Co-clustering by Block Value Decomposition

Bo Long
Computer Science Dept.
SUNY Binghamton
Binghamton, NY 13902
blong1@binghamton.edu

Zhongfei (Mark) Zhang
Computer Science Dept.
SUNY Binghamton
Binghamton, NY 13902
zzhang@binghamton.edu

Philip S. Yu
IBM Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
psyu@us.ibm.com

ABSTRACT

Dyadic data matrices, such as co-occurrence matrix, rating matrix, and proximity matrix, arise frequently in various important applications. A fundamental problem in dyadic data analysis is to find the hidden block structure of the data matrix. In this paper, we present a new co-clustering framework, block value decomposition(BVD), for dyadic data, which factorizes the dyadic data matrix into three components, the row-coefficient matrix **R**, the block value matrix **B**, and the column-coefficient matrix **C**. Under this framework, we focus on a special yet very popular case – non-negative dyadic data, and propose a specific novel co-clustering algorithm that iteratively computes the three decomposition matrices based on the multiplicative updating rules. Extensive experimental evaluations also demonstrate the effectiveness and potential of this framework as well as the specific algorithms for co-clustering, and in particular, for discovering the hidden block structure in the dyadic data.

Categories and Subject Descriptors

E.4 [**Coding and Information Theory**]: Data compaction and compression; H.3.3 [**Information search and Retrieval**]: Clustering; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms

Keywords

Co-clustering, Clustering, Matrix Decomposition, Dyadic Data, Hidden Block Structure, Block Value Decomposition (BVD), Non-negative Block Value Decomposition (NBVD).

1. INTRODUCTION

The clustering procedure arises in many disciplines and has a wide range of applications. In many applications, such as document clustering, collaborative filtering, and microarray analysis, the data can be formulated as a two-dimensional matrix representing a set of dyadic data. Dyadic data refer to a domain with two finite sets of objects in which observations are made for *dyads*, i.e., pairs with one element from either set. For the dyadic data in these applications, co-clustering both dimensions of the data matrix simultaneously is often more desirable than traditional one-way clustering. This is due to the fact that co-clustering takes the benefit of exploiting the duality between rows and columns to effectively deal with the high dimensional and sparse data that is typical in many applications. Moreover, there is an additional benefit for co-clustering to provide both row clusters and column clusters at same time. For example, we may be interested in simultaneously clustering genes and experimental conditions in bioinformatics applications [4, 5], simultaneously clustering documents and words in text mining [8], simultaneously clustering users and movies in collaborative filtering.

In this paper, we propose a new co-clustering framework called Block Value Decomposition (BVD). The key idea is that the latent block structure in a two-dimensional dyadic data matrix can be explored by its triple decomposition. The dyadic data matrix is factorized into three components, the row-coefficient matrix **R**, the block value matrix **B**, and the column-coefficient matrix **C**. 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.

Under this framework, we develop a specific novel co-clustering algorithm for a special yet very popular case – non-negative dyadic data, that iteratively computes the three decomposition matrices based on the multiplicative updating rules derived from an objective criterion. By intertwining the row clusterings and the column clusterings at each iteration, the algorithm performs an implicitly adaptive dimensionality reduction, which works well for typical high-dimensional and sparse data in many data mining applications. We have proven the correctness of the algorithm by showing that the algorithm is guaranteed to converge and have conducted extensive experimental evaluations to demonstrate the effectiveness and potential of the framework and the algorithms.

We define the following notations in this paper. Capital-boldface letters such as **R**, **B**, and **C** denote matrices; small-boldface letters such as **r**, **b**, and **c** denote column vectors; lower-case letters such as w denote scalars.

KDD'05, August 21–24, 2005, Chicago, Illinois, USA.
Copyright 2005 ACM 1-59593-135-X/05/0008 ...$5.00.

2. RELATED WORK

This work is primarily related to two main areas: co-clustering in data mining and matrix decomposition in matrix computation.

Although most of the clustering literature focuses on one-sided clustering algorithms, recently co-clustering has become a topic of extensive interest due to its applications to many problems such as gene expression data analysis [4, 5] and text mining [8]. A representative early work of co-clustering was reported in [11] that identified hierarchical row and column clustering in matrices by a local greedy splitting procedure. The BVD framework proposed in this paper is based on the partitioning-based co-clustering formulation first introduced in [11].

Information-theory based co-clustering has attracted intensive attention in the literature. The information bottleneck (IB) framework [16] was first introduced for one-sided clustering. Later, an agglomerative hard clustering version of the IB method was used in [15] to cluster documents after clustering words. The work in [10] extended the above framework to repeatedly cluster documents and then words. An efficient algorithm was presented in [8] that monotonically increases the preserved mutual information by intertwining both the row and column clusterings at all stages. A more generalized co-clustering framework was presented in [2] wherein any Bregman divergence can be used in the objective function, and various conditional expectation based constraints can be incorporated into the framework.

There have been many research studies that perform clustering based on SVD- or eigenvector-based decomposition [7, 3, 9, 14]. The latent semantic indexing method (LSI) [7] projects each data vector into the singular vector space through the SVD, and then conducts the clustering using traditional data clustering algorithms (such as K-means) in the transformed space. Since the computed singular vectors or eigenvectors do not correspond directly to individual clusters, the decompositions from SVD- or eigenvector-based methods are difficult to interpret and to map to the final clusters; as a result, traditional data clustering methods such as K-means must be applied in the transformed space.

Recently, another matrix decomposition formulation, Nonnegative Matrix Factorization (NMF) [6], has been used for clustering [17]. NMF has the intuitive interpretation for the result. However, it focuses on one-dimension of the data matrix and does not take advantage of the duality between the rows and the columns of a matrix.

3. BLOCK VALUE DECOMPOSITION

we start by reviewing the notion of dyadic data. The notion dyadic refers to a domain with two sets of objects, $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{Y} = \{y_1, \ldots, y_m\}$ in which the observations are made for $dyads(x, y)$. Usually a dyad is a scalar value $w(x, y)$, e.g., the frequency of co-occurrence, or the strength of preference/association /expression level. For the scalar *dyads*, the data can always be organized as an n-by-m two-dimensional matrix $\mathbf{Z}$ by mapping the row indices into $\mathcal{X}$ and the column indices into $\mathcal{Y}$. Then, each $w(x, y)$ corresponds to one element of $\mathbf{Z}$.

We are interested in simultaneously clustering $\mathcal{X}$ into k disjoint clusters and $\mathcal{Y}$ into l disjoint clusters. This is equivalent to finding block structure of the matrix $\mathbf{Z}$, i.e., finding $k \times l$ submatrices of $\mathbf{Z}$ such that the elements within each submatrix are similar to each other and elements from different submatrices are dissimilar to each other. The original data matrix and the permuted matrix with explicit block structure in Figure 1 give an illustrative example.

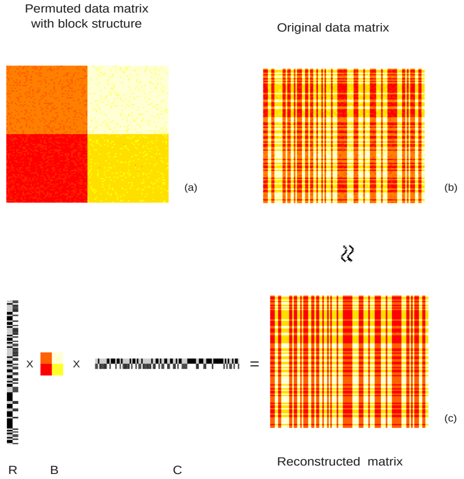

Figure 1: The original data matrix (b) with a 2×2 block structure which is demonstrated by the permuted data matrix (a). The row-coefficient matrix R, the block value matrix B, and the column-coefficient matrix C give a reconstructed matrix (c) to approximate the original data matrix (b).

Since the elements within each block are similar to each other, we expect one center to represent each block. Therefore a $k \times l$ small matrix is considered as the compact representation for the original data matrix with a $k \times l$ block structure. In the traditional one-way clustering, given the cluster centers and the weights that denote degrees of observations associated with their clusters, one can approximate the original data by linear combinations of the cluster centers. Similarly, we should be able to ”reconstruct” the original data matrix by the linear combinations of the block centers. Based on this observation, we formulate the problem of co-clustering dyadic data as the optimization problem of matrix decomposition, i.e., block value decomposition (BVD).

Definition 1. Block value decomposition of a data matrix $\mathbf{Z} \in \Re^{n \times m}$ is given by the minimization of

$$f(\mathbf{R}, \mathbf{B}, \mathbf{C}) = \|\mathbf{Z} - \mathbf{RBC}\|^2 \qquad (1)$$

subject to the constraints $\forall ij : \mathbf{R}_{ij} \geq 0$ and $\mathbf{C}_{ij} \geq 0$, where $\| \cdot \|$ denote Frobenius matrix norm, $\mathbf{R} \in \Re^{n \times k}$, $\mathbf{B} \in \Re^{k \times l}$, $\mathbf{C} \in \Re^{l \times m}$, $k \ll n$, and $l \ll m$.

We call the elements of $\mathbf{B}$ as the block values; $\mathbf{B}$ as the block value matrix; $\mathbf{R}$ as the row-coefficient matrix; and $\mathbf{C}$ as the column-coefficient matrix. As is discussed before, $\mathbf{B}$ may be considered as a compact representation of $\mathbf{Z}$; $\mathbf{R}$ denotes the degrees of rows associated with their clusters; and $\mathbf{C}$ denotes the degrees of the columns associated with their clusters. We seek to approximate the original data matrix by the reconstructed matrix, i.e., $\mathbf{RBC}$, as illustrated in Figure 1.

Under the BVD framework, the combinations of the components also have an intuitive interpretation. **RB** is the matrix containing the basis for the column space of **Z** and **BC** contains the basis for the row space of **Z**. For example, for a word-by-document matrix **Z**, each column of **RB** captures a base topic of a particular document cluster and each row of **BC** captures a base topic of a word cluster.

Comparing with SVD-based approaches, there are two main differences between BVD and SVD. First, in BVD, it is natural to consider each row or column of a data matrix as an additive combinations of the block values since BVD does not allows negative values in **R** and **C**. In contrast, since SVD allows the negative values in each component, there is no intuitive interpretation for the negative combinations. Second, unlike the singular vectors in SVD, the basis vectors contained in **RB** and **BC** are not necessarily orthogonal. Although singular vectors in SVD have a statistical interpretation as the directions of the variance, they typically do not have clear physical interpretations. In contrast, the directions of the basis vectors in BVD have much more straightforward correspondence to the clusters. In summary, compared with SVD or eigenvector-based decomposition, the decomposition from BVD has an intuitive interpretation, which is necessary for many data mining applications.

BVD provides a general framework for co-clustering. Depending on different data types in different applications, various formulations and algorithms may be developed under the BVD framework. An interesting observation is that the data matrices in many important applications are typically non-negative, such as the co-occurrence tables, the performance/rating matrices and the proximity matrices. Some other data may be transformed into the non-negative form, such as the gene expression data. Therefore, in the rest of the paper, we concentrate on a sub-framework of BVD, the non-negative block value decomposition (NBVD).

NBVD puts an extra non-negative constraint on the **B**. The formal definition is given as follows.

Definition 2. Non-negative block value decomposition of a non-negative data matrix $\mathbf{Z} \in \Re^{n\times m}$ (i.e., $\forall ij : \mathbf{Z}_{ij} \geq 0$) is given by the minimization of

$$f(\mathbf{R}, \mathbf{B}, \mathbf{C}) = \|\mathbf{Z} - \mathbf{RBC}\|^2 \tag{2}$$

subject to the constraints $\forall ij : \mathbf{R}_{ij} \geq 0, \mathbf{B}_{ij} \geq 0$ and $\mathbf{C}_{ij} \geq 0$, where $\mathbf{R} \in \Re^{n\times k}$, $\mathbf{B} \in \Re^{k\times l}$, $\mathbf{C} \in \Re^{l\times m}$, $k \ll n$, and $l \ll m$.

Finally, we compare NBVD with NMF [6]. Given a non-negative data matrix **V**, NMF seeks to find an approximate factorization $\mathbf{V} \approx \mathbf{WH}$ with non-negative components **W** and **H**. Essentially, NMF concentrates on the one-way clustering and does not take the advantage of the duality between the row clustering and the column clustering. In fact NMF may be considered as a special case of NBVD in the sense that $\mathbf{WH} = \mathbf{WIH}$, where **I** is an identity matrix. By this formulation, NMF is a special case of NBVD and it does co-clustering with the additional restrictions that the number of the row clusters equals to the number of the column clusters and that each row cluster is associated with one column cluster.

4. DERIVATION OF THE ALGORITHM

The objective function in (2) is convex in **R**, **B** and **C** respectively. However, it is not convex in all of them simultaneously. Thus, it is unrealistic to expect an algorithm to find the global minimum. We derive an EM [1] style algorithm that converges to a local minimum by iteratively updating the decomposition using a set of multiplicative updating rules.

First, we prove the following theorem which is the basis of NBVD algorithm.

THEOREM 1. *If* **R**, **B** *and* **C** *are a local minimizer of the objective function in (2), then the equations*

$$\begin{aligned}
(\mathbf{ZC}^T\mathbf{B}^T) \odot \mathbf{R} - (\mathbf{RBCC}^T\mathbf{B}^T) \odot \mathbf{R} &= 0 \qquad (3)\\
(\mathbf{R}^T\mathbf{ZC}^T) \odot \mathbf{B} - (\mathbf{R}^T\mathbf{RBCC}^T) \odot \mathbf{B} &= 0 \qquad (4)\\
(\mathbf{B}^T\mathbf{R}^T\mathbf{Z}) \odot \mathbf{C} - (\mathbf{B}^T\mathbf{R}^T\mathbf{RBC}) \odot \mathbf{C} &= 0 \qquad (5)
\end{aligned}$$

are satisfied, where $\odot$ *denotes the Hadamard product or entrywise product of two matrices.*

PROOF. Let λ_1, λ_2, and λ_3 be the Lagrange multipliers for the constraint $\mathbf{R}, \mathbf{B}$, and $\mathbf{C} \geq 0$, respectively, where $\lambda_1 \in \Re^{k\times n}$, $\lambda_2 \in \Re^{l\times k}$, and $\lambda_3 \in \Re^{m\times l}$. The Lagrange function $L(\mathbf{R}, \mathbf{B}, \mathbf{C}, \lambda_1, \lambda_2, \lambda_3)$ becomes:

$$L = f(\mathbf{R}, \mathbf{B}, \mathbf{C}) - \mathrm{tr}(\lambda_1\mathbf{R}^T) - \mathrm{tr}(\lambda_2\mathbf{B}^T) - \mathrm{tr}(\lambda_3\mathbf{C}^T) \tag{6}$$

The Kuhn-Tucker conditions are:

$$\begin{aligned}
\partial L/\partial \mathbf{R} &= 0 \qquad (7)\\
\partial L/\partial \mathbf{B} &= 0 \qquad (8)\\
\partial L/\partial \mathbf{C} &= 0 \qquad (9)\\
\lambda_1 \odot \mathbf{R} &= 0 \qquad (10)\\
\lambda_2 \odot \mathbf{B} &= 0 \qquad (11)\\
\lambda_3 \odot \mathbf{C} &= 0 \qquad (12)
\end{aligned}$$

Taking the derivatives, we obtain the following three equations from (7), (8), and (9), respectively.

$$\begin{aligned}
2\mathbf{ZC}^T\mathbf{B}^T - 2\mathbf{RBCC}^T\mathbf{B}^T + \lambda_1 &= 0 \qquad (13)\\
2\mathbf{R}^T\mathbf{ZC}^T - 2\mathbf{R}^T\mathbf{RBCC}^T + \lambda_2 &= 0 \qquad (14)\\
2\mathbf{B}^T\mathbf{R}^T\mathbf{Z} - 2\mathbf{B}^T\mathbf{R}^T\mathbf{RBC} + \lambda_3 &= 0 \qquad (15)
\end{aligned}$$

Applying the Hadamard multiplication on both sides of (13), (14), and (15) by **R**, **B**, and **C**, respectively, and using conditions (10), (11), and (12), the proof is completed. □

Based on Theorem 1, we propose following updating rules.

$$\begin{aligned}
\mathbf{R}_{ij} &\leftarrow \mathbf{R}_{ij}\frac{(\mathbf{ZC}^T\mathbf{B}^T)_{ij}}{(\mathbf{RBCC}^T\mathbf{B}^T)_{ij}} \qquad (16)\\
\mathbf{B}_{ij} &\leftarrow \mathbf{B}_{ij}\frac{(\mathbf{R}^T\mathbf{ZC}^T)_{ij}}{(\mathbf{R}^T\mathbf{RBCC}^T)_{ij}} \qquad (17)\\
\mathbf{C}_{ij} &\leftarrow \mathbf{C}_{ij}\frac{(\mathbf{B}^T\mathbf{R}^T\mathbf{Z})_{ij}}{(\mathbf{B}^T\mathbf{R}^T\mathbf{RBC})_{ij}} \qquad (18)
\end{aligned}$$

The time complexity of NBVD algorithm can be shown as $\mathcal{O}(t(k + l)nm)$ where t is the number of iterations. The complexity is the same as that of the classic one-way clustering algorithm, k-means clustering whose time complexity is $\mathcal{O}(tknm)$. Since NBVD algorithm is simple to implement and only involves basic matrix operations, it is easy to take the benefit of distributed computing when dealing with very large data set.

The conditions in Theorem 1 are the necessary conditions but not the sufficient conditions for a local minimum. To assure that the NBVD algorithm is correct, we need to prove that the objective function (2) is nonincreasing under the updating rules (16), (17) and (18). This can be done by making use the concept of an auxiliary function similar to that used in the EM algorithm [1] and NMF [13]. Due to the space limit, we omit the details here.

Finally, we consider a special case of NBVD. In practice, there exists a special type of data, symmetric dyadic data. The notion of symmetric dyadic refers to a domain with two identical sets of objects , $\mathcal{X} = \{x_1, \ldots, x_n\}$, in which the observations are made for $dyads(a, b)$, where both a and b are from $\mathcal{X}$ and $dyads(a, b) = dyads(b, a)$. Symmetric dyadic data may be considered as a two-dimensional symmetric matrix. For example, a proximity matrix may be considered as a symmetric dyadic data.

NBVD algorithm cannot directly be applied to non-negative symmetric dyadic data, because the extra condition $\mathbf{R} = \mathbf{C}^T$ needs to be satisfied. Consequently, NBVD algorithm needs to be revised to accommodate this special case. The formal definition for symmetric NBVD is,

Definition 3. Symmetric non-negative block value decomposition of a symmetric non-negative data matrix $\mathbf{Z} \in \Re^{n\times n}$ (i.e., $\forall ij : \mathbf{Z}_{ij} \geq 0$) is given by the minimization of

$$f(\mathbf{S}, \mathbf{B}) = \|\mathbf{Z} - \mathbf{SBS}^T\|^2 \quad (19)$$

$\forall ij : \mathbf{S}_{ij} \geq 0$ and $\mathbf{B}_{ij} \geq 0$, where $\mathbf{S} \in \Re^{n\times k}$, $\mathbf{B} \in \Re^{k\times k}$, and $k \ll n$.

Given this definition, we derive the updating rules for symmetric NBVD as follows.

$$\mathbf{S}_{ij} \leftarrow \mathbf{S}_{ij}\frac{(\mathbf{ZSB})_{ij}}{(\mathbf{SBS}^T\mathbf{SB})_{ij}} \quad (20)$$

$$\mathbf{B}_{ij} \leftarrow \mathbf{B}_{ij}\frac{(\mathbf{S}^T\mathbf{ZS})_{ij}}{(\mathbf{S}^T\mathbf{SBS}^T\mathbf{S})_{ij}} \quad (21)$$

Note that the symmetric NBVD provides only one clustering result though it does clustering on both dimensions of the data matrix. The symmetric NBVD is not a trivial special case of NBVD. It has a very important application, graph partition.

5. EMPIRICAL EVALUATIONS

5.1 Data Sets and Parameter Settings

We conduct the performance evaluation using the various subsets of 20-Newsgroup data (*NG20*) [12]and *CLASSIC3* data set [8]. The *NG20* data set consists of approximately 20,000 newsgroup articles collected evenly from 20 different usenet newsgroups. We have exactly duplicated this data set that is also used in [8, 10] for document co-clustering in order to ensure the direct comparability in the evaluations. Many of the newsgroups share similar topics and about 4.5% of the documents are cross posted making the boundaries between some news-groups rather fuzzy. To make our comparison consistent with the existing algorithms we have reconstructed various subsets of NG20 used in [8, 10] to all the subsets, i.e., removing stop words, ignoring file headers, and selecting the top 2000 words based on the mutual information. As in [10], we include the subject line in the articles. Specific details of the subsets are given in Table 1.

Since each document vector of word-by-document matrix is normalized to have unit L^2 norm, in the implementation of the NBVD, we normalize each column of $\mathbf{RB}$ to have the unit L^2 norm. Assume that $\mathbf{RB}$ is normalized to $\mathbf{RBV}$. The cluster labels for the documents are given by $\mathbf{V}^{-1}\mathbf{C}$ instead of $\mathbf{C}$.

We measure the clustering performance using the accuracy given by the confusion matrix of the obtained clusters and the "real" classes. Each entry (i, j) in the confusion matrix represents the number of documents in cluster i that are in true class j. Specifically, we use the micro-averaged-precision.

5.2 Experiment on Word-Document Data

This section provides empirical evidence to demonstrate that, as a general co-clustering algorithm, how NBVD improves the document clustering accuracy in comparison with NMF[6], and two other co-clustering algorithms, Information-theoretic Co-Clustering (ICC) [8] and Iterative Double Clustering algorithm (IDC) [10] .

In the experiments, initial matrices are generated as follows. All the elements of $\mathbf{R}$ and $\mathbf{C}$ are generated from uniform distribution between 0 and 1 and all the elements of B are simply assigned to the mean value of the data matrix. Since NBVD algorithm is not guaranteed to find the global minimum, it is beneficial to run the algorithm several times with different initial values and choose one trial with a minimal objective value. In reality, usually a few number of trials is sufficient. In the experiments reported in this paper, three trials of NBVD are performed in each test run and the final results are averages for twenty test runs. The experiments for NMF are conducted in the same way.

Table 2 records the two confusion matrices obtained on the *CLASSIC3* data set using NMF and NBVD, respectively, with 3 word clusters that is the number of true word clusters. Observe that NBVD extracted the original clusters with micro-averaged-precision of 0.9879 and NMF led to a micro-averaged-precision of 0.9866. It is not surprising that NBVD and NMF have almost the same performance on the *CLASSIC3* data set. This is due to the fact that when there exists perfect one-to-one correspondence between row clusters and column clusters, the block value matrix $\mathbf{B}$ is close to the identity matrix and the NMF is equivalent to NBVD.

Table 3 shows a block value matrix for the data set *CLASSIC3*. The perfect diagonal structure of Table 3 indicates the one-to-one correspondence structure of document clusters and word clusters for *CLASSIC3*.

Table 4 shows the two confusion matrices obtained on the *Multi5* data set by NBVD and NMF respectively. NBVD and NMF yield micro-averaged-precision of 0.944 and 0.884 respectively. This experiment shows that NBVD has a better performance than NMF on the data set *Multi5*. Compared with *CLASSIC3*, *Multi5* has more complicated hidden block structure and there is no simple one-to-one relationship between document clusters and word clusters. This demonstrates that by exploiting the duality of the row clustering and the column clustering, NBVD is more powerful to discover the complicated hidden block structure of the data than NMF.

Table 5 shows the micro-averaged-precision measures on

Dataset Name	Newsgroups Included	# Documents per Group	Total Documents
Binary	talk.politics.mideast, talk.politics.misc	250	500
Multi5	comp.graphics, rec.motorcycles, res.sports.baseball, sci.space, talk.politics.mideast	100	500
Multi10	alt.atheism, comp.sys.mac.hardware, misc.forsale, res.autos, res.sport.hockey, sci.crypt, sci.eldectronics, sci.med, sci.space, talk.politics.gun	50	500

Table 1: Datasets details. Each data set is randomly and evenly sampled from specific newsgroups.

NBVD			NMF		
1008	1	2	1014	4	2
25	1459	19	18	1454	25
0	0	1379	1	2	1373

Table 2: Both NBVD and NMF accurately recover the original clusters in the *CLASSIC3* data set.

0.701	0.000	0.000
0.000	0.608	0.000
0.000	0.000	1.000

Table 3: A normalized block value matrix on the *CLASSIS3* data set.

all data sets from *NG20* data. All NBVD precision values are obtained by running NBVD on the number of the true document clusters and the corresponding optimal numbers of the word clusters which are found by extra experiments that evaluate the precision at varied number of word clusters (the details are omitted due to the space limit). The peaked ITC and IDC precision values are quoted from [8] and [10], respectively. On all data sets NBVD performs better than its one-sided counterpart NMF. This result justifies the need to exploit duality between the word clustering and the document clustering. Compared with other two state-of-the-art co-clustering algorithms, NBVD shows clear improvements on precision for almost all data sets. In particular more substantial improvements are observed on the complicated data sets with more clusters, which is the typical scenario in practice.

5.3 Experiments on Proximity Data

In this section we provide empirical evidence to demonstrate the potential of the symmetric NBVD on the important application, graph partition. We still concentrates on the task of document clustering; but this time it is formed as a graph partition problem. The whole document data collection is represented as an undirected graph. Each node of the graph represents a document, and each edge (i, j) is assigned a weight w_{ij} to reflect the similarity between documents i and j. We conduct experiments on document clustering based on graph partition and show that the symmetric NBVD has superior performance to two state-of-the-art methods, the Average Association (AA) [18] and the Normalized Cut (NC) [14].

NBVD					NMF				
92	1	4	3	1	94	4	4	13	2
2	96	3	3	0	1	88	5	5	4
1	0	93	1	0	2	3	90	5	2
4	1	0	93	1	3	4	1	77	0
1	2	0	0	98	0	1	0	3	93

Table 4: NBVD extracts the block structure more accurately than NMF on *Multi5* data set.

	NBVD	NMF	ICC	IDC
Binary	0.95	0.91	0.96	0.85
Multi5	0.93	0.88	0.89	0.88
Multi10	0.67	0.60	0.54	0.55

Table 5: NBVD shows clear improvements on the micro-averaged-precision values on different newsgroup data sets over other algorithms.

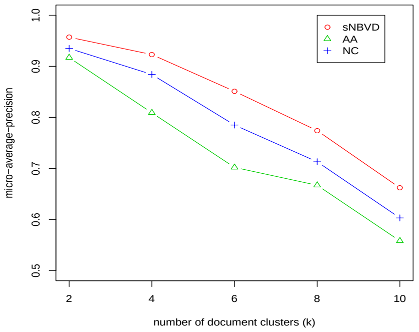

Figure 2: The symmetric NBVD shows substantial improvements measured as micro-averaged-precision values on the newsgroup data sets with different cluster numbers over AA and NC.

We use the same data set, *NG20*, with the same preprocessing steps defined before. Since each column of the word-document co-occurrence matrix $\mathbf{Z}$ has been normalized to the unit L_2 norm, the proximity matrix for the documents is simply determined as $\mathbf{W} = \mathbf{Z}^T\mathbf{Z}$. The similarity between two documents is the cosine similarity. Similarly, micro-averaged-precision is used as the measure metric.

At each run of the test, k news groups are randomly selected from twenty newsgroups and 250 documents are randomly selected from each selected newsgroup. For each given cluster number k, 20 test runs are conducted and the final precision value is the average of the twenty test runs. As we did for the general NBVD experiments, 3 trials of the symmetric NBVD are performed in each test run.

From the performance results reported in Figure 2, it is clear that as a graph partition algorithm, the symmetric

1.134	0.000	0.013	0.000	0.000	0.000
0.000	1.638	0.000	0.000	0.000	0.000
0.013	0.000	1.993	0.000	0.000	0.000
0.000	0.000	0.000	1.425	0.000	0.000
0.000	0.000	0.000	0.000	1.629	0.000
0.000	0.000	0.000	0.000	0.000	1.686

Table 6: The block value matrix of the symmetric NBVD on the *Multi5* data set with 6 document clusters.

0.931	0.000	0.000	0.000	0.000
0.000	1.206	0.000	0.000	0.000
0.000	0.000	0.890	0.000	0.000
0.000	0.000	0.000	1.057	0.000
0.000	0.000	0.000	0.000	1.629

Table 7: The block value matrix of the symmetric NBVD on the *Multi5* data set with 5 document clusters.

NBVD improves the document clustering precision substantially over AA and NC.

Finally, we apply the symmetric NBVD to the *Multi5* data set to demonstrate the nice property of the block value matrix under the symmetric NBVD. Since the row clusters and the column clusters are identical under symmetric NBVD, the block values under the symmetric NBVD have a very intuitive interpretation. They represent the similarity or distance between the clusters. When applying the symmetric NBVD to the proximity matrix, the resulting block value matrix $\mathbf{B}$ may be considered as the generic proximity matrix for the clusters, i.e., $\mathbf{B}_{ij}$ denotes the similarity between the ith clusters and jth clusters. Consequently, the better diagonal structure $\mathbf{B}$ has, the better clustering we obtain. In the block value matrix of Table 6, Cluster 1 and Cluster 3 are more similar to each other than any other pair of different clusters. Thus, Cluster 1 and Cluster 3 may be merged to make a better clustering. Applying the symmetric NBVD to *Multi5* with 5 document clusters that is the true number of document clusters, we obtain a perfect diagonal block value matrix shown in Table 7. The nice property of the block value matrix not only provides the intuitive information for the distribution of the clusters and the quality of the clustering, but also indicates some interesting and open questions under the BVD framework, e.g., how to enforce the nice property of the block value matrix within an algorithm to make the algorithm more robust and efficient.

6. CONCLUSIONS

In this paper, we have proposed a new co-clustering framework for dyadic data called Block Value Decomposition (BVD). Under this framework, we focus on a special but also very popular case, Non-negative Block Value Decomposition, and have presented the specific novel NBVD algorithm. We have also reported extensive empirical evaluations to demonstrate the effectiveness and the great potential of the BVD framework as well as the NBVD algorithms.

7. ACKNOWLEDGMENTS

This research is supported in part by a grant from AFRL/ Information Institute through the award number FA8750-04-1-0234. We acknowledge the advice from Dr. John Salerno at AFRL.

8. REFERENCES

[1] N. M. L. A. P. Dempster and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(8):1–38, 1977.

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

[3] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *DAC '93*.

[4] Y. Cheng and G. M. Church. Biclustering of expression data. In *ICMB*, pages 93–103.

[5] H. Cho, I. Dhillon, Y. Guan, and S. Sra. Minimum sum squared residue co-clustering of gene expression data. In *SDM*, 2004.

[6] D.D.Lee and H.S.Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[7] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[8] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD'03*, pages 89–98.

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

[10] R. El-Yaniv and O. Souroujon. Iterative double clustering for unsupervised and semi-supervised learning. In *ECML*, pages 121–132, 2001.

[11] J.A.Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, March 1972.

[12] K. Lang. NewsWeeder: learning to filter netnews. In *ICML'95*, pages 331–339, 1995.

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

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

[15] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *SIGIR '00*.

[16] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[17] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR '03*, pages 267–273. ACM Press, 2003.

[18] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. *Advances in Neural Information Processing Systems*, 14, 2002.