2009 Ninth IEEE International Conference on Data Mining

Extended Boolean Matrix Decomposition

Haibing Lu, Jaideep Vaidya, Vijayalakshmi Atluri and Yuan Hong
MSIS Department and CIMIC
Rutgers University, USA
{*haibing,jsvaidya,atluri,yhong*}*@cimic.rutgers.edu*

***Abstract*—With the vast increase in collection and storage of data, the problem of data summarization is most critical for effective data management. Since much of this data is categorical in nature, it can be viewed in terms of a Boolean matrix. Boolean matrix decomposition (BMD) has been used to provide concise and interpretable representations of Boolean data sets. A Boolean matrix can be expressed as a product of two Boolean matrices, where the first matrix represents a set of meaningful concepts, and the second describes how the observed data can be expressed as combinations of those concepts. Typically, the combination is only in terms of the set union. In other words, a successful Boolean matrix decomposition gives a set of concepts and shows how every column of the input data can be expressed as a union of some subset of those concepts. However, this way of modeling only incompletely represents real data semantics. Essentially, it ignores a critical component – the set difference operation: a column can be expressed as the combination of union of certain concepts as well as the exclusion of other concepts. This has two significant benefits. First, the total number of concepts required to describe the data may itself be reduced. Second, a more succinct summarization may be found for every column. In this paper, we propose the extended Boolean matrix decomposition (EBMD) problem, which aims to factor Boolean matrices using both the set union and set difference operations. We study several variants of the problem, show that they are NP-hard, and propose efficient heuristics to solve them. Extensive experimental results demonstrate the power of EBMD.**

I. Introduction

A matrix decomposition represents an input matrix as a product of two factor matrices. In data mining, matrix decompositions are often employed to produce concise representations of data. While the conciseness of factor matrices is important, for knowledge discovery, their interpretability is also critical. Standard numerical matrix decomposition has been long studied but does not enable analysis of categorical data. Since much of the real data such as market basket data, word-document data and gene expression data is categorical, or even Boolean in nature, a new matrix decomposition method called Boolean matrix decomposition (BMD) has attracted much attention lately from the data mining and knowledge discovery community.

The goal of BMD is to decompose a Boolean matrix (A) into two Boolean matrices (C and X), where the columns of the first matrix, called the concept matrix, can be viewed as a set of meaningful concepts (e.g. interesting itemsets, topics, etc.) and that of the second matrix, called the combination matrix, describe how each observed record (i.e., each column of A), can be expressed as a union of a subset of the concepts. Formally, a Boolean matrix decomposition of A is represented as $A = C \otimes X$, where $\otimes$ represents the Boolean matrix product such that $a_{ij} = \vee_k (c_{ik} \wedge x_{kj})$. By letting the value of each matrix cell denote the presence/absence of the item corresponding to its row, each column of a matrix can be viewed as a subset of items. For example, a column $\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ represents the itemset of $\{1, 3\}$. For real data, each row corresponds to an attribute value, like a word in word-document data and a product in market basket data. Semantically, each column of the concept matrix C could be assigned a meaning based on the actual items contained in it. Viewing the Boolean column as an itemset, the Boolean matrix decomposition can also be represented as $A_j = \cup_{x_{ij}=1} C_i$, where A_j and C_i denote the jth column of A and the ith column of C respectively.

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (1)$$

Consider the above illustrative BMD example: the input Boolean matrix containing four records can be summarized by only three concepts. The combination matrix shows how to reconstruct the input matrix with the three concepts – the fourth column of the input matrix is the union of the first two concepts, while each of the remaining columns is represented by one concept, respectively. In the text mining context, where each row corresponds to a word and each column of the input matrix represents a document, such a BMD describes four documents by three topics corresponding to the three columns of the concept matrix respectively. If the input matrix represents movie feedback, with rows corresponding to movies, columns corresponding to users, and 1 denoting "like", the columns of the concept matrix can be interpreted as movie types and thus a person's preference can be described as a combination of movie types.

BMD has proven to be quite useful in analyzing and summarizing some real data sets. However, this way of

1550-4786/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDM.2009.61

modeling only incompletely represents real data semantics. Essentially, standard BMD ignores a critical component – the set difference operation. Given a BMD solution, an input Boolean record can only be described as a union of a subset of concepts. In reality, a record may be more succintly described as an inclusion of some concepts with exclusion of other concepts. For example, consider a long presidential speech that covers all topics except "EDUCATION". With only the set union operation, to describe that speech, we have to list all topics that appear in it. If the set difference operation can be utilized, we can create a topic called "ALL-TOPICS" and represent the presidential speech by "ALL-TOPICS \ EDUCATION". Consider Equation (1) again, in the context of movie feedbacks. Suppose that the films corresponding to the second and third rows are horror films, and the films corresponding to the first, second and fourth rows are directed by the same director "John". If employing the BMD model, we are able to identify these two movie types, which are represented by the first and the second columns of the concept matrix. However, the third concept cannot be intuitively described and is somehow extraneous. If the set difference operation is allowed, the third concept can be eliminated as the movies favored by the third user can be represented by "John\Horror".

To address this modeling deficiency, we extend the BMD model by including the set difference operation. We denote this as *extended Boolean matrix decomposition* (EBMD), and its goal is to summarize Boolean data sets with a set of meaningful concepts, such that each data record can be represented as the inclusion of a subset of concepts while excluding a different (possibly empty) subset of concepts.

With EBMD, we expect to find concepts that are more powerful and interpretable. Additionally, with both the set difference operation and the set union operation, we can summarize the same Boolean data set with much less concepts than BMD. Furthermore, given an equal number of concepts, the accuracy of EBMD solutions should be much higher than BMD solutions. This is obvious since in the worst case an EBMD solution would be the same as the BMD solution (i.e., with no exclusion taking place). Thus the EBMD solution is always as good as or better than the corresponding BMD solution. These features make the EBMD model desirable.

However, it is a technically challenging problem to employ EBMD. Given a Boolean matrix, even with small size, the number of feasible EBMD solutions is extremely large. To fully benefit from the modeling power of EBMD, we need to find an EBMD solution, that is not only accurate, but also concise. Thus, the main task is to find such a good decomposition solution. Generally, we study two related problems. First, given an input Boolean matrix and a number k, find a set of k concepts and a corresponding combination matrix, such that the input matrix can be recontructed with minimum error. Second, when the concepts are also given, find just the combination matrix, that gives the minimum reconstruction error. For Boolean data, there are two types of reconstruction errors, a 0 becoming a 1 and a 1 becoming a 0. For some applications, only one type of error is acceptable. Thus for each problem we additionally consider two subproblems, $1-0$ error free and $0-1$ error free. Therefore, in total we study six problems. We theoretically analyze each problem and give efficient heuristics for them.

The rest of this paper is organized as follows. In Section II, we overview related work in the literature. Section III formally states the EBMD model and defines the decomposition problems. Section IV studies the computational complexity of the defined six problems. In Section V, we provide efficient heuristics for each problem. Section VI details the extensive experimental evaluation conducted using both synthetic and real data sets to validate the performance of our heuristics. Finally, Section VII concludes the paper and discusses future work.

II. Related Work

Matrix decomposition is a well-studied problem that has been the focus of significant research. Indeed, one of the earliest motivations of matrix decomposition came from the problem of solving linear equations. It is known that if a matrix A can be decomposed into the product of a lower triangular matrix L and a upper triangular matrix U, solving the systems $L(UX) = b$ and $UX = L^{-1}U$ is much easier than $AX = b$ [3]. In recent years, a big motivation for matrix decomposition is for data analysis and data processing. One of the best known methods is perhaps the Singular Value Decomposition, $X = U \sum V$, where U and V are orthogonal real-valued matrices containing the left and right singular vectors of A, and $\sum$ is a diagonal matrix containing the singular values of A [3]. One classic application of this method is to get the optimal rank-k factorization of A by setting all but the top k singular values in $\sum$ to 0. In this sense, matrix decomposition can be used for compressing data. The underlying reason is that if we find $A_{m\times n} = C_{m\times k} \cdot X_{k\times n}$ and k is much less than m and n, storing C and X instead of A will save great space [8].

While the SVD is optimal in terms of the Frobenius norm, recently, people have realized that it does not have sufficient interpretability. To address this problem, multiple new methods were proposed, like Probabilistic Latent Semantic Indexing [7], Latent Dirichlet Allocation [1] and Nonnegative Matrix Factorization (NMF)[9]. In which, NMF has attracted much attention. In NMF, the added restriction is that all the matrices should be non-negative. This can help cluster data, find centroids and even describe the probabilistic relationships between individual points and centroids. Ding et al. [4] show the equivalence of NMF, spectral clustering and K-means clustering.

Since many real applications involve Boolean data, such as document-term data, web click-stream data (users vs

websites) and protein-protein complex interaction network [15], Boolean data have obtained a special and important space in the domain of data analysis [10]. It is natural to represent Boolean data by Boolean matrices, which are a special case of non-negative matrices. Many research problems involved in Boolean data analysis can be reduced to Boolean matrix factorization. Geerts et al. [6] propose the tiling databases problem which aims to find a set of tiles to cover a 0-1 database. Since a tile can be represented by a Boolean vector, the tiling databases problem is reduced to finding a factorization of $A = C \bigotimes X$ by limiting each column of C to be a subset of one column of A, because each tile can only cover cells with value 1. Miettinen et al. [13] consider C as a discrete basis of A, from which A can be reconstructed. Given A, how to find a good basis is their core problem. This work is further developed by Miettinen [12] where C is limited to be the subset of columns of A. This limitation gives increased interpretability since each column of C can be seen as a centroid of A from the perspective of clustering. Mittinen [12] also allows the factorization $C \bigotimes X$ to cover cells containing zeros in A as long as the amount of error is within a tolerable threshold. Lu et al.[11] look at the Boolean matrix factorization problem in the context of role based access control (RBAC). The first and most difficult step of implementing RBAC is mining roles given the user-permission assignment. By representing the user-permission assignment, the user-role assignment and the mined role set by Boolean matrices A, C and X, we have $A = C \bigotimes A$. Therefore, the role mining problem is to find a Boolean matrix factorization of A.

III. EBMD

As discussed earlier, the goal of EBMD is to describe a set of observed records with a small set of concepts, such that each record can be represented as inclusion of one subset of concepts with exclusion of another subset of concepts. If a record includes one concept, that record should contain all elements of that concept; if a record excludes one concept, that record should not contain any element of that record. As is natural in set operations, exclusion overrides inclusion. In other words, if a record excludes one concept, any element in that concept is not included in the reconstructed record, even if it is present in any other concept that is included in that record. This is quite realistic as can be seen by the following example: consider a user who is a fan of some actress, but does not like horror films. So he watched all films starring that actress except for horror films.

The essential task of EBMD is to find a set of concepts and the way of reconstructing the input Boolean matrix with those concepts. Similar to BMD, a concept is represented by a Boolean vector. In BMD, the combinations are represented by a Boolean matrix, where an element of 1 for a record denotes that the corresponding concept is included, otherwise not. To reflect the set difference operation, we introduce elements of -1. So an EBMD solution of a Boolean matrix $A_{m\times n}$ is in a form of $\{C_{m\times k}, X_{k\times n}\}$, where the concept matrix C is a Boolean matrix and the combination matrix is in $\{-1, 0, 1\}$ where $x_{ij} = 1$ denotes the jth record includes the ith concept and $x_{ij} = -1$ denotes the jth record excludes the ith concept. In contrast to BMD, we denote EBMD as $A = C \odot X$. The following is the formal set-theoretic EMBD definition:

Definition 1 (EBMD): $\{C \in \{0,1\}, X \in \{-1,0,1\}\}$ is called an EBMD solution of $A \in \{0,1\}$, denoted by $A = C \odot X$, if $A_j = \cup_{x_{ij}=1} C_i \setminus \cup_{x_{ij}=-1} C_i$, where A_j denotes the item subset corresponding to elements of 1 in the jth column of A and C_i denotes similarly.

Although the definition of EBMD is intuitive, the $\odot$ operator cannot be directly executed as the $\otimes$ operator of BMD. So we give the following definition of the $\odot$ operator based on logic arithmetic.

Definition 2 ($\odot$ operator): The $\odot$ operator operates over a matrix $C_{m\times k} \in \{0,1\}^{m\times k}$ and a matrix $X_{k\times n} \in \{-1,0,1\}^{k\times n}$. If $A_{m\times n} = C_{m\times k} \odot X_{k\times n}$, we have

$$\begin{cases} a_{ij=1} & if\ (\exists\ t_1)\ (c_{i,t_1} = 1\ AND\ x_{t_1,j} = 1) \\ & AND\ (\neg\exists\ t_2)\ (c_{i,t_2} = 1\ AND\ x_{t_2,j} = -1) \\ a_{ij=0} & if\ (\neg\exists\ t_1)\ (c_{i,t_1} = 1\ AND\ x_{t_1,j} = 1) \\ & OR\ (\exists\ t_2)\ (c_{i,t_2} = 1\ AND\ x_{t_2,j} = -1) \end{cases},$$

where $i \in [1, m]$ and $j \in [1, n]$

Note that the $\odot$ and $\otimes$ operators are equivalent when all entries in X are 0 or 1.

To illustrate EBMD, we decompose the Boolean matrix employed in the introduction section. One of its feasible EBMD solutions is as follows:

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \odot \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & -1 & 1 \end{pmatrix}.$$

Utilizing the set difference operation, the EBMD solution needs only two concepts to describe the same Boolean matrix, while at least three concepts are required with BMD. These are also more interpretable. In the context of film feedback, if two columns of the concept matrix happen to correspond to films directed by the same director and horror films respectively, the EBMD solution can be interpreted as: the first user is a fan of that director; the second user loves horror films; the third user is a fan of that director but dislikes horror films; while the fourth is a fan of that director and also loves horror films.

As we will use the $||.||_1$ norm to calculate the dissimilarity between two matrices, we give its definition in advance.

Definition 3 ($||.||_1$ norm):

$$||X_{m\times n}||_1 = \sum_{i=1}^{m}\sum_{j=1}^{n}|x_{ij}|.$$

Note that if X is a Boolean matrix, its $||.||_1$ norm is equivalent to the square of its Frobenius norm, which is $||X_{m\times n}||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij})^2}$.

A. Problems

The major advantages of EBMD over BMD are: (i) using less number of concepts to describe the same data set; (ii) using the same set of concepts to represent the same data set in a more succinct way. However, to realize those two advantages, we have to solve two problems, the extended BMD (EBMD) problem and the extended basis usage (EBU) problem.

The EBMD problem is given an input Boolean matrix to find an EBMD solution. Since in real applications people usually have some prior knowledge or expectations on the number of concepts, without loss of generality, we fix the number of concepts to be k, because even if there is no prior knowledge, by adjusting k, we may search through the whole feasible solution space. A concept is a Boolean vector. Supposing the number of rows of the input Boolean matrix is m, potential concepts are in the space of $\{0,1\}^m$ which grows exponentially with m. To make the EBMD problem practical, a natural simplification is to limit the concepts to be a subset of observed records. This is a reasonable simplification. For example, we may summarize a large document set by a few representative documents. This simplification is also employed for BMD problems. However, when the number of concepts is fixed and concepts must be a subset of observed records, there might be no feasible exact EBMD (or BMD) solutions for an input Boolean matrix. In such cases, we need to find the EBMD solution with the least error. Hence, the definition of the EBMD problem is as follows.

Problem 1 (EBMD): Given a binary matrix $A_{m\times n}$ and a nonnegative integer k, find Boolean matrices $C_{m\times k}$ and $X_{k\times n}$, such that $C_{m\times k}$ contains a subset of columns of $X_{k\times n}$, and $||A_{m\times k} - C_{m\times k} \odot X_{k\times n}||_1$ is minimized.

The EBMD problem is essentially an optimization problem. If the optimal solution of an EBMD problem with small k has few reconstruction errors, it is quite likely to correspond to patterns underlying the raw data and the discovered concepts deserve further thorough examination.

A variant of this is when the concepts are given and the task is to best describe the input Boolean matrix with the given concepts. For BMD, there is a similar problem, called the basis usage problem. Corrspondingly, we call our problem the extended basis usage problem. From the theoretical perspective, the concept matrix of a EBMD solution can be viewed as a kind of basis. In this sense, the combination matrix shows how to use the basis to construct the input Boolean matrix. The definition of the EBU problem is formally stated as follows.

Problem 2 (EBU): Given two binary matrices $A_{m\times n}$ and $C_{m\times k}$, find a binary matrix $X_{k\times m}$ such that $||A_{m\times k} - C_{m\times k} \odot X_{k\times n}||_1$ is minimized.

Note that we allow reconstruction errors in the above two problems. There are two types of reconstruction errors. The first occurs when a 1 in the input matrix is reconstructed as a 0, while the other is the reverse (a 0 is reconstructed as a 1). We call these 1-0 errors and 0-1 errors respectively. Depending on the application, one or both types of errors are not acceptable. Consider fraud and anomaly detection in data mining where a Boolean matrix is sparse and cells that are 1 are viewed as outliers. The EBMD model can be employed to mine underlying outlier patterns. As outliers are few, missing any of them in the reconstructed Boolean matrix could cause legal problems. Therefore, 1-0 errors could be strictly prohibited in such an application. Similarly, 0-1 errors cause serious problems in other applications such as the role mining problem[14], which can be modeled through Boolean matrix decomposition models. Here the input Boolean matrix is the user-permission assignment, and the decomposition solution returns the mined roles and the role-user assignment. If allowing negative role assignment, that is if a role is assigned to a user negatively, that user cannot have any permission contained by that role, this extended role mining problem with both positive and negative role assignment can be modeled through EBMD. As the user-permission assignment affects the security and privacy of an organization directly, if there is a false assignment that a user is assigned to a permission that was not assigned to him before adopting the user-role assignment scheme, the organization might be in serious danger. In such a case, 0-1 errors should be avoided absolutely. Concerned by two types of reconstruction errors, we have two extended problems for each above stated problem.

Problem 3 (1-0 error free EBMD): Given a binary matrix $A_{m\times n}$ and a nonnegative integer k, find Boolean matrices $C_{m\times k}$ and $X_{k\times n}$, such that $C_{m\times k}$ contains a subset of columns of $X_{k\times n}$ and $||A_{m\times k} - A'_{m\times k}||_1$ is minimized, where $A'_{m\times n} = C_{m\times k} \odot X_{k\times n}$ and there is no cell a_{ij} of 1 while $a'_{ij} = 0$.

Problem 4 (0-1 error free EBMD): Given a binary matrix $A_{m\times n}$ and a nonnegative integer k, find Boolean matrices $C_{m\times k}$ and $X_{k\times n}$, such that $C_{m\times k}$ contains a subset of columns of $X_{k\times n}$, and $||A_{m\times k} - A'_{m\times k}||_1$ is

minimized, where $A'_{m\times n} = C_{m\times k} \odot X_{k\times n}$ and there is no cell a_{ij} of 0 while $a'_{ij} = 1$.

Problem 5 (1-0 error free EBU): Given two binary matrices $A_{m\times n}$ and $C_{m\times k}$, find a binary matrix $X_{k\times m}$, such that $||A_{m\times k} - A'_{m\times k}||_1$ is minimized, where $A'_{m\times n} = C_{m\times k} \odot X_{k\times n}$ and there is no cell a_{ij} of 1 while $a'_{ij} = 0$.

Problem 6 (0-1 error free EBU): Given two binary matrices $A_{m\times n}$ and $C_{m\times k}$, find a binary matrix $X_{k\times m}$, such that $||A_{m\times k} - A'_{m\times k}||_1$ is minimized, where $A'_{m\times n} = C_{m\times k} \odot X_{k\times n}$ and there is no cell a_{ij} of 0 while $a'_{ij} = 1$.

IV. Computational Complexity

In this section, we study the computational complexity of Problems $1 \sim 6$. As all six problems are optimization problems, we will study the computational complexity for their decision version.

Problem 7 (Decision EBMD): Given a binary matrix $A_{m\times n}$ and nonnegative integers k and t, are there Boolean matrices $C_{m\times k}$ and $X_{k\times n}$, such that $C_{m\times k}$ contains a subset of columns of $X_{k\times n}$, and $||A_{m\times n} - C_{m\times k} \bigodot X_{k\times n}||_1 \leq t$?

To prove the decision EBMD problem is NP-complete, we polynomially transform it to a known NP-complete problem, the decision BCX problem, which is described as follows.

Problem 8 (Decision BCX [12]): Given a binary matrix $A_{m\times n}$ and nonnegative integers k and t, are there binary matrices $C_{m\times k}$ and $X_{k\times n}$ such that $C_{m\times k}$ contains a subset of columns of $A_{m\times n}$, and $||A_{m\times n} - C_{m\times k} \bigotimes X_{k\times n}||_1 \leq t$?

Theorem 1: The decision EBMD problem is NP-complete.

Proof. Given $C_{m\times k}$ and $X_{k\times n}$, it is easy to check if the inequality is satisfied. So the decision EBMD problem belongs to NP. Next, we will show that the decision EBMD problem can be reduced to the decision BCX problem polynomially. Let an instance of the decision BCX problem be $\{A'_{m\times n}, k', t'\}$. Create a decision EBMD instance $\{A, k', t'\}$, where A is a $(m + 2t') \times n$ binary matrix where the first $2t'$ rows contain all 1 cells and the rest m rows are A'. Note that if $t' \geq m * n$, the EBMD instance is trivially true. So without loss of generality, we assume $t' \leq m * n$. Now, we need to show that the decision EBMD instance is satisfied if and only if the BCX instance is satisfied. If the BCX instance is satisfied, the decision EBMD instance is obviously satisfied. To prove the other direction, we will show that the solution to the constructed decision EBMD instance cannot use exclusion at all. Let us view each binary column as an element subset containing elements corresponding to cells with value 1. Then the jth column of the reconstructed matrix from a solution of the decision EBMD instance, is equivalent to $\bigcup_{x_{ij}=1} A_i \backslash \bigcup_{x_{ij}=-1} A_i$. If there exists i such that $x_{ij=-1}$, the jth column does not contain the first $2t'$ elements, since those get excluded. In other words, the first $2t'$ elements of that column are 0, because every column of A has those $2t'$ elements. Since these are counted as reconstruction errors, naturally $||A_{m\times k} - C_{m\times k} \bigotimes X_{k\times n}||_1 \geq 2t' > t'$. Therefore, if the decision EBMD instance is satisfied, its feasible solution must have X without -1 elements. Clearly, this is also a feasible solution to the BCX instance. Therefore it is also satisfied. As $t' \leq m * n$, the above reduction is performed in polynomial time. □

Similar to the definition of the decision EBMD problem, the remaining problems can also be easily defined. So we skip the formal definitions and proceed directly to the proofs. Next, we will prove there is no polynomial algorithm for the decision 1-0 error free EBMD problem unless $P = NP$. To achieve this, we need to refer to a known NP-complete problem, the minimum cover problem.

Problem 9 (Minimum Cover [5]): Given a collection $\mathcal{C}$ of subsets of a finite set S and a positive integer $k \leq |\mathcal{C}|$, does $\mathcal{C}$ contain a cover for S of size k or less, i.e., a subset $\mathcal{C}' \in \mathcal{C}$ with $|\mathcal{C}'| \leq k$ such that every element of S belongs to at least one member of $\mathcal{C}'$?

Theorem 2: There is no polynomial algorithm for the decision 1-0 error free EBMD problem unless $P = NP$.

Proof. Assume there is a polynomial algorithm for the decision 1-0 error free EBMD problem. For any minimum cover instance $\{S, \mathcal{C}, k\}$, we can construct a decision 1-0 error free EBMD instance $\{A, k, t\}$ as follows. t is a large enough positive integer (say $|S| \cdot |C|$). A is defined as a $|S| \times |C|$ binary matrix, where $a_{ij} = 1$ if the jth subset of $\mathcal{C}$ contains the ith element, otherwise 0 (i.e., A is the matrix representation of C). Since t is large enough (i.e., number of errors is unbounded), as long as there exist k columns of A such that the union of their corresponding elements cover every element of S the decision 1-0 error free EBMD instance is satisfied – simply reconstruct every column of A using those k columns, without regard to the number of $0-1$ errors. Therefore, there exists a polynomial algorithm to check if there is a subcollection $\mathcal{C}'$ with $|\mathcal{C}'| = k$ covering $\mathcal{S}$. Thus there exists a polynomial algorithm for the minimum cover problem. As the minimum cover problem is NP-complete, the assumption is true only if $P = NP$. □

The decision EBU problem is also NP-complete. To prove it, we will refer to another known NP-complete

problem, the decision BU problem.

Problem 10 (Decision BU [12]): Given binary matrices $A_{m\times n}$ and $C_{m\times k}$, and a nonnegative integer t, is there a binary matrix X such that $||A_{m\times n}-C_{m\times k} \bigotimes X_{k\times n}||_1 \leq t$?

Theorem 3: The decision EBU problem is NP-complete.

Proof. The decision EBU problem obviously belongs to NP. The decision EBU problem can be polynomially reduced to the decision BU problem. The reduction is similar to that for the decision EBMD problem. A decision BU instance is a triplet $\{A'_{m\times n}, C'_{m\times k}, t'\}$, where t' is a positive integer. We construct a decision EBU instance $\{A, C, t\}$, where A is a $(m+2t)\times n$ binary matrix where the first $2t$ rows containing all 1's and the remaining m rows are A', and C' is a $(m+2t)\times n$ binary matrix where the first $2t$ rows contain all 1's and the remaining m rows are C'. If the solution X for that decision EBU instance consists of cells of -1, $||A_{m\times n} - C_{m\times k} \bigotimes X_{k\times n}||_1$ must be greater than t. Therefore X can only be in $\{0,1\}$. When X is limited to be in $\{0,1\}$, the $\bigodot$ operator is equivalent to the $\bigotimes$ operator. Therefore, the decision BU instance is true if and only if the constructed decision EBU instance is true.□

The decision 1-0 error free EBU problem is also NP-complete. To prove this, we construct an auxiliary problem, the decision extended RBSC problem, and prove it is NP-complete by a transformation to a known NP-complete problem, the decision RBSC problem. Then we prove the decision 1-0 error free EBU problem NP-complete by a polynomial transformation to the decision extended RBSC problem.

Problem 11 (Decision RBSC [2]): Given disjoint sets R and B of red and blue elements, a collection $\mathcal{S} = \{S_1, ..., S_n\} \in 2^{R\cup B}$, and a nonnegative number t, is there a subcollection $\mathcal{C} \subseteq \mathcal{S}$ that covers all blue elements, i.e.. $B \in \bigcup \mathcal{C}$, where $\bigcup \mathcal{C}$ denotes the union of all sets in $\mathcal{C}$, while the number of covered red elements is less than t?

The decision extended RBSC problem, we construct, is extended from the decision RBSC problem by including the set difference operation, which is formally stated as follows.

Problem 12 (Decision Extended RBSC): Given disjoint sets R and B of red and blue elements, a collection $\mathcal{S} = \{S_1, ..., S_n\} \in 2^{R\cup B}$, and a nonnegative number t, are there two subcollections $\mathcal{C}^1, \mathcal{C}^2 \subseteq \mathcal{S}$ such that $\bigcup \mathcal{C}^1 \setminus \bigcup \mathcal{C}^2$ covers all blue elements, while the number of covered red elements is less than t?

Theorem 4: The decision Extended RBSC problem is NP-complete.

Proof. It is obvious that Extended RBSC is in NP. Next, we will show RBSC can be polynomially reduced to ERBSC. Consider an instance of RBSC, i.e., a triplet $\{R, B, \mathcal{S}\}$. $\mathcal{S}$ can be divided into three parts, $\mathcal{S}^B, \mathcal{S}^{\{B,R\}}, \mathcal{S}^R$, where every subset of the subcollection $\mathcal{S}^B$ contains only blue elements, every subset of $\mathcal{S}^{\{B,R\}}$ contains both blue and red elements, and every subset of $\mathcal{S}^R$ contains only red elements. Let $\mathcal{C}$ be the optimal solution of that RBSC instance. $\mathcal{C}$ must not contain any subset from $\mathcal{S}^R$, because otherwise simply deleting that subset from $\mathcal{C}$ would reduce the cost value. We construct an instance of ERBSC of a triplet $\{R, B, \mathcal{S}^B \bigcup \mathcal{S}^{\{B,R\}}\}$. Hence, any subset of $\mathcal{S}^B \bigcup \mathcal{S}^{\{B,R\}}$ must contain at least one blue element. For any solution $\{\mathcal{C}_1, \mathcal{C}_2\}$, $\cup\mathcal{C}_1 \setminus \cup\mathcal{C}_2$ must cover all blue elements. Therefore, $\mathcal{C}_2$ has to be empty. It implies that the decision RBSC instance is true if and only if its corresponding decision ERBSC instance is true.□

Theorem 5: The decision 1-0 error free EBU problem is NP-complete.

Proof. The decision 1-0 error free EBU problem obviously belongs to NP. It can be easily reduced to the decision ERBSC problem. Consider an decision ERBSC instance, $\{R, B, \mathcal{S}, t\}$. Create a decision 1-0 error free EBU instance $\{A_{m\times 1}, C_{m\times k}, t\}$ as follows. Let $m = |R| + |B|$ and $k = |\mathcal{S}|$, each row of A and C correspond to an element of $\{R \cup B\}$, and each column of C correspond to a subset of $\mathcal{S}^B \bigcup \mathcal{S}^{\{B,R\}}$. Further let $A_{i,1} = 1$ if the ith row corresponds to a blue element, otherwise 0, and $C_{ij} = 1$ if its corresponding subset contains its corresponding blue element, otherwise 0. According to the definitions of the EBU problem and the $\bigodot$ operator, those two instances are equivalent.□

Different from other problems, the 0-1 error free EBU problem is polynomially solvable. We will prove the following theorem.

Theorem 6: The 0-1 error free EBU problem can be solved in polynomial time.

Proof. In the following, we will present a polynomial algorithm for the 0-1 error free EBU problem. For a 0-1 error free EBU instance $\{A_{m\times n}, C_{m\times k}\}$, reconstructing A from C is equivalent to reconstructing each column of A separately. Thus without loss of generality, we consider A to be a vector. For ease of explanation, we transform this problem to an extended RBSC problem, by viewing the cells with value 1 in A to be blue elements B and the cells of 0 to be red elements R, and C to be the collection $\mathcal{C}$ of corresponding subsets from $R \cup B$. Thus, the 0-1 error free EBU problem is equivalent to finding two subcollections $\mathcal{C}_1$ and $\mathcal{C}_2$, such that $\bigcup \mathcal{C}_1 \setminus \bigcup \mathcal{C}_2$ cover the most blue elements while not covering any red element. The collection $\mathcal{C}$ can be

partitioned into three groups $\{\mathcal{C}^B, \mathcal{C}^R, \mathcal{C}^{R,B}\}$. To achieve the objective, include $\mathcal{C}^B$ obviously in $\mathcal{C}_1$ and include $\mathcal{C}^R$ in $\mathcal{C}_2$. For each subset of $\mathcal{C}^{R,B}$, if all of its contained red elements belong to $\bigcup \mathcal{C}^R$, include it in $\mathcal{C}_1$. Such constructed solution is the optimal solution, as no more blue elements can be added in without bringing in red elements. The procedures are done in polynomial time.□

Unfortunately, the complexity of the 1-0 error free EBMD problem is still an open problem and left for future work.

V. Heuristics

In this section, we will propose efficient heuristics for Problems 1∼ 5. As EBMD problems can be divided into two parts, determining the concept matrix and finding the combination matrix, where the latter is a EBU problem, we study EBU problems first. After that, we discuss how to locate concepts heuristically.

A. EBU

The EBU problem is given Boolean matrices A and C to find a matrix $X \in \{-1, 0, 1\}$, such that $||A - C \odot X||_1$ is minimized. As $||A - C \odot X||_1 = \sum_i ||A_i - C \odot X_i||_1$, where A_i and X_i denote the ith column of A and X respectively, an EBU problem can be divided into a set of subproblems with each column of A as an input Boolean matrix. So without loss of generality, in the following, we consider the input Boolean matrix of the EBU problem as a Boolean vector.

In Section IV, we proved 0-1 error free EBU np-complete by transforming the extended RBSC problem to it. The EBU problem can be described as an extended RBSC problem as well. Consider the following EBU problem, where the variables $\{x_1, x_2, x_3\}$ need to be determined.

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \odot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Let each row correspond to a distinct element, where the rows at which the elements of the input Boolean vector are 1, correspond to blue elements and the remaining rows correspond to red elements. Then, the first, second and fourth rows correspond to blue elements $\{b_1, b_2, b_3\}$ respectively and the third row corresponds to the red element $\{r_1\}$. Consequently, the given concept matrix C become a collection $\mathcal{C}$ of subsets of red-blue elements $\{B \cup R\}$, where B and R denote the blue element set and the red element set respectively =, such that $\{\{b_1, b_3\}, \{b_2, r_1\}, \{r_1\}\}$. Hence, the EBU problem becomes to find two subcollections $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $\cup\mathcal{C}_1 \backslash \cup\mathcal{C}_2$ covers the most blue elements while introducing the least red elements. The measure evaluating the goodness of a solution is the number of uncovered blue elements plus the number of covered red elements.

The collection $\mathcal{C}$ can be divided into three groups $\{\mathcal{C}^B, \mathcal{C}^R, \mathcal{C}^{B,R}\}$, where $\mathcal{C}^B$ includes subsets containing blue elements only, $\mathcal{C}^R$ consists of red elements only and the subsets in $\mathcal{C}^{B,R}$ have both red and blue elements. Obviously, including $\mathcal{C}^B$ in $\mathcal{C}_1$ and $\mathcal{C}^R$ in $\mathcal{C}_2$ dose not introduce any covering error. Since $\mathcal{C}^R$ has been included in $\mathcal{C}_2$, assigning any subset of $\mathcal{C}_2$, in which all contained red elements belong to $\mathcal{C}^R$, to $\mathcal{C}_1$ does not introduce covering error either. Next, we need to assign the remaining subsets of $\mathcal{C}^{B,R}$ to either $\mathcal{C}_1$ or $\mathcal{C}_2$. As the goal is to cover the most blue elements and introduce the least red elements, we propose a greedy algorithm. At each step, choose a subset c from the remaining $\mathcal{C}^{B,R}$ with the highest covering ratio. The covering ratio is calculated by

$$CR(c) = \frac{|(c \backslash B') \cap B|}{|(c \backslash R') \cap R|},$$

where $(c \backslash B') \cap B$ gives the blue elements uncovered yet, and $(c \backslash R') \cap R$ is new red elements brought by c.

Then update B' by deleting newly covered blue elements and update R' by adding newly covered red elements in R'. The repetition stops when the highest covering ratio is not greater than 1. In other words, there is no gain by introducing a new subset in $\mathcal{C}_1$. This greedy algorithm is inspired from the classic greedy algorithm for the Knapsack problem that at each step puts the item of the highest density in the box. We hope that by including the subset of the highest covering ratio each time, blue elements can be quickly covered while the least red elements are introduced. The whole algorithm is formally stated as Algorithm 1. Note that in the greedy algorithm part, all chosen subsets are included in $\mathcal{C}_1$.

Algorithm 1 EBU

```
Input: a collection C of subsets of {B ∪ R};
Output: subcollections of C_1 and C_2;
Divide C into {C^B, C^R, C^{B,R}}
Include C^B in C_1, and C^R in C_2;
Update C^{B,R} by deleting elements contained in C^B and C^R;
Set B' = ∪C^{B,R} ∩ B and R' = ∅;
while the largest value of CR(c) over all {c ∈ C^{B,R}} is greater than 1 do
    Select the subset c ∈ C^{B,R} with the largest value of CR(c) and include it in C_1;
    Update B' as B'\(c ∩ B), and R' as R' ∪ (c ∩ R).
end while
```

B. 1-0 Error Free EBU

Without loss of generality, we consider the input Boolean matrix as a Boolean vector for the 1-0 error free EBU problem. Taken in the semantic of the RBSC problem, similar to what we did for the EBU problem, 1-0 error free EBU can be described as: given a red element set R, a blue element set B, and a collection $\mathcal{C}$ of subsets of $\{R, B\}$, find two subcollections $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $\cup\mathcal{C}_1 \backslash \mathcal{C}_2$ covers

all blue elements while introducing the least red elements. From this perspective, the difference between 1-0 error EBU and EBU is that in 1-0 error EBU all blue elements should be covered, while not for EBU. So by changing the stop condition in Algorithm 1 to be all blue elements being covered, which is represented by $B' \neq \varnothing$, we have a heuristic for 1-0 error free EBU.

C. 0-1 Error Free EBU

Still consider the input matrix as a vector. We have provided a polynomial algorithm for the 0-1 error free EBU problem when proving its polynomial solvability in Section IV. That algorithm is designed from the perspective of the red-blue set cover problem as well. Its formal statement is as follows. Note that this algorithm returns the optimal solution in polynomial time.

Algorithm 2 0-1 Error Free EBU

```
Input: blue and red element sets B and R, a collection
  C of subsets of {B ∪ R}; ;
Output: subcollections of C_1 and C_2;
Divide C into {C^B, C^R, C^{B,R}}
Include C^B in C_1 and C^R in C_2
for each c ∈ C^{B,R} do
   if c ∩ R ∈ C^R then
      Include c in C_1.
   end if
end for
```

D. EBMDs

Different from EBUs, EBMDs need to determine both the concept matrix C and the combination matrix X. If C is fixed, we can simply adopt the respective algorithms for EBUs to find X. However, determining the right C is not an easy job. Suppose the number of columns of A is n and the number of columns of C is k, even though C is limited to be a subset of columns of A, there are $\frac{n!}{k!(n-k)!}$ options. To locate C, we adopt the Loc algorithm in [12]. The Loc algorithm is to find C given A, which is executed at the first phase of solving the BMD problem. After C is located, another algorithm is employed to find X such that $A = C \otimes X$. The Loc algorithm has been experimentally proven to have good performance. It starts with a random set of columns of A in C. Then it iteratively replaces columns of C with columns of A given that it decreases the reconstruction error. Interested readers may refer to [12] for detailed procedures.

Our heuristics for EBMDs are: (i) first employ the Loc algorithm to find a BMD solution $\{C, X\}$, having $A \approx C \otimes X$; (ii) employ the found C and adopt the respective EBU algorithm to find X' in the hope that the resulting EBMD solution $C \odot X'$ would be closer to A. For example, 1-0 error free EBMD employs the heuristic of 1-0 error free EBU problem.

VI. Experiments

In this section, we present extensive experimental results on both synthetic and real data sets to validate the performance of our proposed heuristics. Our algorithms are compared on one hand against standard matrix decomposition, and on the other hand against conventional Boolean matrix decomposition. The algorithm computing standard matrix decomposition is SVD, which is a benchmark for computing standard matrix decomposition. As SVD returns results of real values, to be fair, we round them to be binary by setting all values less than 0.5 to 0, and all other values to 1. Hence, this algorithm is called SVD 0/1. The algorithm computing conventional Boolean matrix decomposition is the Loc& IterX algorithm proposed in [12], which has been experimentally proven to have better performance than other Boolean matrix decomposition algorithms.

A. Synthetic Data

We study the behavior of the heuristics with respect to the decomposition size and noise.

The synthetic data is generated as follows. First, generate k Boolean vectors randomly as basis, each of which has 50 elements and about 1/3 elements are 1. Second, use basis vectors to generate other $100-k$ vectors. The detailed procedures are : (i) randomly selecting $k/3$ basis vectors; (ii) randomly assigning half selected basis vectors to $\mathcal{C}_1$ and the other half to $\mathcal{C}_2$; (iii) computing $\cup\mathcal{C}_1 \setminus \cup\mathcal{C}_2$ as a generated vector. The size of such a generated matrix is 50×100. After that, add noise to the matrix by randomly flipping the values of a given fraction of the data.

To compare reconstruction error, we use two kinds of measure. The first is

$$ER1 = ||A - A'||_1/size(A),$$

where A is the input matrix and A' is the reconstructed matrix. The second is

$$ER2 = ||A - A'||_1/||A||_1.$$

$ER1$ reflects how much fraction of the data is not correctly reconstructed. $ER2$ is to compare the amount of reconstruction error against the total number of 1's cells. The reason of employing $ER2$ is that if the input Boolean matrix is sparse, a low value of $ER1$ does not demonstrate the input matrix is correctly reconstructed, as simply returning a matrix of zeros would have a low value of $ER1$.

The first experiment is to test the effect of size k with respect to reconstruction error. We vary k from 4 to 20 and for each size we generate 5 matrices. Reported results are mean values over these five matrices. For algorithms of Loc&IterX, EBU, 0-1 error free EBU and 1-0 error free EBU, we use basis vectors as C. The experimental

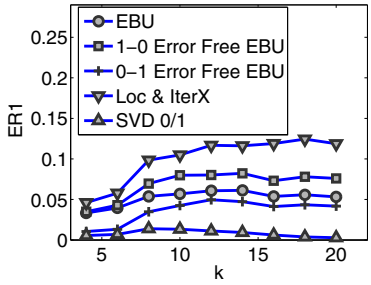

Figure 1. Reconstruction Error Ratio with $ER1$ w.r.t. k

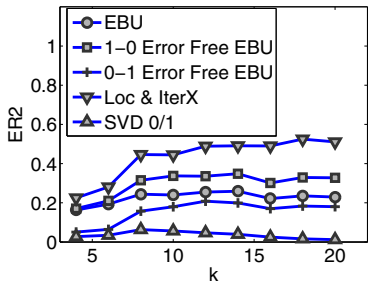

Figure 2. Reconstruction Error Ratio with $ER2$ w.r.t. k

results are as shown in Figures 1 and 2. As we can see, the reconstruction error ratios of all three EBU approaches are lower than those of Loc&IterX and close to those of SVD 0/1. With $ER1$, the reconstruction error ratios of EBU approaches are as low as 0.05 on average. With $ER2$, they are still as low as 0.2 on average.

The second experiment is to test the effect of noise with respect to reconstruction error. We vary noise ratio from 0 to 0.5. The experimental results are as shown in Figures 3 and 4. The reconstruction error ratios of EBU and 1-0 error free EBU are still lower than those of Loc&IterX and close to SVD 0/1. However, the reconstruction error ratios of 0-1 error free EBU are much higher than other approaches even though the amount of noise is little.

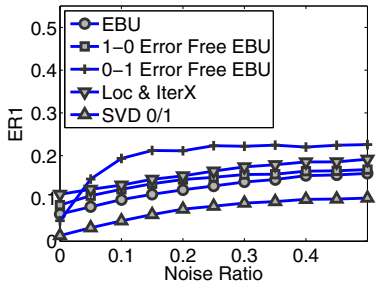

Figure 3. Reconstruction Error Ratio with $ER1$ w.r.t. Noise Ratio

B. Real Data

One of our main contributions that we claim is that our heuristic algorithm can decompose a Boolean matrix

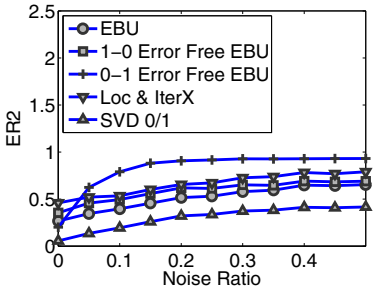

Figure 4. Reconstruction Error Ratio with $ER2$ w.r.t. Noise Ratio

with much less reconstruction error than the conventional Boolean matrix decomposition method. In this section, we demonstrate that the claim holds for real data as well.

Four real data sets are used. The News data set is a subset of 20 Newsgroups dataset [1]. We select the first 400 messages and the top frequent 100 words in them, and replace the counts with 1 or 0. Then, we obtain a 100×400 matrix, which happens to have no repetitive columns. Votes dataset [2] contains plenary votes of Finnish Parliament. Same as [12], we only consider those MPs that served an entire term. During 1999-2001, there were 773 plenary votes and 196 MPs served the entire term. As an MP can cast four different types of votes (yea, Nay, Abstain, and Absent), two different dataset are actually used: VotesYes sets Yeas as 1s and all other votes are 0s, while VotesNo sets Nays as 1's and all other votes as 0's. The Query data set is a user/clicked URL binary matrix, extracted from a large-scale query log. The query log data include two important attributes, UserID (the identity query issuer), and ClickedURL (the URL eventually clicked by that user in that single query). We have first selected the top 40 frequent clicked URLs from the query log with 1,889,761 queries in total and removed all the queries that are not related to those 40 Clicked URLs (we thus obtain 196,218 queries with 40 clicked URLs). Consequently, we have generated another dimension of the matrix by choosing the top 1000 users who have executed most queries in this small group of query log. The result is a binary matrix with the dimension of 40*1000. After deleting repetitive columns, finally we have a matrix of 40×200 .

The experimental results are shown in Tables I and II. We can see that the heuristic of EBMD decomposes real datasets with less reconstruction errors than Loc&IterX and close to SVD 0/1, while decomposition solutions provided by the heuristics for 0-1 EBMD and 1-0 EBMD have higher reconstruction error ratios.

VII. Conclusion

We introduced extended Boolean matrix decomposition, and studied their six subproblems, EBMD, 0-1 error free

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/
[2] http://www.fsd.uta.fi/english/data/cagalogue/FSD2117/meF2117e.html

Data	k	EBMD	0-1 EBMD	1-0 EBMD	Loc& IterX	SVD 0/1
News	5	0.2575	0.3966	0.3228	0.2811	0.2085
	10	0.2350	0.3791	0.2958	0.2633	0.1750
	15	0.2200	0.3647	0.2767	0.2397	0.1457
VotesNo	5	0.0777	0.1921	0.1251	0.0830	0.0642
	10	0.0704	0.1860	0.1017	0.0763	0.0496
	15	0.0684	0.1755	0.0864	0.0722	0.0398
VotesYes	5	0.1531	0.6491	0.2215	0.1613	0.0779
	10	0.1334	0.6421	0.2054	0.1459	0.0929
	15	0.1244	0.6320	0.1911	0.1336	0.0775
Query	5	0.1162	0.1220	0.3548	0.1168	0.0892
	10	0.0921	0.0974	0.2385	0.1071	0.0536
	15	0.0710	0.0812	0.1451	0.0822	0.0301

Table I
RECONSTRUCTION ERROR RATIOS WITH $ER1$ FOR REAL DATASETS

Data	k	EBMD	0-1 EBMD	1-0 EBMD	Loc& IterX	SVD 0/1
News	5	0.6154	0.9477	0.7714	0.6718	0.4982
	10	0.5616	0.9058	0.7069	0.6293	0.4181
	15	0.5258	0.8714	0.6611	0.5728	0.3482
VotesNo	5	0.3829	0.9464	0.6165	0.4091	0.3161
	10	0.3471	0.9164	0.5010	0.3761	0.2445
	15	0.3368	0.8647	0.4258	0.3557	0.1963
VotesYes	5	0.2311	0.9799	0.3343	0.2435	0.1176
	10	0.2014	0.9692	0.3100	0.2203	0.1403
	15	0.1878	0.9540	0.2885	0.2017	0.1170
Query	5	0.6961	0.7305	2.1243	0.6992	0.5340
	10	0.5516	0.5831	1.4281	0.6415	0.3211
	15	0.4251	0.4865	0.8690	0.4925	0.1804

Table II
RECONSTRUCTION ERROR RATIOS WITH $ER2$ FOR REAL DATASETS

EBMD, 1-0 error free EBMD, EBU, 0-1 error free EBU, and 1-0 error free EBU. We proved that the decision version of EBMD, 1-0 error free EBMD, EBU, and 1-0 error free EBU are NP-complete, and 0-1 error free EBU is polynomially solvable. For all subproblems except 0-1 error free EBU, we formulate them through integer programming and also present efficient heuristics. We tested the performance of our heuristics on both synthetic and real datasets. The experiential results show that the EBMD method provides better results than the conventional BMD method.

There are many types of real datasets, which could be better described with both the set difference operation and the set union operation, such as word-document data, movie feedback, and human behavior, which shows broad potential applications of EBMD.

REFERENCES

[1] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

[2] R. D. Carr, S. Doddi, G. Konjevod, and M. Marathe. On the red-blue set cover problem. In *SODA '00: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 345–353, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[3] B. I. David and T. L. N. *Numerical Linear Algebra*. Philadelphia: Society for Industrial and Applied Mathematics, 1997.

[4] C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proc. SIAM Data Mining Conf*, pages 606–610, 2005.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[6] F. Geerts, B. Goethals, and T. Mielikainen. Tiling databases. In *Discovery Science*, pages 278–289. Springer, 2004.

[7] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international conference on Research and development in information retrieval (SIGIR)*, pages 50–57, New York, NY, USA, 1999. ACM.

[8] R. A. Horn and C. R. Johnson. *Matrix Analysis*. 1997.

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

[10] T. Li. A general model for clustering binary data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 188–197, 2005.

[11] H. Lu, J. Vaidya, and V. Atluri. Optimal boolean matrix decomposition: Application to role engineering. *IEEE 24th International Conference on Data Engineering*, pages 297–306, 2008.

[12] P. Miettinen. The boolean column and column-row matrix decompositions. *Data Min. Knowl. Discov.*, 17(1):39–56, 2008.

[13] M. Pauli, M. Taneli, G. Aristides, D. Gautam, and M. Heikki. The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering*, 20(10):1348–1362, 2008.

[14] J. Vaidya, V. Atluri, and J. Warner. Roleminer: mining roles using subset enumeration. In *The 13th ACM conference on Computer and communications security*, pages 144–153, 2006.

[15] Z. Zhang, T. Li, C. Ding, and X. Zhang. Binary matrix factorization with applications. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 391–400, 2007.