive (*wap*) and multiplicative noise (*re0*)

5 Conclusions

We proposed the Isoperimetric Co-clustering Algorithm - a new method for partitioning the document-word bipartite graph. The proposed algorithm requires a solution to a sparse system of linear equations. Experiments performed demonstrate the advantages of our approach over spectral approach in terms of the quality, efficiency and stability in partitioning the document-word bipartite graph.

References

[1] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *proc. ACM SIGKDD 2003*.

[2] B. Long, Z. Zhang, and P. S. Yu, "Co-clustering by block value decomposition," in *proc. ACM SIGKDD*, 2005.

[3] N. Slonim and N. Tishby, "Document clustering using word clusters via the information bottleneck method," in *Research and Development in IR*, 2000, pp. 208–215.

[4] B. Mandhani, S. Joshi, and K. Kummamuru, "A matrix density based algorithm to hierarchically co-cluster documents and words," in *proc. WWW, 2003*.

[5] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematics Journal*, vol. 23, pp. 298–305, 1973.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on PAMI*, vol. 22, 2000.

[7] G. H. Golub and C. F. Van-Loan, *Matrix Computations*, John Hopkins Press, 1989.

[8] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *ACM SIGKDD*, 2001.

[9] H.Zha, X.He, C.H.Q.Ding, H.Simon, and M.Gu, "Bipartite graph partitioning and data clustering," in *CIKM*, 2001.

[10] H. Zha and X. Ji, "Correlating multilingual documents via bipartite graph modeling," in *proc. of ACM SIGIR*, 2002.

[11] M. Rege, M. Dong, and F. Fotouhi, "Co-clustering image features and semantic concepts," in *IEEE ICIP 2006*.

[12] X.Wu, C.W.Ngo, and Q.Li, "Co-clustering of time-evolving news story with transcript and keyframe," in *IEEE ICME*, 2005.

[13] J. Dodziuk, "Difference equations, isoperimetric inequality and the transience of certain random walks," *Trans. of the American Mathematical Society*, vol. 284, 1984.

[14] J. Dodziuk and W. S. Kendall, "Combinatorial laplacians and isoperimetric inequality," *From Local Times to Global Geometry, Control and Physics of Pitman Research Notes in Mathematics Series, Longman Scientific and Techical*, vol. 150, pp. 68–74, 1986.

[15] B. Mohar, "Isoperimetric numbers of graphs," *Journal of Combinatorial Theory, Series B*, vol. 47, pp. 274–291, 1989.

[16] L. Grady and E. L. Schwartz, "Isoperimetric graph partitioning for image segmentation," *IEEE Trans. on PAMI*, vol. 28, no. 3, pp. 469– 475, 2006.

[17] J. Cheeger, "A lower bound for the smallest eigenvalue of the laplacian," *Problems in Analysis*, pp. 195–199, 1970.

[18] M. Fiedler, *Special Matrices and Their Applications in Numerical Mathematics*, Martinus Nijhoff Publishers, 1986.

[19] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 11, no. 9, 1992.

[20] W. N. Anderson and T. D. Morley, "Eigenvalues of the laplacian of a graph," *Linear and Multilinear Algebra*, vol. 18, pp. 141–145, 1985.

[21] S. Guattery and G. L. Miller, "On the quality of spectral separators," *SIAM Journal on Matrix Analysis and Applications*, vol. 19, no. 3, pp. 701–719, 1998.

[22] A. B. J. Kuijlaars, "Which eigenvalues are found by the Lanczos method?," *SIAM Journal on Matrix Analysis and Applications*, vol. 22, no. 1, pp. 306–321, 2001.

[23] E-H. Han and G. Karypis, "Centroid-based document classification: Analysis and experimental results," in *proc. of PKDD*, 2000.

[24] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam, "Ohsumed: an interactive retrieval evaluation and new large test collection for research," in *proc. of ACM SIGIR*, 1994.

[25] D. D. Lewis, "Reuters-21578 text categorization test collection distribution 1.0, http://www.research.att.com/ lewis," 1999.

[26] D. Boley, M. Gini, R. Gross, E-H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore, "Document categorization and query generation on the world wide web using webace," *AI Review*, vol. 11, pp. 365391, 1999.