An Effective Approach on Overlapping Structures Discovery for Co-clustering

Wangqun Lin[1], Yuchen Zhao[2], Philip S. Yu[2], and Bo Deng[1]

[1] Beijing Institute of System Engineering, Beijing, China
{linwangqun,bodeng}@nudt.edu.cn
[2] University of Illinois, Chicago, USA
{yzhao,psyu}@cs.uic.edu

Abstract. Co-clustering, which explores the inter-connected structures between objects and features simultaneously, has drawn much attention in the past decade. Most existing methods for co-clustering focus on partition-based approaches, which assume that each entry of the data matrix can only be assigned to one cluster. However, in the real world applications, the cluster structures can potential be overlapping. In this paper, we propose a novel overlapping co-clustering method by introducing the density guided principle for discriminative features (objects) identification. This is done by simultaneously finding the non-overlapping blocks. Based on the discovered blocks, an effective strategy is utilized to select the features (objects), which can discriminate the specified object (feature) cluster from other object (feature) clusters. Finally, according to the discriminative features (objects), a novel overlapping method, OPS, is proposed. Experimental studies on both synthetic and real-world data sets demonstrate the effectiveness and efficiency of the proposed OPS method.

1 Introduction

Co-clustering attracted much attentions during the past decade, where the task is to perform clustering on two types of inter-connected entities (i.e., rows and columns of a data matrix) simultaneously. Usually, each row of the data matrix represents an object, and each column of the data matrix represents a feature. For example, in the document analysis, the rows and columns of the data matrix correspond to the documents and words. Co-clustering on documents and words simultaneously can achieve better quality than clustering on documents alone. However, most existing co-clustering methods [1] [4] [8] are mainly partition-based, which usually assume that each entry in a data matrix can only be assigned to one cluster. Some cases of such application scenarios are as follows:

- **Row/Object Overlapping:** In many clustering applications, each individual object should be assigned to more than one cluster. For example, news articles can belong to multiple categories; Movies can have more than one genre; Chemical compounds can be associated with multiple types of efficacy.

L. Chen et al. (Eds.): APWeb 2014, LNCS 8709, pp. 56–67, 2014.

- **Column/Feature Overlapping:** For the scientific paper clustering problem, it is desirable that the clustering algorithm can automatically put papers from the same discipline into the same cluster. In the co-clustering setting, each paper is represented as one row while the term features are represented in columns. It is natural that some terms (columns) can be significant features for multiple clusters. For example, the term ”matrix” can be important for both Math and Computer Science, and the term ”molecule” can be frequently used in both Biology and Chemistry.

Such overlapping structures can often appear in a variety of clustering applications. The clustering quality can be greatly improved if the real overlapping structures of both rows and columns are captured. However, the overlapping scenarios make the problem very challenging from a number of aspects:

- Most existing works [5] [7] [14] on overlapping structures discovery focus on traditional clustering environment. The new challenge is on how to simultaneously find overlapping structures on both rows and columns. Discovered overlapping structures on rows will actively reinforce to discover the overlapping structures on columns and vice versa.
- Another challenge is on how to effectively define the overlapping criteria? If the criteria are set too strict, few overlapping structures will be discovered. However, if the criteria are too loose, many objects will be incorrectly identified to have overlapping structures, which will lead to poor clustering quality.
- Traditional co-clustering approaches usually require users to specify how many row clusters and column groups to cluster. Nonetheless, these two parameters are often difficult to obtain in reality. Designing an efficient and effective approach which requires no user specified parameter is quite challenging yet much desired.

In this paper, we will study the problem of overlapping structures discovery in the context of co-clustering. This is done by first finding the blocks, which have either dense or sparse connections, by non-overlapping co-clustering. Then based on the discovered blocks, we propose a density guided strategy to select the features (objects), which can discriminate the specified object (feature) clusters from other object (feature) clusters. Finally, according to the discriminative features (objects), a novel overlapping strategy (OPS), which can work with any non-overlapped co-clustering methods, is developed.

The rest of the paper is organized as follows. In Section 2, we introduce the related work. The strategy of overlapping co-clustering is elaborated in Section 3. Then, in Section 4 we introduce the co-clustering methods based on MDL, followed by the experimental evaluation in section 5. Finally, we conclude in Section 6.

2 Related Work

Co-clustering focuses on simultaneously clustering both dimensions of a matrix by exploiting the clear duality between rows and columns [13] [6]. Most works in co-clustering attempt to discover non-overlapping structures. Chakrabarti et al. [1] assumed the process of co-clustering as the problem of how to transfer the matrix with the least bits. By minimizing the total bits used to describe the matrix, the homogeneous blocks, whose densities are either very high or very low, are discovered. The denser blocks are used as co-clusters. Later, Papadimitriou et al. [15] further extended this method to the hierarchical situation. Long [12] proposed a Spectral Relational Clustering (SRC) approach, which iteratively embeds each type of data objects into low dimensional spaces. Since SRC needs to calculate eigenvectors, it is very time-consuming for large data set. Cheng et al.[2] devised the sequential bi-clustering model that finds one co-cluster, which has low mean squared residue scores in expression data at each time. Later, Lazzeroni et al. [10] proposed a plaid model for directly finding the overlapping co-clusters, but still can not identify multiple co-clusters simultaneously. Deodhar et al. [3] proposed a robust co-clustering algorithm called ROCC, which can work with various distance measures and different co-cluster definitions. However, in order to handle noisy or incoherent data, where a large fraction of the data points and features is irrelevant and needs to be discarded, ROCC focuses more on pruning. But in this paper, our assumption is that all of the objects and features are useful. In addition, approaches in [2] [3] [10] only focus on the overlapping structures between co-clusters but not among the row clusters and column clusters. Hence, their goals are quite different from our problem. Wang et al. [16] proposed a method similar to k-means by making use of the correlations between users and tags in social media. However, this method is only tailored for social media domain and is ineffective for the general case of overlapping structures.

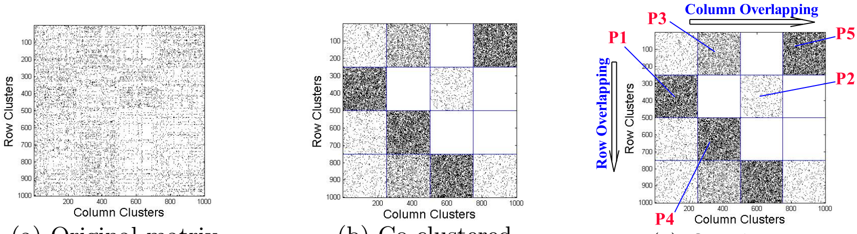
3 The Framework of Discovering Overlapping Structures for Co-clustering

An un-weighted bipartite graph G is described by a binary matrix D of $m \times n$, in which each element $e_{i,j}(1 \leq i \leq m, 1 \leq j \leq n)$ indicates whether the i-th object has a link relation with the j-th feature or not. R represents the set of rows and C represents the set of columns in D. $\mathcal{A}$ is the set of co-clustering algorithms which aim at co-clustering the set of rows, i.e., R into k row clusters and the set of columns, i.e., C into l column clusters. We use $\mathcal{I}$ denoting the set of row clusters, i.e., $\mathcal{I} = \{\mathcal{I}_i\}_{i=1}^{k}$, and $\mathcal{J}$ denoting the set of column clusters, i.e., $\mathcal{J} = \{\mathcal{J}_j\}_{j=1}^{l}$. Since each row r stands for an object and each column c stands for a feature in matrix D, we use r to represent both row and object and c to represent both column and feature in this paper.

Definition 1 (Pattern). *Given an object-feature matrix D of size $m \times n$, assume matrix D is to be co-clustered into k row clusters and l column clusters. A pattern $\mathcal{M}_i = (Q_X^i, Q_Y^i)$ is a mapping of rows and columns of matrix D respectively, where Q_X^i denotes the mapping of rows and Q_Y^i denotes the mapping of columns, i.e., $Q_X^i : \{1, 2, \cdots, m\} \to \{1, 2, \cdots, k\}$; $Q_Y^i : \{1, 2, \cdots, n\} \to \{1, 2, \cdots, l\}$. $\mathcal{M}$ denotes the set of the patterns in D, i.e., $\mathcal{M}_i \in \mathcal{M}$.*

In order to gather the similar objects into the same row clusters and the similar features into the same column clusters, co-clustering approach $\mathcal{A}_i \in \mathcal{A}$ searches the appropriate optimal pattern $\mathcal{M}^* \in \mathcal{M}$ for optimizing a specified objective function as shown in [1] [13]. A regular co-clustering process is given from Figure 1(a) to 1(b). All the discussion in this section is under the assumption that we have already computed a co-clustered matrix D. In other words, given an object-feature matrix D, the co-clustering approach $\mathcal{A}_i \in \mathcal{A}$ has already co-clustered different objects into different row clusters and different features into different column clusters.

We notice that, any row (column) cluster becomes an independent row (column) cluster because it has some discriminative feature (object) sets. Before we give the detailed description of discriminative feature (object) set, we give the observations of co-clustering process in Figures 1(a) and 1(b). It is clear that there are four row clusters and four column clusters in this example. From Figure 1(b), we notice that, for each row cluster, it certainly has some features that distinguish the row cluster itself from other row clusters. Otherwise, this row cluster will be merged into other row clusters. As shown in Figure 1(c), the first row cluster and the second row cluster are separated from each other because they have different features. In details, in the first row cluster, features in block $P3$ and $P5$ are most important features. In addition, features in $P5$ can be more discriminative than those features in $P3$ since other row clusters have much lower densities for features in $P5$. Similarly, features in $P1$ are more discriminative than features in $P2$ for the second row cluster in terms of separating from other row clusters. Symmetrically, for column clusters, objects located in $P1$ are more important than objects located in $P3$ to discriminate the first column cluster from other column clusters. Compared to objects located in $P3$, objects located in $P4$ contribute more for discriminating the second column cluster from other column clusters.



(a) Original matrix (b) Co-clustered (c) Overlapping

Fig. 1. The process of overlapping co-clustering

For any row cluster $\mathcal{I}_p \in \mathcal{I}(1 \leq p \leq k)$, in order to measure the importance of the features in column cluster $\mathcal{J}_s \in \mathcal{J}(1 \leq s \leq l)$ for distinguishing row cluster $\mathcal{I}_p$ from other row clusters, the difference of density between the block $D_{p,s}$ and average density of all blocks in the s-th column group of matrix D should be considered. This is referred to as the *density guided principle* for discriminative features (objects) identification. We give the *Discriminative Feature Function* $w(p,s)$ to evaluate the contribution of column cluster $\mathcal{J}_s$ for separating row cluster $\mathcal{I}_p$ from other row clusters as follow:

$$w(p,s) = N(D_{p,s}) - \frac{1}{|\mathcal{I}|}\sum_{i=1}^{|\mathcal{I}|} N(D_{i,s}) \tag{1}$$

where $N(D_{p,s})$ is the density function which measures the percentage of "1"s in block $D_{p,s}$. Obviously, the larger value of $w(p,s)$, the more contribution of features in $\mathcal{J}_s$ for discriminating row cluster $\mathcal{I}_p$ from other row clusters. Symmetrically, we further define the *Discriminative Object Function* $w'(s,p)$ to evaluate the contribution of row cluster $\mathcal{I}_p$ for distinguishing column cluster $\mathcal{J}_s$ from other column clusters below.

$$w'(s,p) = N(D_{p,s}) - \frac{1}{|\mathcal{J}|}\sum_{j=1}^{|\mathcal{J}|} N(D_{p,j}) \tag{2}$$

Definition 2 (Discriminative Feature Set). *Given the row cluster $\mathcal{I}_p \in \mathcal{I}$ and the column cluster $\mathcal{J}_s \in \mathcal{J}$, the group of features located in the column cluster $\mathcal{J}_s$ is the discriminative feature set for the row cluster $\mathcal{I}_p$ iff $\mathcal{J}_s$ contributes to the distinction of row cluster $\mathcal{I}_p$ from other row cluster $\mathcal{I}_q \in \mathcal{I}(p \neq q)$, i.e., $w(p,s) \geq 0$.*

Definition 3 (Discriminative Object Set). *Given the column cluster $\mathcal{J}_s \in \mathcal{J}$ and the row cluster $\mathcal{I}_p \in \mathcal{I}$, the group of objects located in the row cluster $\mathcal{I}_p$ is the discriminative object set for the column cluster $\mathcal{J}_s$ iff $\mathcal{I}_p$ contributes to the distinction of column cluster $\mathcal{J}_s$ from other column cluster $\mathcal{J}_t \in \mathcal{J}(s \neq t)$, i.e., $w'(s,p) \geq 0$.*

Given an object $r \in \mathcal{I}_q$, when we consider its relation with row cluster $\mathcal{I}_p (p \neq q)$, we examine its features shared with the objects in $\mathcal{I}_p$. Concretely, the more discriminative feature sets they shared, the closer relation they are. Moreover, for a specified discriminative feature set in $\mathcal{I}_p$, if an object r has a higher feature density in this discriminative feature set, it indicates the closer relation between object r and objects in $\mathcal{I}_p$. Consequently, for any row $r \in \mathcal{I}_q$, in order to test whether row r should also be placed into row cluster $\mathcal{I}_p$ $(p \neq q)$ or not, all the discriminative feature sets in p-th row group are considered by Equation (3).

$$E_{or}(r,p) = \sum_{f \in F} w(p,f)(N(r_f) - N(D_{p,f})) - \alpha \tag{3}$$

where r_f is the set of elements from row r located in column cluster $\mathcal{J}_f$; F is index set of discriminative feature set of column cluster, i.e., $F = \{f | w(p,f) \geq 0, \mathcal{J}_f \in \mathcal{J}\}$; α is a parameter used to control the extent of the row overlap. On the right hand side of Equation (3), the first term in the summation indicates the significance of the discriminative feature set to the p-th row group, while the second term measures the density difference of row r relative to the p-th row group. Intuitively, the larger of their product, the more likely that row r will be related to the p-th row group. We take the sum of the products over all discriminative feature sets of the p-th row group as the measure. If $E_{or}(r,p) \geq 0$, row r will not only be placed into its original row cluster $\mathcal{I}_q$ but also to row cluster $\mathcal{I}_p$. We notice that, it is possible for $(N(r_f) - N(D_{p,f}))$ to be negative. If in this case, it means row r has a lower feature density located in column cluster $\mathcal{J}_s$ than $\mathcal{I}_p$. Consequently, the possibility of placing row r into row cluster $\mathcal{I}_p$ is penalized.

Similarly, we have Equation (4) for evaluating any column $c \in \mathcal{J}_t$ whether should also be placed into column cluster $\mathcal{J}_s$ $(s \neq t)$.

$$E_{oc}(c,s) = \sum_{b \in B} w'(s,b)(N(c_b) - N(D_{b,s})) - \beta \tag{4}$$

where c_b is the set of elements from column c located in row cluster $\mathcal{I}_b$; B is the index set of discriminative object set of row cluster, i.e., $B = \{b | w'(s,b) \geq 0, \mathcal{I}_b \in \mathcal{I}\}$; β is a parameter used to control the extent of the column overlap. The description of Overlapping Pattern Search (OPS) for overlapping co-clustering is given in Algorithm 1. We note that the order of step 2 and step 3 does not matter, since going through the rows and columns are solely based on the appropriate optimal non-overlapping patterns. In other words, step 2 and step 3 are independent. Besides, $\alpha = \beta = 0$ is used in this paper.

Algorithm 1. $OPS(\mathcal{A}_i, D)$

```
1. Call co-clustering algorithm $\mathcal{A}_i$ to find the approximate optimal non-overlapping
   pattern $(Q_X^*, Q_Y^*)$ of $D$.
2. Based on $(Q_X^*, Q_Y^*)$, for each row $r \in R$ and each row cluster $\mathcal{I}_p \in \mathcal{I}$ $(r \notin \mathcal{I}_p)$,
   compute $E_{or}(r,p)$ according to Equation (3). If $E_{or}(r,p) \geq 0$, copy row $r$ to row
   cluster $\mathcal{I}_p$, i.e., $\mathcal{I}_p \leftarrow \mathcal{I}_p \cup r$.
3. Based on $(Q_X^*, Q_Y^*)$, for each column $c \in C$ and each column cluster $\mathcal{J}_s \in \mathcal{J}$
   $(c \notin \mathcal{J}_s)$, compute $E_{oc}(c,s)$ according to Equation (4). If $E_{oc}(c,s) \geq 0$, copy
   column $c$ to column cluster $\mathcal{J}_s$, i.e., $\mathcal{J}_s \leftarrow \mathcal{J}_s \cup c$.
4. Return overlapping row clusters and column clusters $(\mathcal{I}_o^*, \mathcal{J}_o^*) = (\mathcal{I}, \mathcal{J})$.
```

4 Co-clustering Approach Based on Information Compression

We have presented a general framework for overlapping co-clustering based on non-overlapping co-clustering in the last section. In this section, we give the co-clustering approach used in this paper for generating the non-overlapping row clusters and column clusters. Although the overlapping framework described in Section 3 can work with any non-overlapping co-clustering method, here we further extend FACA [1] to generate the non-overlapping co-clusters because it can be parameter free and generate good quality results. Due to the space limit, we first briefly introduce the process of FACA. Then we explain how we extend it further to be more efficient and effective. Finding the optimal co-clustering pattern is NP-hard [1]. In order to find the appropriate optimal pattern $\mathcal{M}^*$, FACA makes use of Minimum Description Length (MDL) theory to encode matrix without information loss. Assume a matrix D with $m \times n$ is divided into k row clusters and l column clusters. Compressing matrix D includes two parts which are *description complexity* $T_m(D)$ and *code length* $T_c(D)$. Therefore, the total bits used for condensing matrix D is

$$\begin{aligned}
&T(D) \\
=&T_m(D) + T_c(D) \\
=&\log^* m + \log^* n + m\lceil \log m \rceil + n\lceil \log n \rceil + \log^* k + \log^* l \\
&+ \left\lceil \log \binom{m}{m_1, \cdots, m_k} \right\rceil + \left\lceil \log \binom{n}{n_1, \cdots, n_l} \right\rceil + \sum_{i=1}^{|\mathcal{I}|} \sum_{j=1}^{|\mathcal{J}|} \lceil \log(m_i n_j + 1) \rceil \\
&+ \sum_{i=1}^{|\mathcal{I}|} \sum_{j=1}^{|\mathcal{J}|} \sum_{h=0}^{1} N_h(D_{ij}) \log \left(\frac{N(D_{ij})}{N_h(D_{ij})} \right)
\end{aligned} \tag{5}$$

where $N_h(D_{ij})$ denotes the number of "h"s (h=0 or 1) in block D_{ij}, m_i denotes the number of objects in row cluster $\mathcal{I}_i$, and n_j denotes the number of features in column cluster $\mathcal{J}_j$. All the logarithms are based 2 in Equation (5). Besides, in Equation (5), the first term to the ninth term represents the *description complexity*, and the tenth term represents the *code length*. The original algorithms only used the tenth term as the objective function. The detailed description of FACA can be found in [1]. As we will see in the experimental section, by retaining all these terms to capture the effect of both description complexity and code length, we can achieve better clustering quality.

5 Experiments

In this section, we test our method on both synthetic and real-world data sets. Each experiment is repeated 10 times and the average is reported. We use two metrics to measure the performance.The first metric used in this paper is *Purity* and the second one is *Normalized Mutual Information* (NMI) [9].

5.1 Data Set Description

Synthetic Data Set. We generate the synthetic data based on Classic3[1]. Classic3 data set contains three types of non-overlapping documents, which are MEDLINE (medical journals), CISI (information retrieval) and CRANFIELD

[1] `http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets`

(aero-dynamics). The documents and words form a bipartite graph described by a binary matrix of 3891 × 5896. In order to get the overlapping documents, firstly, we randomly select 1000 documents from each type of documents. Secondly, we randomly choose two documents d_i and d_j, which belong to two different types T_i, T_j $(i \neq j)$, from the total 3000 documents. Thirdly, we merge documents d_j and d_i together to form a new document d_{ij}, which is tagged with two types T_i and T_j. The above processes repeat until the total specified overlapping percentage $OV\%$ of new documents are generated.

Real-World Data Sets. In addition to the synthetic data set, we use two real-world data sets to test the proposed method. The first real data set is *Reuter* data set[2] , which contains 294 documents. Each document records a story happened from February 2009 to April 2009. Among these stories, 40 stories are tagged more than one type of the total six types, which are *business*, *entertainment*, *health*, *politics*, *sport* and *technology*. The second real data set is *BBC* data set[3] , which also contains six types of documents as the Reuter data set. BBC data set contains the total number of 352 documents and 40 documents are annotated with more than one type.

We notice that the standard text preprocessing approaches such as stemming, stop words removal have already been applied to all of these data sets.

5.2 Experiment Results

We compare our method with four state-of-the-art methods. The first one is FACA [1]. The second compared method is NMF [11], which is a co-clustering method based on non-negative matrix factorization. The third compared method is SRC [12] and the fourth one is DOGSM [16]. We note that FACA, NMF and SAC can detect effective row clusters and column clusters but cannot discover the overlapping structures. While DOGSM can detect the overlapping structures of row clusters and column clusters, but the number of row clusters and column clusters are limited to be exactly the same because DOGSM is a k-means based method.

We first consider the case that the number of row clusters and column clusters are presumably given. In Classic3 data set, the number of row cluster $k = 3$ is given and the number of column cluster l is unknown. Hence, different values of $l = 15, 20, 25$ are provided to all of these compared methods. In this situation, OPS calls extended FACA for non-overlapping co-clustering. Besides, DOGSM is a clustering method similar to k-means, it has no parameter l. The metrical scores on Purity and NMI on Classic3 data set are given in Tables 1 and 2 respectively. We observe that, generally speaking, FACA, NMF, SRC and OPS achieve comparative scores over two metrics on Classic3 data set. In detail, OPS outperforms all of the compared methods. Especially, OPS takes more advantage than the other compared methods as the number of overlapping percentage

[2] http://mlg.ucd.ie/datasets

[3] http://mlg.ucd.ie/datasets

$OV\%$ increases. This is because, compared to FACA, NMF and SRC, OPS can discover the overlapping structures which can further reveal the cluster structure of the data set. Even though DOGSM can also discover the overlapping clusters, its performance is not as good as OPS. This is because it can not distinguish the different number of row clusters and column clusters. Besides, compared to other three methods, DOGSM can not make use of the relations between objects and features when performing clustering. This is critical for the performance of clustering when the data is very sparse and noisy. Therefore, OPS gains much better scores on two metrics than DOGSM at different overlapping levels.

Another observation of Tables 1 $\sim$ 2 is that the metrical scores of two metrics on the compared methods decrease as the percentage of overlapping $OV\%$ increasing. The probable reasons for this phenomenon are as follows. Firstly, the higher overlapping percentage of the documents makes the data set more complicated and challenging to all of the compared methods. Secondly, the number of hybrid objects, which belong to two clusters, are increasing as the overlapping percentage getting higher. This makes the number of hybrid objects of the same type, such as hybrid objects of LINE and CISI, enough to form new independent clusters. In other words, the ground truth of the number of the row clusters is moving from 3 to 6 as the number of overlapping documents increasing. However, OPS still performs very well even the overlapping percentage $OV\% = 20\%$.

Table 1. Purity scores on Classic3 data set

OV%	k	l	FACA [1]	NMF [11]	SRC [12]	OPS	DOGSM [16]
5%	3	15	0.9387	0.8971	0.85778	**0.9632**	0.3931
		20	0.9397	0.9107	0.84571	**0.9613**	
		25	0.9444	0.91175	0.8565	**0.9698**	
10%	3	15	0.8882	0.8684	0.8200	**0.9294**	0.3681
		20	0.8870	0.8724	0.7781	**0.9312**	
		25	0.8945	0.8666	0.6857	**0.9367**	
15%	3	15	0.8412	0.8379	0.61014	**0.9113**	0.4052
		20	0.8478	0.8310	0.59362	**0.9136**	
		25	0.8475	0.8307	0.61014	**0.9168**	
20%	3	15	0.7981	0.7933	0.6908	**0.8894**	0.4157
		20	0.8036	0.7936	0.5927	**0.8814**	
		25	0.8067	0.7930	0.5536	**0.8825**	

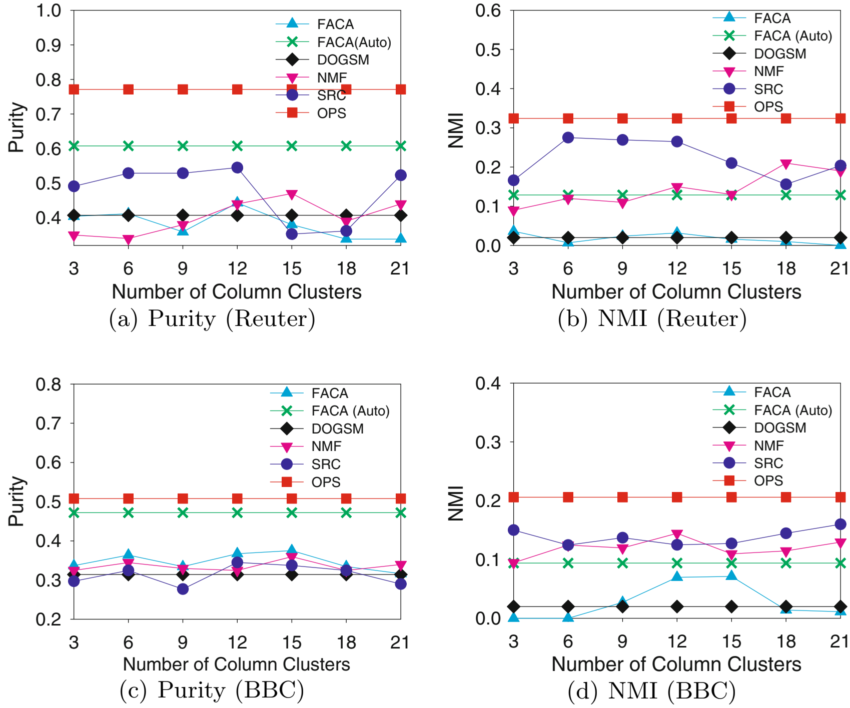
In the real data sets of Reuter and BBC, the number of row clusters $k = 6$ is given, but the number of column clusters is unknown. In order to test the ability of automatically finding the number of row clusters k and column clusters l, we run OPS without given the number of row clusters and column clusters. Since FACA can also automatically detect the number of row clusters and column clusters when searching for the optimal pattern, we use FACA(Auto) to denote this situation. While for the other compared methods, we provide exactly the number of row clusters and different parameters for the number of column clusters.

Table 2. NMI scores on Classic3 data set

OV%	k	l	FACA [1]	NMF [11]	SRC [12]	OPS	DOGSM [16]
5%	3	15	0.8284	0.6995	0.4567	**0.8693**	0.0501
		20	0.8309	0.6060	0.7359	**0.8697**	
		25	0.8506	0.7331	0.5874	**0.8891**	
10%	3	15	0.7345	0.6674	0.4651	**0.7875**	0.0208
		20	0.7296	0.5300	0.6596	**0.7895**	
		25	0.7567	0.6674	0.4850	**0.8009**	
15%	3	15	0.6503	0.6331	0.4375	**0.7270**	0.0615
		20	0.6724	0.4434	0.6146	**0.7526**	
		25	0.6694	0.6118	0.3128	**0.7503**	
20%	3	15	0.5866	0.5562	0.3038	**0.6499**	0.0411
		20	0.6013	0.4190	0.5587	**0.6706**	
		25	0.6077	0.5558	0.3611	**0.6790**	

The results of the different methods on Reuter data set are presented in Figures 2(a)-2(b). Since the results of OPS and FACA(Auto) are not affected by the number of column clusters, their results are horizontal lines over different values of l. Despite without any information of the number of row clusters and column clusters, OPS still gains the highest scores over all of the three metrics. It is evident that OPS has better advantage than other compared methods for finding the most appropriate row clusters and column clusters. Though FACA(Auto) can also detect the number of row clusters and column clusters automatically, its performance is not as good as OPS. That is because of the following reasons. Firstly, OPS can discover the overlapping structures hidden among the clusters. Secondly, OPS uses the total bits used to describe the whole matrix as the objective function, which can get an appropriate balance between model description complexity and code length, and improve the co-clustering quality. We keep in mind that OPS automatically detects the number of row clusters and column clusters. We also notice that FACA(Auto) performs better than FACA in this data set. Besides, SRC outperforms NMF in most of the cases. Though DOGSM can also discover the overlapping structure, it seems very sensitive to the sparsity and noise of the data set. Hence, the performance of DOGSM is relative poor in our tests.

In Figures 2(c)-2(d), we illustrate the results of the compared methods on BBC data set. Once again, OPS gains the highest scores on two different metrics. We note that FACA(Auto) does much poorly on NMI compared to OPS. Moreover, NMF and SRC do poorly on Purity. We observe that the NMI scores of all of the compared methods are not very high. We carefully analyze this phenomenon and find BBC data set is very unbalanced. For example, the number of document annotated as *sports* is 44, while the number of documents annotated as *business* is 102, which is more than two times the number of documents annotated as *sports*. Besides, compared to the Classic3 data set, the number of documents is relatively small, but the number of document clusters is relatively large in this data set. Both of which make the co-clustering in BBC data set a non-trivial



(a) Purity (Reuter)

(b) NMI (Reuter)

(c) Purity (BBC)

(d) NMI (BBC)

Fig. 2. The results of two metrics with different methods on the data sets of Reuter and BBC

challenge for all of these compared methods. However, even in this case, OPS still gains a comparative performance.

6 Conclusion

Discovering the overlapping structures of objects and features simultaneously is significant in many real-world applications. However, this problem is neglected by many existing works. In this paper, a novel parameter-free algorithm OPS, which utilize a density guided principle to discover the overlapping structures among row clusters and column clusters simultaneously, is proposed. Experiments including real-world and synthetic data sets demonstrate our method is effective and efficient. Further works will focus on non-binary matric co-clustering.

Acknowledgements. This work is supported in part by National Natural Science Foundation of China through grants 61271252, NSF through grants CNS-1115234, DBI-0960443, and OISE-1129076.

References

1. Chakrabarti, D., Papadimitriou, S., Modha, D.S., Faloutsos, C.: Fully automatic cross-associations. In: KDD, pp. 79–88 (2004)
2. Cheng, Y., Church, G.M.: Biclustering of expression data. In: ISMB, pp. 93–103 (2000)
3. Deodhar, M., Cho, H., Gupta, G., Ghosh, J., Dhillon, I.: Robust overlapping co-clustering. In: IDEAL-TR (2008)
4. Dhillon, I.S., Guan, Y.: Information theoretic clustering of sparse co-occurrence data. In: ICDM, pp. 517–528 (2003)
5. Evans, T.S., Lambiotte, R.: Line graphs, link partitions and overlapping communities. Physical Review E 80, 016105 (2009)
6. Gossen, T., Kotzyba, M., Nürnberger, A.: Graph clusterings with overlaps: Adapted quality indices and a generation model. Neurocomputing 123, 13–22 (2014)
7. Huang, J., Sun, H., Han, J., Deng, H., Sun, Y., Liu, Y.: Shrink: a structural clustering algorithm for detecting hierarchical communities in networks. In: CIKM, pp. 219–228 (2010)
8. Huh, Y., Kim, J., Lee, J., Yu, K., Shi, W.: Identification of multi-scale corresponding object-set pairs between two polygon datasets with hierarchical co-clustering. ISPRS Journal of Photogrammetry and Remote Sensing 88, 60–68 (2014)
9. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics 11(3), 033015 (2009)
10. Lazzeroni, L., Owen, A.: Plaid models for gene expression data. Statistica Sinica 12, 61–86 (2000)
11. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS, pp. 556–562. MIT Press (2000)
12. Long, B., Zhang, Z., Wu, X., Yu, P.S.: Spectral clustering for multi-type relational data. In: ICML, pp. 585–592 (2006)
13. Long, B., Zhang, Z., Yu, P.S.: Co-clustering by block value decomposition. In: KDD, pp. 635–640 (2005)
14. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
15. Papadimitriou, S., Sun, J., Faloutsos, C., Yu, P.S.: Hierarchical, parameter-free community discovery. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 170–187. Springer, Heidelberg (2008)
16. Wang, X., Tang, L., Gao, H., Liu, H.: Discovering overlapping groups in social media. In: ICDM, pp. 569–578 (December 2010)