Non-Exhaustive, Overlapping Co-Clustering

Joyce Jiyoung Whang
Sungkyunkwan University
Suwon, Korea
jjwhang@skku.edu

Inderjit S. Dhillon
University of Texas at Austin
Austin, TX, USA
inderjit@cs.utexas.edu

ABSTRACT

The goal of co-clustering is to simultaneously identify a clustering of the rows as well as the columns of a two dimensional data matrix. Most existing co-clustering algorithms are designed to find pairwise disjoint and exhaustive co-clusters. However, many real-world datasets might contain not only a large overlap between co-clusters but also outliers which should not belong to any co-cluster. We formulate the problem of Non-Exhaustive, Overlapping Co-Clustering where both of the row and column clusters are allowed to overlap with each other and the outliers for each dimension of the data matrix are not assigned to any cluster. To solve this problem, we propose an intuitive objective function, and develop an efficient iterative algorithm which we call the NEO-CC algorithm. We theoretically show that the NEO-CC algorithm monotonically decreases the proposed objective function. Experimental results show that the NEO-CC algorithm is able to effectively capture the underlying co-clustering structure of real-world data, and thus outperforms state-of-the-art clustering and co-clustering methods.

KEYWORDS

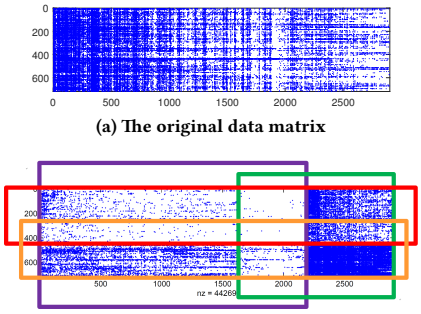
co-clustering; clustering; overlap; outlier; k-means

1 INTRODUCTION

Let us consider a two dimensional data matrix such that each row represents an object and each column represents an attribute or a feature of an object. While one-way clustering algorithms focus on clustering only one of the dimensions of the data matrix, it has been recognized that simultaneously clustering both dimensions of the data matrix is desirable to detect more semantically meaningful clusters in many applications such as gene expression data analysis [9], word-document clustering, and market-basket analysis. Many different kinds of co-clustering methods have been proposed to simultaneously identify a clustering of the rows as well as the columns of the data matrix. However, most existing co-clustering methods are based on an assumption that every object belongs to exactly one row cluster and every attribute belongs to exactly one column cluster. This assumption hinders the existing methods from correctly capturing the underlying structure of data because in many real-world datasets, both of the row and column clusters can overlap with each other and the data might contain outliers.

CIKM'17, November 6–10, 2017, Singapore.

ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: https://doi.org/10.1145/3132847.3133078



(a) The original data matrix

(b) Rearrangement of the rows and the columns according to the output of the NEO-CC method.

Figure 1: Visualization of a user-movie ratings dataset.

We propose the Non-Exhaustive, Overlapping Co-Clustering (NEO-CC) method to efficiently detect coherent co-clusters such that both of the row and column clusters are allowed to overlap with each other and the outliers for each dimension of the data matrix are not assigned to any cluster. For example, Figure 1 shows the output of our NEO-CC method on a user-movie ratings dataset where each row represents a user and each column represents a movie. As shown in Figure 1(a), when we simply visualize the original data matrix, it is hard to discover particular patterns of the matrix. In Figure 1(b), we rearrange the rows and the columns according to the output of the NEO-CC method, and explicitly mark the row and column clusters. Now, we can observe the overlapping co-clustering structure of the data matrix. Also, the NEO-CC method detects one outlier from the rows, and when we look at the detected outlier, it corresponds to a user who randomly gives ratings to a number of movies without any particular pattern.

In this paper, we mathematically formulate the non-exhaustive, overlapping co-clustering problem. To solve this problem, we propose an intuitive objective function, and develop a simple iterative algorithm called the NEO-CC algorithm which monotonically decreases the proposed objective. Experimental results on real-world datasets show that the NEO-CC algorithm is able to effectively capture the underlying co-clustering structure of real-world data, and thus outperforms state-of-the-art clustering and co-clustering methods in terms of discovering the ground-truth clusters.

2 RELATED WORK

We note that [11] and [10] study the overlapping co-clustering problem but do not consider outlier detection. On the other hand, [5] proposes a robust co-clustering algorithm by assuming the presence of outliers, but does not consider overlapping co-clustering. The ROCC algorithm [4] and an infinite plaid model (IPM) [8] have been recently proposed to find non-exhaustive, overlapping co-clusters. However, the ROCC algorithm includes complicated heuristics, and the infinite bi-clustering method requires a user to provide many

non-intuitive hyperparameters with the model. On the other hand, our NEO-CC algorithm is a more principled method with simple and intuitive parameters which can be easily estimated. Part of this work has been presented in the first author's Ph.D. dissertation [12].

For the one-way clustering problem, the NEO-K-Means [13] has been recently proposed to identify overlapping clusters and outliers in a unified manner where the method is designed to cluster only the rows of a data matrix. However, extending this idea to the co-clustering problem is far from straightforward because of the complicated interactions between the rows and the columns of the data matrix where both the cluster overlap and the non-exhaustiveness are allowed for each dimension of the matrix.

3 THE NEO-CC OBJECTIVE FUNCTION

Given a two-dimensional data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$, let $\mathcal{X}^r$ denote the set of data points for row clustering, and $\mathcal{X}^c$ denote the set of data points for column clustering. The co-clustering problem is to cluster $\mathcal{X}^r = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ ($\mathbf{x}_i \in \mathbb{R}^m$ for $i = 1, \ldots, n$) into k row clusters $\{C_1^r, C_2^r, \ldots, C_k^r\}$, and cluster $\mathcal{X}^c = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ ($\mathbf{x}_j \in \mathbb{R}^n$ for $j = 1, \ldots, m$) into l column clusters $\{C_1^c, C_2^c, \ldots, C_l^c\}$.

Inspired by the traditional MSSR co-clustering objective [3], we design our NEO-CC objective function by considering the sum of squared differences between each entry and each mean of the co-clusters the data point belongs to. Let $\boldsymbol{U} = [u_{ij}]_{n \times k}$ denote the assignment matrix for row clustering, i.e., $u_{ij} = 1$ if $\mathbf{x}_i$ belongs to cluster j; $u_{ij} = 0$ otherwise. Similarly, let $\boldsymbol{V} = [v_{ij}]_{m \times l}$ denote the assignment matrix for column clustering. Suppose that we have a small data matrix $\boldsymbol{X} \in \mathbb{R}^{4 \times 5}$ and the assignment matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ as shown in Figure 2(a). For an entry x_{21}, Figure 2(b) shows the contribution of the entry x_{21} to the NEO-CC objective when $\mathbf{x}_2^r \in C_1^r$, $\mathbf{x}_2^r \in C_2^r$ ($\mathbf{x}_2^r \in \mathbb{R}^5$), $\mathbf{x}_1^c \in C_1^c$, $\mathbf{x}_1^c \in C_2^c$ ($\mathbf{x}_1^c \in \mathbb{R}^4$). For the entry x_{21}, the NEO-CC objective considers the squared differences between x_{21} and four different means, each of which corresponds to a different combination of the row and column clusters the entry belongs to. Now, let us represent this idea using matrices and vectors. Let $\hat{\boldsymbol{U}} = [\frac{\mathbf{u}_1}{\sqrt{n_1}}, \cdots, \frac{\mathbf{u}_k}{\sqrt{n_k}}]$ denote a normalized assignment matrix where $\mathbf{u}_c$ is the c-th column of $\boldsymbol{U}$ and n_c is the size of cluster c. Let $\hat{\mathbf{u}}_i$ denote the i-th column of $\hat{\boldsymbol{U}}$. Similarly, we also define $\hat{\boldsymbol{V}}$ and let $\hat{\mathbf{v}}_j$ denote the j-th column of $\hat{\boldsymbol{V}}$. Let $\mathbb{I}\{exp\} = 1$ if exp is true; 0 otherwise, and let $\mathbf{1}$ denote a vector having all the elements equal to one. Finally, given a vector $\mathbf{y} \in \mathbb{R}^m$, let us define $D(\mathbf{y}) = [d_{ij}]_{m \times m}$ as the diagonal matrix with $d_{ii} = y_i$ ($i = 1, \ldots, m$). Then, our NEO-CC objective function is:

$$\begin{aligned} \underset{\boldsymbol{U}, \boldsymbol{V}}{\text{minimize}} \quad & \sum_{i=1}^{k} \sum_{j=1}^{l} \| D(\mathbf{u}_i) \boldsymbol{X} D(\mathbf{v}_j) - \hat{\mathbf{u}}_i \hat{\mathbf{u}}_i^T \boldsymbol{X} \hat{\mathbf{v}}_j \hat{\mathbf{v}}_j^T \|_F^2 \\ \text{subject to} \quad & \text{trace}(\boldsymbol{U}^T \boldsymbol{U}) = (1 + \alpha_r) n, \\ & \textstyle\sum_{i=1}^{n} \mathbb{I}\{(\boldsymbol{U}\mathbf{1})_i = 0\} \leq \beta_r n, \\ & \text{trace}(\boldsymbol{V}^T \boldsymbol{V}) = (1 + \alpha_c) m, \\ & \textstyle\sum_{i=1}^{m} \mathbb{I}\{(\boldsymbol{V}\mathbf{1})_i = 0\} \leq \beta_c m, \end{aligned} \tag{1}$$

where α_r and β_r are the parameters for row clustering, and α_c and β_c are the parameters for column clustering. The parameters α_r and α_c control the amount of overlap among the clusters while β_r and β_c control the degree of non-exhaustiveness. We introduce these parameters motivated by the one-way NEO-K-Means method [13].

The first two constraints in (1) are associated with the row clustering whereas the last two constraints are associated with the column clustering. The first constraint indicates that the total number of assignments in $\boldsymbol{U}$ is equal to $(1 + \alpha_r)n$. Thus, more than n assignments are made in $\boldsymbol{U}$ for $\alpha_r > 0$, which implies that some data points belong to more than one cluster. The second constraint indicates the upper bound of the number of outliers, i.e., there can be at most $\beta_r n$ outliers. We can similarly interpret the last two constraints for the column clustering. The NEO-CC objective seamlessly generalizes the NEO-K-Means [13] and the MSSR [3] objectives. If $\boldsymbol{V} = \boldsymbol{I}$, $\alpha_c = 0$, $\beta_c = 0$, then (1) is equivalent to the NEO-K-Means objective. If $\alpha_r = 0$, $\alpha_c = 0$, $\beta_r = 0$, $\beta_c = 0$, then (1) is equivalent to the MSSR objective.

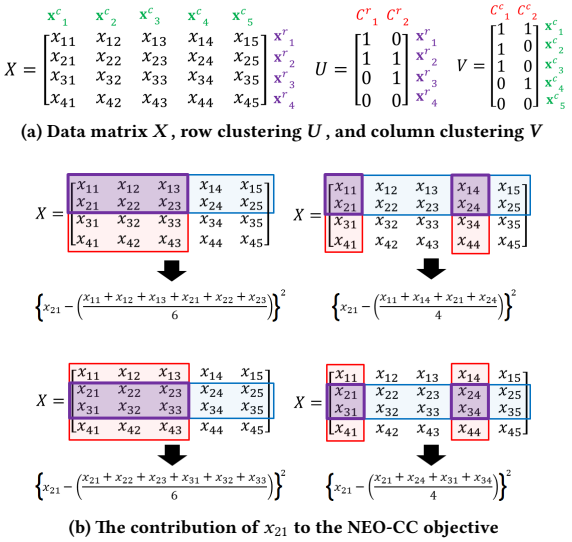


Figure 2: The NEO-CC objective considers the differences between each entry and the co-cluster means the entry belongs to.

4 THE NEO-CC ALGORITHM

To optimize our NEO-CC objective function, we develop an iterative NEO-CC algorithm which we describe in Algorithm 1. The NEO-CC algorithm repeatedly updates $\boldsymbol{U}$ and $\boldsymbol{V}$ until the change in the objective becomes sufficiently small or the maximum number of iterations is reached. Within each iteration, $\boldsymbol{U}$ and $\boldsymbol{V}$ are alternatively updated. We initialize $\boldsymbol{U}$ and $\boldsymbol{V}$ by running the one-way NEO-K-Means clustering algorithm on the data matrix and the transpose of the matrix, respectively. We also estimate the parameters α_r, β_r, α_c, and β_c using the strategies suggested in [13].

Algorithm 1 consists of two main parts – updating row clustering (lines 3–20) and updating column clustering (lines 21–38). Let us first describe how $\boldsymbol{U}$ is updated. To update $\boldsymbol{U}$, we need to compute distances between every data point in $\mathcal{X}^r$ and the clusters C_q^r for $q = 1, \cdots, k$. Let $[d_{pq}^r]_{n \times k}$ denote these distances, and let $\boldsymbol{I}_{p\cdot}$ denote the p-th row of the identity matrix of size n. The distance between a data point $\mathbf{x}_p \in \mathcal{X}^r$ and a cluster C_q^r is computed by

$$d_{pq}^r = \sum_{j=1}^{l} \left\| (\boldsymbol{I}_{p\cdot}) \boldsymbol{X} D(\mathbf{v}_j) - \frac{1}{\sqrt{\|\mathbf{u}_q\|_1}} \hat{\mathbf{u}}_q^T \boldsymbol{X} \hat{\mathbf{v}}_j \hat{\mathbf{v}}_j^T \right\|^2 . \tag{2}$$

For each data point $\mathbf{x}_p \in \mathcal{X}^r$ ($p = 1, \cdots, n$), we record its closest cluster and that distance. By sorting the data points in ascending order by the distance to its closest cluster, we assign the first $(1 - \beta_r)n$ data points to their closest clusters to satisfy the second constraint in (1). Then, we make $\alpha_r n + \beta_r n$ assignments by taking $\alpha_r n + \beta_r n$

Algorithm 1 NEO-CC Algorithm

Input: $\boldsymbol{X} \in \mathbb{R}^{n\times m}$, k, l, α_r, α_c, β_r, β_c, t_{max}
Output: Row clustering $\boldsymbol{U} \in \{0,1\}^{n\times k}$, Column clustering $\boldsymbol{V} \in \{0,1\}^{m\times l}$
1: Initialize $\boldsymbol{U}$, $\boldsymbol{V}$, and $t = 0$.
2: **while** not converged and $t < t_{max}$ **do**
3: /* Update Row Clustering */
4: **for each** $\mathbf{x}_p \in \mathcal{X}^r$ **do**
5: **for** $q = 1, \cdots, k$ **do**
6: Compute d^r_{pq} using (2).
7: **end for**
8: **end for**
9: Initialize $w = 0$, $\mathcal{T} = \emptyset$, $\mathcal{S} = \emptyset$, and $\bar{C}^r_i = \emptyset$, $\hat{C}^r_i = \emptyset$ $\forall i$ $(i = 1, \cdots, k)$.
10: **while** $w < (n + \alpha_r n)$ **do**
11: **if** $w < (n - \beta_r n)$ **then**
12: Assign $\mathbf{x}^r_{i^*}$ to $\bar{C}^r_{j^*}$ such that $(i^*, j^*) = \underset{i,j}{\operatorname{argmin}}\, d^r_{ij}$ where $\{(i,j)\} \notin \mathcal{T}$, $i \notin \mathcal{S}$.
13: $\mathcal{S} = \mathcal{S} \cup \{i^*\}$.
14: **else**
15: Assign $\mathbf{x}^r_{i^*}$ to $\hat{C}^r_{j^*}$ such that $(i^*, j^*) = \underset{i,j}{\operatorname{argmin}}\, d^r_{ij}$ where $\{(i,j)\} \notin \mathcal{T}$.
16: **end if**
17: $\mathcal{T} = \{(i^*, j^*)\} \cup \mathcal{T}$.
18: $w = w + 1$.
19: **end while**
20: Update clusters $C^r_i = \bar{C}^r_i \cup \hat{C}^r_i$ $\forall i$ $(i = 1, \cdots, k)$.
21: /* Update Column Clustering */
22: **for each** $\mathbf{x}_p \in \mathcal{X}^c$ **do**
23: **for** $q = 1, \cdots, l$ **do**
24: Compute d^c_{pq} using (3).
25: **end for**
26: **end for**
27: Initialize $w = 0$, $\mathcal{T} = \emptyset$, $\mathcal{S} = \emptyset$, and $\bar{C}^c_j = \emptyset$, $\hat{C}^c_j = \emptyset$ $\forall j$ $(j = 1, \cdots, l)$.
28: **while** $w < (m + \alpha_c m)$ **do**
29: **if** $w < (m - \beta_c m)$ **then**
30: Assign $\mathbf{x}^c_{i^*}$ to $\bar{C}^c_{j^*}$ such that $(i^*, j^*) = \underset{i,j}{\operatorname{argmin}}\, d^c_{ij}$ where $\{(i,j)\} \notin \mathcal{T}$, $i \notin \mathcal{S}$.
31: $\mathcal{S} = \mathcal{S} \cup \{i^*\}$.
32: **else**
33: Assign $\mathbf{x}^c_{i^*}$ to $\hat{C}^c_{j^*}$ such that $(i^*, j^*) = \underset{i,j}{\operatorname{argmin}}\, d^c_{ij}$ where $\{(i,j)\} \notin \mathcal{T}$.
34: **end if**
35: $\mathcal{T} = \{(i^*, j^*)\} \cup \mathcal{T}$.
36: $w = w + 1$.
37: **end while**
38: Update clusters $C^c_j = \bar{C}^c_j \cup \hat{C}^c_j$ $\forall j$ $(j = 1, \cdots, l)$.
39: $t = t + 1$.
40: **end while**

minimum distances among $[d^r_{pq}]_{n\times k}$. Note that, in total, we make $n + \alpha_r n$ assignments in $\boldsymbol{U}$, which satisfies the first constraint in (1).

Similarly, we can also update $\boldsymbol{V}$. Let $\boldsymbol{I}_{\cdot p}$ denote the p-th column of the identity matrix of size m. The distance between a data point $\mathbf{x}_p \in \mathcal{X}^c$ and a column cluster C^c_q is computed by

$$d^c_{pq} = \sum_{i=1}^{k} \left\| D(\mathbf{u}_i)\boldsymbol{X}(\boldsymbol{I}_{\cdot p}) - \frac{1}{\sqrt{\|\mathbf{v}_q\|_1}}\hat{\mathbf{u}}_i\hat{\mathbf{u}}_i^T\boldsymbol{X}\hat{\mathbf{v}}_q \right\|^2 . \tag{3}$$

After computing the distances between every data point in $\mathcal{X}^c$ and the column clusters using (3), $\boldsymbol{V}$ can be similarly updated.

Now, we show the monotonic decrease of the NEO-CC objective in Algorithm 1 (Theorem 1) using the following lemma.

Lemma 1. *Let us consider the function* $h(\mathbf{z}) = \sum_i \pi_i \|\mathbf{a}_i - c\mathbf{z}\boldsymbol{M}\|_2^2$ *where* $\mathbf{a}_i \in \mathbb{R}^{1\times m}$, $\mathbf{z} \in \mathbb{R}^{1\times m}$, $\pi_i > 0$ $\forall i$, $c = \frac{1}{\sqrt{\sum_i \pi_i}}$, *and* $\boldsymbol{M} \in \mathbb{R}^{m\times m}$ *such that* $\boldsymbol{M}\boldsymbol{M}^T = \boldsymbol{M}$. *Let* $\mathbf{z}^*$ *denote the minimizer of* $h(\mathbf{z})$. *Then,* $\mathbf{z}^*$ *is given by* $\Big(\sqrt{\sum_i \pi_i}\Big)\boldsymbol{M}\mathbf{z}^{*T} = \boldsymbol{M}\Big(\sum_i \pi_i \mathbf{a}_i^T\Big)$.

Proof. We can express $h(\mathbf{z})$ as follows:

$$h(\mathbf{z}) = \sum_i \pi_i\big(\mathbf{a}_i\mathbf{a}_i^T - 2c\mathbf{z}\boldsymbol{M}\mathbf{a}_i^T + c^2\mathbf{z}\boldsymbol{M}\boldsymbol{M}^T\mathbf{z}^T\big),$$

and the gradient is given by

$$\frac{\partial h(\mathbf{z})}{\partial \mathbf{z}} = \sum_i \pi_i\big(-2c\boldsymbol{M}\mathbf{a}_i^T + 2c^2\boldsymbol{M}\boldsymbol{M}^T\mathbf{z}^T\big).$$

By setting the gradient to zero, we get

$$\sum_i \pi_i \boldsymbol{M}\mathbf{a}_i^T = c\Big(\sum_i \pi_i\Big)\boldsymbol{M}\boldsymbol{M}^T\mathbf{z}^{*T} = c\Big(\sum_i \pi_i\Big)\boldsymbol{M}\mathbf{z}^{*T} \quad \text{since } \boldsymbol{M}\boldsymbol{M}^T = \boldsymbol{M}$$

By setting $c = \frac{1}{\sqrt{\sum_i \pi_i}}$, we get

$$\Big(\sqrt{\sum_i \pi_i}\Big)\boldsymbol{M}\mathbf{z}^{*T} = \boldsymbol{M}\Big(\sum_i \pi_i \mathbf{a}_i^T\Big). \qquad \square$$

Theorem 1. *Algorithm 1 monotonically decreases the NEO-CC objective function defined in (1).*

Proof. Let $J^{(t)}$ denote the NEO-CC-M objective (1) at t-th iteration. Let $\boldsymbol{U}$ denote the assignment matrix of the current row clustering C, and $\boldsymbol{U}^*$ denote the assignment matrix of the updated row clustering C^* obtained by line 20 in Algorithm 1.

$$\begin{aligned}
J^{(t)} &= \sum_{i=1}^{k}\sum_{j=1}^{l}\left\|D(\mathbf{u}_i)\boldsymbol{X}D(\mathbf{v}_j) - \hat{\mathbf{u}}_i\hat{\mathbf{u}}_i^T\boldsymbol{X}\hat{\mathbf{v}}_j\hat{\mathbf{v}}_j^T\right\|_F^2 \\
&= \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{\mathbf{x}_p\in C_i}\left\|(\boldsymbol{I}_{p\cdot})\boldsymbol{X}D(\mathbf{v}_j) - \frac{1}{\sqrt{\|\mathbf{u}_i\|_1}}\hat{\mathbf{u}}_i^T\boldsymbol{X}\hat{\mathbf{v}}_j\hat{\mathbf{v}}_j^T\right\|_2^2 \\
&\geq \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{\mathbf{x}_p\in C^*_i}\left\|(\boldsymbol{I}_{p\cdot})\boldsymbol{X}D(\mathbf{v}_j) - \frac{1}{\sqrt{\|\mathbf{u}_i\|_1}}\hat{\mathbf{u}}_i^T\boldsymbol{X}\hat{\mathbf{v}}_j\hat{\mathbf{v}}_j^T\right\|_2^2 \\
&\geq \sum_{i=1}^{k}\sum_{j=1}^{l}\sum_{\mathbf{x}_p\in C^*_i}\left\|(\boldsymbol{I}_{p\cdot})\boldsymbol{X}D(\mathbf{v}_j) - \frac{1}{\sqrt{\|\mathbf{u}^*{}_i\|_1}}\hat{\mathbf{u}^*}_i^T\boldsymbol{X}\hat{\mathbf{v}}_j\hat{\mathbf{v}}_j^T\right\|_2^2 \\
&= \sum_{i=1}^{k}\sum_{j=1}^{l}\left\|D(\mathbf{u}^*{}_i)\boldsymbol{X}D(\mathbf{v}_j) - \hat{\mathbf{u}^*}_i\hat{\mathbf{u}^*}_i^T\boldsymbol{X}\hat{\mathbf{v}}_j\hat{\mathbf{v}}_j^T\right\|_F^2 \\
&\geq \sum_{i=1}^{k}\sum_{j=1}^{l}\left\|D(\mathbf{u}^*{}_i)\boldsymbol{X}D(\mathbf{v}^*{}_j) - \hat{\mathbf{u}^*}_i\hat{\mathbf{u}^*}_i^T\boldsymbol{X}\hat{\mathbf{v}^*}_j\hat{\mathbf{v}^*}_j^T\right\|_F^2 \\
&= J^{(t+1)}
\end{aligned}$$

The first inequality holds because we make assignments by line 12 & line 15, and the second inequality holds by Lemma 1 with $\mathbf{a}_i = (\boldsymbol{I}_{p\cdot})\boldsymbol{X}D(\mathbf{v}_j)$, $\boldsymbol{M} = \hat{\mathbf{v}}_j\hat{\mathbf{v}}_j^T$, $\mathbf{z}^* = \hat{\mathbf{u}^*}_i^T\boldsymbol{X}$, and $\sqrt{\sum_i \pi_i} = \sqrt{\|\mathbf{u}^*{}_i\|_1}$. The last inequality indicates that we can similarly show the decrease from $\boldsymbol{V}$ to $\boldsymbol{V}^*$. $\square$

5 EXPERIMENTAL RESULTS

We compare the performance of our NEO-CC algorithm with other state-of-the-art co-clustering and one-way clustering algorithms: IPM [8], ROCC [4], MSSR [3], NEO-iter [13], and NEO-lrsdp [7]. We repeat the experiments for 5 times on each dataset, and use the default parameters for the baseline methods. We compute F_1 scores to compare the algorithmic clusters and the ground-truth clusters [13]. Higher F_1 score indicates better clusters.

We first test the clustering performance on user-movie ratings datasets from MovieLens[1] where we know the genres of the movies (e.g., action, romance, comedy, etc.). We can consider the genres as the ground-truth clusters [1]. Since a movie usually belongs to multiple genres, there exist overlaps among the clusters. We have two different data matrices – ML1 contains 44,269 ratings, and ML2 contains 28,335 ratings. In Table 1, we show the F_1 scores (%) of

[1] http://grouplens.org/datasets/movielens/

Table 1: F_1 scores (%). The NEO-CC algorithm achieves higher F_1 scores than other methods.

		IPM	ROCC	MSSR1	MSSR2	NEO-iter	NEO-lrsdp	NEO-CC
ML1	average	22.4	55.7	43.8	44.2	56.3	56.4	**58.1**
ML2	best	36.2	53.3	50.6	50.5	56.8	56.8	**58.8**
	worst	18.6	53.3	50.2	48.2	56.8	56.8	**58.1**
	average	26.7	53.3	50.5	49.4	56.8	56.8	**58.4**
Yeast	best	N/A	15.0	17.4	19.3	36.6	39.1	**40.7**
	worst	N/A	12.8	16.4	18.0	35.6	**39.0**	36.2
	average	N/A	14.3	16.9	18.5	36.0	39.1	**40.0**
Facebook	best	N/A	26.9	30.6	31.8	34.7	37.6	**37.7**
	worst	N/A	24.0	28.7	27.3	33.3	33.7	**37.1**
	average	N/A	25.2	29.7	29.7	33.9	35.9	**37.3**

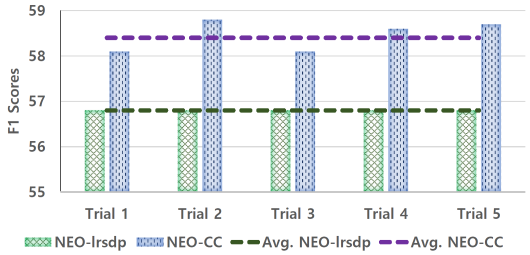


Figure 3: F_1 scores of the best baseline method (NEO-lrsdp) and the NEO-CC method on the ML2 dataset.

each method (the best, the worst, and the average scores). On ML1 dataset, all the methods except IPM produce identical results for the five trials. Figure 3 shows the F_1 scores of the best baseline method (NEO-lrsdp) and the NEO-CC algorithm for each trial on the ML2 dataset. We see that the NEO-CC algorithm outperforms all the baseline methods.

We get an yeast gene expression dataset from [6] where each row represents a gene, and each column represents an expression level under a particular biological condition. By clustering or co-clustering this gene expression data, we can group genes with similar functions [2]. Indeed, each gene can belong to multiple functional classes, and we can treat each functional class as a ground-truth cluster. There are 2,417 genes, 103 features, and 14 functional classes. In Table 1, we see that the performance of MSSR is not good because the dataset contains a large overlap among the clusters (MSSR generates pairwise disjoint co-clusters). The IPM method failed to process this dataset. Note that the NEO-* methods significantly outperform the other methods (IPM, ROCC, MSSR1, and MSSR2). We observed that among five trials, NEO-CC outperforms NEO-lrsdp four times. It is interesting to see that the performance of NEO-CC is even better than NEO-lrsdp because we know that the NEO-lrsdp method involves much more expensive operations than the NEO-CC method which is a simple iterative algorithm. We expect that we can further improve the performance of the NEO-CC algorithm by adapting an SDP-based approach.

On a social network, a community can be interpreted as a cluster. Since an individual tends to belong to multiple communities, it is likely that the communities are overlapped with each other. From SNAP[2], we get an ego network (which contains 171 nodes and 1,826 undirected edges), the attributes of the nodes (63 attributes), and the ground-truth communities ($k = 14$) on Facebook. By concatenating the adjacency matrix and the attribute matrix, we get the data matrix. As can be seen in the last row of Table 1, overall, NEO-CC shows the best performance (the IPM method failed to process this dataset). Co-clustering enables us to perform an implicit dimensionality reduction, which leads to performing an implicit regularized clustering. This can be a explanation why the F_1 score of NEO-CC can be even higher than the NEO-lrsdp method.

[2]http://snap.stanford.edu/

6 CONCLUSIONS & FUTURE WORK

The NEO-CC method provides a principled way to effectively capture the underlying co-clustering structure of many different types of real-world data. We plan to investigate a low-rank semi-definite programming for the NEO-CC method to develop a more sophisticated algorithm and further improve the performance.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the NRF of Korea funded by MOE(2016R1D1A1B03934766 and NRF-2010-0020210) and by the National Program for Excellence in SW supervised by the IITP(2015-0-00914), Korea to JW, and by NSF grants CCF-1320746 and IIS-1546452 to ID. J. Whang is the corresponding author.

REFERENCES

[1] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. J. Mooney. 2005. Model-based Overlapping Clustering. In *KDD*. 532–537.
[2] Y. Cheng and G. Church. 2000. Biclustering of expression data. In *ISMB*. 93–103.
[3] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra. 2004. Minimum Sum-Squared Residue Co-clustering of Gene Expression Data. In *SDM*. 114–125.
[4] M. Deodhar, G. Gupta, J. Ghosh, H. Cho, and I. S. Dhillon. 2009. A Scalable Framework for Discovering Coherent Co-clusters in Noisy Data. In *ICML*.
[5] L. Du and Y. Shen. 2013. Towards Robust Co-clustering. In *IJCAI*. 1317–1322.
[6] A. Elisseeff and J. Weston. 2001. A Kernel Method for Multi-Labelled Classification. In *NIPS*. 681–687.
[7] Y. Hou, J. J. Whang, D. F. Gleich, and I. S. Dhillon. 2015. Non-exhaustive, Overlapping Clustering via Low-Rank Semidefinite Programming. In *KDD*. 427–436.
[8] K. Ishiguro, I. Sato, M. Nakano, A. Kimura, and N. Ueda. 2016. Infinite Plaid Models for Infinite Bi-Clustering. In *AAAI*. 1701–1708.
[9] G. Pio, M. Ceci, D. Malerba, and D. D'Elia. 2015. ComiRNet: A web-based system for the analysis of miRNA-gene regulatory networks. In *BMC Bioinformatics*.
[10] M. M. Shafiei and E. E. Milios. 2006. Latent Dirichlet Co-Clustering. In *ICDM*.
[11] H. Shan and A. Banerjee. 2008. Bayesian Co-clustering. In *ICDM*. 530–539.
[12] J. J. Whang. 2015. Overlapping community detection in massive social networks. In *UT Electronic Theses and Dissertations*.
[13] J. J. Whang, I. S. Dhillon, and D. F. Gleich. 2015. Non-exhaustive, Overlapping k-means. In *SDM*. 936–944.