Bipartite Edge Correlation Clustering: Finding an Edge Biclique Partition from a Bipartite Graph with Minimum Disagreement

Mikio Mizukami[1], Kouich Hirata[1] and Tetsuji Kuboyama[2]

[1]*Kyushu Institute of Technology, Kawazu 680-4, Iizuka 820-8502, Japan*

[2]*Gakushuin University, Mejiro 1-5-1, Toshima, Tokyo 171-8588, Japan*

Keywords: Bipartite Edge Correlation Clustering, Edge Biclique Partition, Minimum Disagreement, Bipartite Correlation Clustering, Bicluster Graph Editing, Biclustering.

Abstract: In this paper, first we formulate the problem of a *bipartite edge correlation clustering* which finds an *edge biclique partition* with the minimum disagreement from a bipartite graph, by extending the bipartite correlation clustering which finds a biclique partition. Then, we design a simple randomized algorithm for bipartite edge correlation clustering, based on the randomized algorithm of bipartite correlation clustering. Finally, we give experimental results to evaluate the algorithms from both artificial data and real data.

1 INTRODUCTION

The notion of *biclustering* has first introduced by Cheng and Chruch (Cheng and Church, 2000) in the context of computational biology or bioinformatics and developed by several researchers with many alternative formulations and different applications and approaches, see (Madeira and Oliveira, 2004; Oghabian et al., 2014; Pio et al., 2013; Pio et al., 2015) and their references, for example. The biclustering performs simultaneous row-column clustering from a matrix. In other words, it finds a *bicluster* as a subset of rows and a subset of columns, defining together a submatrix that shows unique, similar expression patterns according to some sorting method.

In this paper, as combinatorial optimization rather than computational biology, we focus on the formulation of the biclustering by regarding matrices as *bipartite graphs*. In this formulation, we call the biclustering the *bipartite correlation clustering* (Alion et al., 2012; Asteris et al., 2016) or *bicluster graph editing* (Amit, 2004), which is the problem to find the collection of bicliques as a *biclique partition* from a bipartite graph with *the minimum disagreement*. Here, a *biclique* in a bipartite graph is a set of vertices such that every left vertex is adjacent to every right vertex. Also a *disagreement* is the number of edges when constructs a biclique if added or when divides two bicliques if removed.

In this context, Amit (Amit, 2004) has first shown that the bipartite correlation clustering with the minimum disagreement is NP-hard and provided a polynomial-time algorithm that guarantees an approximation factor of 11. Also Alion *et al.* (Alion et al., 2012) have designed both the deterministic and the randomized algorithms of the bipartite correlation clustering whose expected value of the disagreement is at most 4 times of the optimum solution. Furthermore, Asteris *et al.* (Asteris et al., 2016) have shown a PTAS when adopting *the maximum agreement*, not the minimum disagreement. In particular, since the randomized algorithm, called PIVOTBICLUSTER (Alion et al., 2012), is simple and runs efficiently, in this paper, first we focus on this algorithm.

Note that, when applying this algorithm to real problems, it tends to construct many *singletons*, that is, bicliques consisting of a single vertex adjacent to no vertices. On the other hand, when we find some communities in community detection from a bipartite graph, the purpose is find subgraphs with high density of edges. Also, a biclique partition in bipartite correlation clustering consists of biclusters such that one vertex belongs to just one bicluster as a community. Then, in order to achieve the purpose, we prefer to extract bicluster with exclusive edges rather than exclusive vertices in bipartite correlation clustering.

As more appropriate setting to achieve the purpose, an *edge biclique* as a set of edges has been researched from the viewpoint of combinatorial optimization in bipartite graphs (Chalermsook et al., 2014; Chandran et al., 2016; Orlin, 1977). It is known that the problem of finding an *edge biclique partition*

Mizukami, M., Hirata, K. and Kuboyama, T.
Bipartite Edge Correlation Clustering: Finding an Edge Biclique Partition from a Bipartite Graph with Minimum Disagreement.
DOI: 10.5220/0007471506990706
In *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019)*, pages 699-706
ISBN: 978-989-758-351-3

as the collection of edge bicliques with the minimum cardinality from a bipartite graph is NP-hard (Jiang and Raviunar, 1993) and hard to approximate as graph coloring (Chalermsook et al., 2014).

In this paper, by extending the bipartite correlation clustering, we formulate the *bipartite edge correlation clustering* which finds an edge biclique partition with the minimum disagreement from a labeled complete bipartite graph. As similar as the bipartite correlation clustering, we can formulate the input of the bipartite edge correlation clustering as a (non-labeled non-complete) bipartite graph.

Then, by improving the algorithm PIVOTBICLUSTER, we design the randomized algorithm PIVOTBICLUSTEREDGE of the bipartite edge correlation clustering, which outputs no singletons. Also, in this paper, we design the deterministic versions of PIVOTBICLUSTER and PIVOTBICLUSTEREDGE, where the former outputs no singletons.

Finally, we give experimental results of evaluating the algorithms in order to compare bipartite edge correlation clustering with bipartite correlation clustering and to confirm the probabilistic execution. We use two kinds of data, one is artificial data and another is real data of not only MovieLens datasets[1] discussed in (Asteris et al., 2016) but also Crime (MC)[2], Sexual escorts (SX)[2], arXiv cond-mat (AC)[2], Jester 100 (J1)[2] and YouTube (YG)[2] provided from KONECT[3].

2 BIPARTITE CORRELATION CLUSTERING

Let $G=(L,R,E)$ be a bipartite graph. We say that $C_i=(L_i,R_i,E_i)$ is a *biclique* in G if $L_i \subseteq L$, $R_i \subseteq R$, $E_i \subseteq E$ and $(l,r) \in E_i$ for every $l \in L_i$ and $r \in R_i$. Note that a *singleton* either $(\{l\},\emptyset,\emptyset)$ or $(\emptyset,\{r\},\emptyset)$ is always a biclique.

Definition 1. Let $\mathcal{C}=\{C_1,\dots,C_k\}$ be a collection of bicliques in G. We say that $\mathcal{C}$ is a *biclique partition* of G if

1. $\bigcup_{i=1}^{k} L_i = L$, $\bigcup_{i=1}^{k} R_i = R$ and
2. $L_i \cap L_j = \emptyset$ and $R_i \cap R_j = \emptyset$ for every i,j ($1 \leq i,j \leq k, i \neq j$).

Note that E is not required to coincide with $\bigcup_{i=1}^{k} E_i$.

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

[2] http://konect.uni-koblenz.de/networks/ {monero_crime, escorts, opshal-collaboration, jester1, youtube-groupmemberships}

[3] http://konect.uni-koblenz.de

Example 1. Let $G=(L,R,E)$ and C_i ($1 \leq i \leq 4$) be the following bipartite graph and bicliques.

$$\begin{aligned} L &= \{1,2\}, R=\{a,b\},\\ E &= \{(1,a),(1,b),(2,b)\},\\ C_1 &= (\{1\},\{a,b\},\{(1,a),(1,b)\}),\\ C_2 &= (\{2\},\emptyset,\emptyset),\\ C_3 &= (\{1\},\{a\},\{(1,a)\}),\\ C_4 &= (\{2\},\{b\},\{(2,b)\}). \end{aligned}$$

Then, both $\{C_1,C_2\}$ and $\{C_3,C_4\}$ are biclique partitions of G.

Let $G=(L,R,E)$ be a complete bipartite graph such that every edge $e \in E$ is assigned to a label $l(e)$ of either 1 (positive) or -1 (negative), which we call a *labeled complete bipartite graph*. Let $\mathcal{C}=\{C_1,\dots,C_k\}$ be a biclique partition of G, where $C_i=(L_i,R_i,E_i)$. Also let $E_{\mathcal{C}}^{+}=\bigcup_{i=1}^{k} E_i$ and $E_{\mathcal{C}}^{-}=E \setminus E_{\mathcal{C}}^{+}$. Then, the *disagreement* $da_G(\mathcal{C})$ of $\mathcal{C}$ in G is defined as follows.

$$\begin{aligned} & da_G(\mathcal{C}) \\ & = |\{e \in E_{\mathcal{C}}^{+} \mid l(e)=-1\}| + |\{e \in E_{\mathcal{C}}^{-} \mid l(e)=1\}|. \end{aligned}$$

Definition 2 (Amit (Amit, 2004))**.** The problem BICOCLUST of *bipartite correlation clustering* (or *bicluster graph editing*) is defined as follows.

> BICOCLUST
> INSTANCE: A labeled complete bipartite graph $G=(L,R,E)$.
> SOLUTION: Find a biclique partition $\mathcal{C}$ of G such that $da_G(\mathcal{C})$ is minimum.

We can formulate the problem BICOCLUST by using a non-labeled non-complete bipartite graph $G=(L,R,E)$. When G is given as an instance the problem BICOCLUST, we assume that every element in E (*resp.*, $(L \times R) \setminus E$) is assigned to 1 (*resp.*, -1). Then, we regard a biclique partition of a labeled complete bipartite graph as a partition of a bipartite graph containing non-bicliques, which we also call a *cluster* or a *bicluster*.

Amit (Amit, 2004) has first shown that the problems of BICOCLUST is NP-hard. Furthermore, Alion *et al.* (Alion et al., 2012) have designed both a deterministic and a randomized algorithms of BICOCLUST that output the biclique partition whose expected value of the disagreement is at most 4 times of the optimum solution.

Algorithm 1 illustrates the randomized algorithm PIVOTBICLUSTER (Alion et al., 2012) which guarantees probabilistic 4-approximation of the problem of BICOCLUST.

As related works to the problem BICOCLUST, Asteris *et al.* (Asteris et al., 2016) have discussed the

procedure PIVOTBICLUSTER(G)
/* $G = (L, R, E)$: bipartite graph */
1 $\Gamma \leftarrow \emptyset$; /* Γ: set of clusters */
2 **while** $L \neq \emptyset$ **do**
3 **select** $l_1 \in L$ uniform randomly;
$C \leftarrow \{l_1\} \cup N(l_1)$; /* C: cluster */
4 $L' \leftarrow L \setminus \{l_1\}$; $R' \leftarrow R \setminus N(l_1)$;
5 **foreach** $l_2 \in L \setminus \{l_1\}$ **do**
6 $R_1 \leftarrow N(l_1) \setminus N(l_2)$;
$R_2 \leftarrow N(l_2) \setminus N(l_1)$;
$R_{1,2} \leftarrow N(l_1) \cap N(l_2)$;
7 With probability $\min\left\{\frac{|R_{1,2}|}{|R_2|}, 1\right\}$ **do**
8 **begin**
9 **if** $|R_{1,2}| \geq |R_1|$ **then** $C \leftarrow C \cup \{l_2\}$;
else $\Gamma \leftarrow \Gamma \cup \{\{l_2\}\}$;
10 $L' \leftarrow L' \setminus \{l_2\}$;
11 **end**
12 $\Gamma \leftarrow \Gamma \cup \{C\}$; $L \leftarrow L'$; $R \leftarrow R'$;
13 $\Gamma \leftarrow \Gamma \cup \{\{r\} \mid r \in R\}$;
14 **output** Γ;

Algorithm 1: PIVOTBICLUSTER (Alion et al., 2012).

problem of finding a biclique partition with maximizing the *agreement* $ag_G(\mathcal{C})$, not minimizing the disagreement, where:

$$ag_G(\mathcal{C}) = |\{e \in E^+_{\mathcal{C}} \mid l(e) = 1\}| + |\{e \in E^-_{\mathcal{C}} \mid l(e) = -1\}|.$$

Then, they have shown that this problem has a PTAS (Asteris et al., 2016). We will use the agreement in Section 4.

3 BIPARTITE EDGE CORRELATION CLUSTERING

In this paper, we focus on bipartite correlation clustering based on *edge bicliques* as a set of edges, not bicliques as a set of vertices, which we call *bicluster edge correlation clustering*.

Definition 3. Let $\mathcal{C} = \{C_1, \dots, C_k\}$ be a collection of bicliques in G such that $C_i = (L_i, R_i, E_i)$. Then, we say that $\mathcal{C}$ is an *edge biclique partition* of G if

1. $\bigcup_{i=1}^{k} E_i = E$ and
2. $E_i \cap E_j = \emptyset$ for every i, j $(1 \leq i, j \leq k, i \neq j)$.

It is possible that $L_i \cap L_j \neq \emptyset$ and $R_i \cap R_j \neq \emptyset$.

Example 2. Consider the bipartite graph G and the bicliques C_i $(1 \leq i \leq 4)$ in Example 1. Also let C_5 and C_6 be the following bicliques.

$$C_5 = (\{2\}, \{b\}, \{(2, b)\}),$$
$$C_6 = (\{1\}, \{b\}, \{(1, b)\}).$$

Then, both $\{C_1, C_5\}$ and $\{C_3, C_4, C_6\}$ are edge biclique partitions of G.

As the hardness results for the problem of finding an edge biclique partition from a bipartite graph, it is known that the problem of finding the edge biclique partition whose cardinality is minimum is NP-hard (Jiang and Raviunar, 1993) and as hard to approximate as graph coloring (Chalermsook et al., 2014). Furthermore, the approximation algorithm (Chalermsook et al., 2014) and the FPT algorithm (Chandran et al., 2016) of this problem have discussed.

In this paper, by extending the problem BICOCLUST from a biclique partition to an edge biclique partition in Definition 2, we introduce the following problem concerned with an edge biclique partition.

Definition 4. The problem BIEGCOCLUST of *bipartite edge correlation clustering* is defined as follows.

> BIEGCOCLUST
> INSTANCE: A labeled complete bipartite graph $G = (L, R, E)$.
> SOLUTION: Find an edge biclique partition $\mathcal{C}$ of G such that $da_G(\mathcal{C})$ is minimum.

As same as the problem BICOCLUST, we can adopt the formulation of the problem BIEGCOCLUST by using a non-labeled bipartite graph.

By improving the algorithm PIVOTBICLUSTER in Algorithm 1, we design the randomized algorithm PIVOTBICLUSTEREDGE in Algorithm 2 of solving the problem of BIEGCOCLUST.

The algorithm PIVOTBICLUSTER in Algorithm 1 finds clusters of vertices with deleting vertices. On the other hand, the algorithm PIVOTBICLUSTEREDGE in Algorithm 2 finds clusters of edges with deleting edges.

Then, the algorithm PIVOTBICLUSTEREDGE uses the condition that $E \neq \emptyset$ in while loop in line 2, which is the condition that $L \neq \emptyset$ in the algorithm PIVOTBICLUSTER. Also it works nothing when $|R_{1,2}| < |R_1|$ in line 10, and deletes l_2 from L' when $N(l_2)$ is empty after deleting edges in $E_{1,2}$ in line 12. As same as PIVOTBICLUSTER, PIVOTBICLUSTEREDGE always works nothing when $R_{1,2} = \emptyset$.

Example 3. Consider the bipartite graph G in Figure 1.

Then, by selecting l_1 as $\langle 1, 5, 6, 3 \rangle$ in this order, the algorithm constructs a cluster C_i and transforms to a graph G_i in the i-th while loop $(1 \leq i \leq 4)$ in Figure 1. In this case, the algorithm constructs every cluster uniquely, since every probability is either 0 or 1. Hence, algorithm outputs a set $\Gamma_1 = \{C_1, C_2, C_3, C_4\}$ of clusters.

```
   procedure PIVOTBICLUSTEREDGE(G)
       /* G = (L,R,E): bipartite graph */
1      Γ ← ∅; /* Γ: set of clusters */
2      while E ≠ ∅ do
3          select l_1 ∈ L uniform randomly;
4          E_1 ← {(l_1,r) | r ∈ N(l_1)}; C ← E_1;
           E ← E \ E_1; /* C: cluster */
5          L' ← L \ {l_1}; R' ← R \ N(l_1);
6          foreach l_2 ∈ L \ {l_1} do
7              R_1 ← N(l_1) \ N(l_2);
               R_2 ← N(l_2) \ N(l_1);
               R_{1,2} ← N(l_1) ∩ N(l_2);
8              With probability min{ |R_{1,2}| / |R_2| , 1 } do
9              begin
10                 if |R_{1,2}| ≥ |R_1| then
11                     E_{1,2} ← {(l_2,r) | r ∈ R_{1,2}};
                       C ← C ∪ E_{1,2}; E ← E \ E_{1,2};
12                     if N(l_2) = ∅ then
                           L' ← L' \ {l_2};
13
14             end
15         Γ ← Γ ∪ {C}; L ← L'; R ← R';
16     output Γ;
```

Algorithm 2: PIVOTBICLUSTEREDGE.

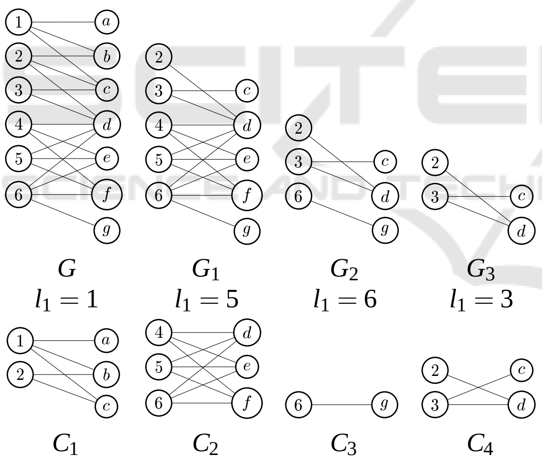

Figure 1: The graph G, the constructed cluster C_i ($i = 1,2,3,4$) and the transformed graph G_j ($j = 1,2,3$).

On the other hand, when the selection of l_1 is changed, the algorithm outputs different sets of clusters. Figure 2 illustrates the sets Γ_1 (same as above), Γ_2, Γ_3, Γ_4 and Γ_5 of clusters when the selection of l_1 is (1) $\langle 1,5,6,3\rangle$, (2) $\langle 2,1,4,6\rangle$, (3) $\langle 6,3,1,2\rangle$, (4) $\langle 6,2,1\rangle$ and (5) $\langle 6,1,3\rangle$ in this order. In all the cases, every probability is also either 0 or 1.

In the remainder of this section, in order to evaluate the probabilistic effect in the randomized algorithms PIVOTBICLUSTER and PIVOTBICLUSTEREDGE, we design the deterministic versions of them.

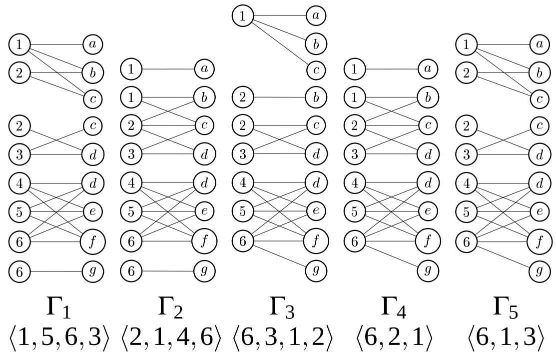

Figure 2: Sets Γ_i ($1 \le i \le 5$) of clusters.

First, we replace the random selection of $l_1 \in L$ in line 3 in Algorithm 1 and line 2 in Algorithm 2 with the following statement.

select $l_1 \in \mathrm{argmax}\{|N(l)| \mid l \in L\}$;

Here, when the candidates of l_1 exist more than two, we select the minimum index of l_1.

Next, we improve the algorithms to execute just when $|R_{1,2}| \ge |R_2|$, that is, the probability of $\min\{\frac{|R_{1,2}|}{|R_2|}, 1\}$ is 1. For PIVOTBICLUSTER, we replace the statements from lines 7 to 11 in Algorithm 1 with the following statements.

if $|R_{1,2}| > 0$ **and** $|R_{1,2}| \ge \max\{|R_1|, |R_2|\}$ **then**
$C \leftarrow C \cup \{l_2\}$;
$L' \leftarrow L' \setminus \{l_2\}$;

We denote this algorithm by DETPIVOTBICLUSTER. Note that the algorithm DETPIVOTBICLUSTER outputs singletons just the last execution corresponding to line 13 in Algorithm 1.

On the other hand, for PIVOTBICLUSTEREDGE, we replace the statements from lines 8 to 13 in Algorithm 2 with the following statements.

if $|R_{1,2}| > 0$ **and** $|R_{1,2}| \ge \max\{|R_1|, |R_2|\}$ **then**
$E_{1,2} \leftarrow \{(l_2, r) \mid r \in R_{1,2}\}$; $C \leftarrow C \cup E_{1,2}$;
$E \leftarrow E \setminus E_{1,2}$;
if $N(l_2) = \emptyset$ **then** $L' \leftarrow L' \setminus \{l_2\}$;

We denote this algorithm by DETPIVOTBICLUSTEREDGE. For example, when applying the algorithm DETPIVOTBICLUSTEREDGE to G in Figure 1, we obtain just Γ_5 in Figure 2, after selecting l_1 as $\langle 6,1,3\rangle$ in this order.

All of the algorithms of PIVOTBICLUSTER, PIVOTBICLUSTEREDGE, DETPIVOTBICLUSTER and DETPIVOTBICLUSTEREDGE run in $O(nm)$ time, where $n = |L|$ and $m = |R|$ for a bipartite graph (L,R,E). Then, the difference between the running time of the algorithms in Section 4 later follows from the number of iterations.

4 EXPERIMENTAL RESULTS

In this section, we give experimental results of evaluating the algorithms. We use two kinds of data. One is artificial data of 4 kinds of biclusters obtained by selecting exclusive or overlapped and row or column. Another is real data of not only MovieLens dataset[1] discussed in (Asteris et al., 2016) but also datasets of Crime (MC)[2], Sexual escorts (SX)[2], arXiv cond-mat (AC)[2], Jester 100 (J1)[2] and YouTube (YG)[2].

4.1 Artificial Data

First, we give experimental results for artificial data. For natural numbers a, b and c ($a < b$), we denote a square enclosing four points (a,b), $(a+c,b)$, $(a,b+c)$ and $(a+c,b+c)$ by $[a,b;c]$. We regard the square $[a,b;c]$ as the complete bipartite graphs such that $L = \{a,\ldots,a+c\}$ and $R = \{b,\ldots,b+c\}$. Then, we prepare the following five sets D_{xy} of squares as data for clustering. Here, $s,t \in \{e,o\}$, e denotes "exclusive" and o denotes "overlapped," according to (Madeira and Oliveira, 2004).

- D_{ee} (exclusive row and column biclusters): $[1,1;100]$, $[101,101;100]$, $[201,201;100]$, $[301,301;100]$ and $[401,401;100]$.
- D_{eo} (exclusive row and overlapped column biclusters): $[1,1;100]$, $[101,91;100]$, $[201,181;100]$, $[301,271;100]$ and $[401,361;100]$.
- D_{oe} (overlapped row and exclusive column biclusters): $[1,1;100]$, $[91,101;100]$, $[181,201;100]$, $[271,301;100]$ and $[361,401;100]$.
- D_{oo} (overlapped row and column biclusters): $[1,1;100]$, $[91,91;100]$, $[181,181;100]$, $[271,271;100]$ and $[361,361;100]$.

Furthermore, we also use data with noises by flipping from 1% to 10% points in whole data.

Figure 3 illustrates the average value of disagreements pointed by y-axis obtained by applying the algorithms to D_{st} $(s,t \in \{e,o\})$ with k% noises pointed by x-axis at 20 times.

Figure 3 shows that the value of disagreements for PIVOTBICLUSTEREDGE is always smaller than that for PIVOTBICLUSTER. Also the value of disagreements for DETPIVOTBICLUSTEREDGE is always smaller than that for DETPIVOTBICLUSTER. Then, the value of disagreements when finding an edge biclique partition is smaller than the value of disagreements when finding a biclique partition.

As the result for comparing probabilistic algorithms with deterministic algorithms, the value of

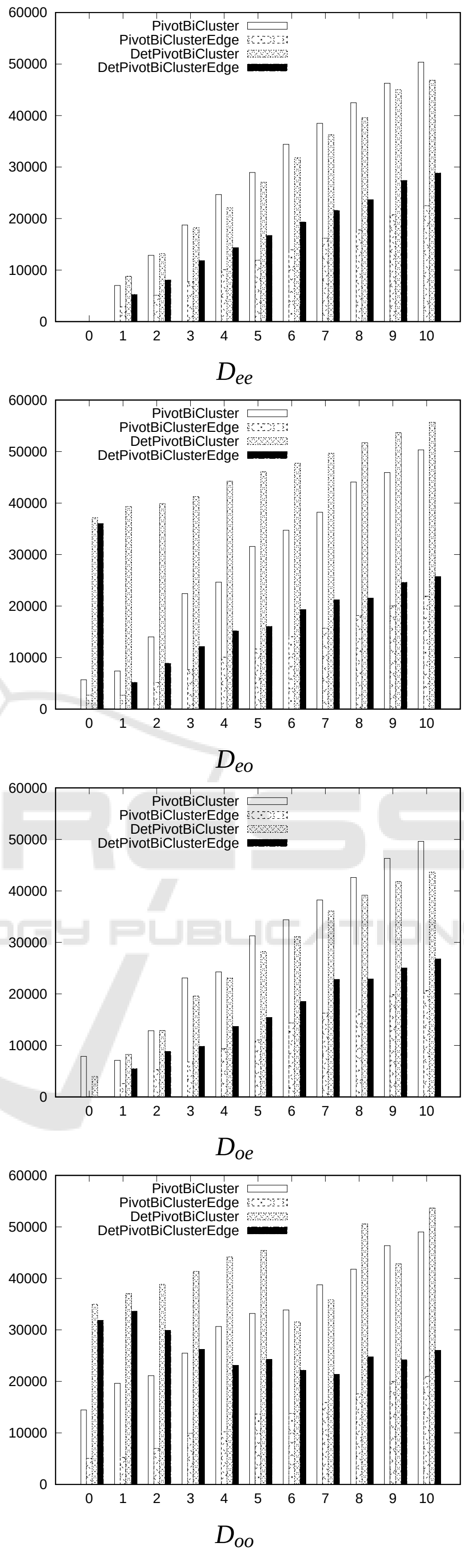

Figure 3: The average value of disagreements obtained by applying the algorithms to D_{st} $(s,t \in \{e,o\})$ with k% noises.

disagreements for PIVOTBICLUSTEREDGE is always smaller than that for DETPIVOTBICLUSTEREDGE. On the other hand, the value of disagreements for PIVOTBICLUSTER is smaller than that for DETPIVOTBICLUSTER for D_{eo} and larger for D_{ee} and D_{oe}. For D_{oo}, the value of disagreements for PIVOTBICLUSTER is larger than that for DETPIVOTBICLUSTER when adding with 6%, 7% and 9% noises and smaller otherwise.

Figure 4 illustrates the average running time (sec.) pointed by y-axis to applying the algorithms to D_{st} for $s,t \in \{e,o\}$ with k% noises at 10 times.

Figure 4 shows that PIVOTBICLUSTER is the fastest algorithm in four algorithms. Also, in almost cases, DETPIVOTBICLUSTER is the second fastest algorithm. On the other hand, both PIVOTBICLUSTEREDGE and DETPIVOTBICLUSTEREDGE are much slower than PIVOTBICLUSTER and DETPIVOTBICLUSTER, except D_{eo} with from 1% to 5% noises.

By incorporating Figure 3 with Figure 4, we can conclude that smaller value of disagreements implies larger running time and vice versa. One of the reasons why the algorithm is slow is that the number of iterations in it is large and then the value of disagreements decreases while iterating.

4.2 Real Data

Next, we give experimental results by using real data such that MovieLens datasets[1] with comparing the result in (Asteris et al., 2016) and datasets of CM, SX, AC, J1 and YG from KONECT[3]. Table 1 summarizes such data as a bipartite graph $G = (L,R,E)$.

Table 1: Summary of MovieLens datasets and datasets of CM, SX, AC,J1 and YG as a bipartite graph $G = (L,R,E)$.

dataset	$\|L\|$	$\|R\|$	$\|E\|$
MovieLens100K	1,000	1,700	10,000
MovieLens1M	6,000	4,000	100,000
MovieLens10M	72,000	10,000	1,000,000
CM	829	551	1,476
SX	10,106	6,624	39,044
AC	16,726	22,015	58,595
J1	73,421	100	4,136,360
YG	94,238	30,087	293,360

In the following tables, we denote the algorithms of PIVOTBICLUSTER implemented by (Asteris et al., 2016), PIVOTBICLUSTER, PIVOTBICLUSTEREDGE, DETPIVOTBICLUSTER and DETPIVOTBICLUSTEREDGE implemented by this paper

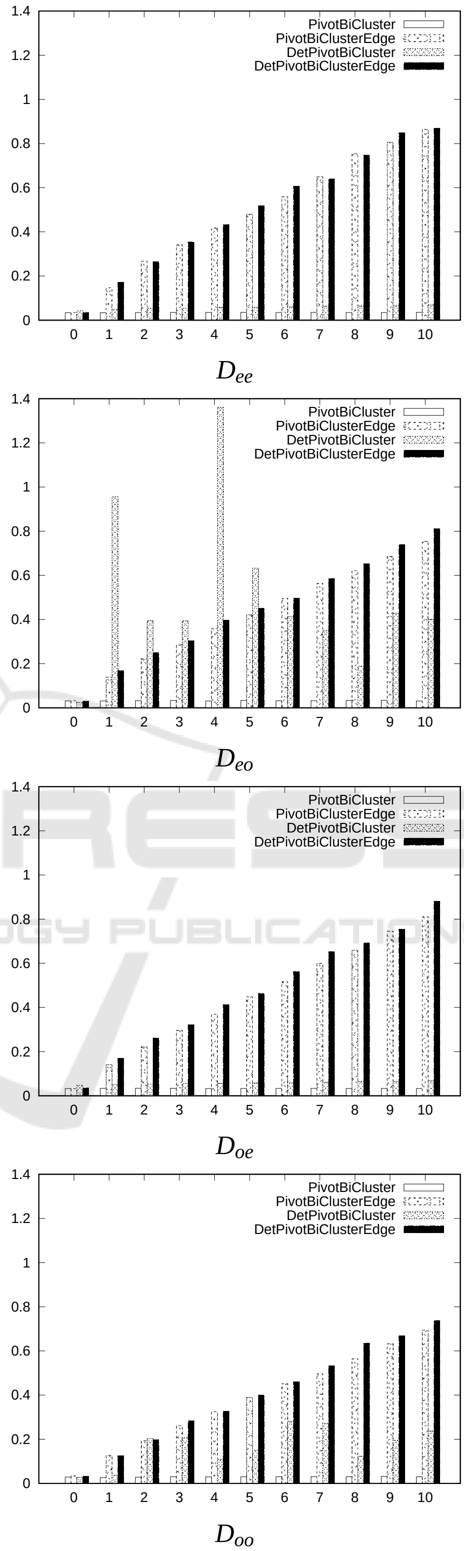

Figure 4: The average running time (sec.) the algorithms for $D_{s,t}$ ($s,t \in \{eo\}$) with k% noises.

by PBC_A, PBC, PBCE, DPBC and DPBCE, respectively.

Table 2 illustrates the average value of *agreements*, in order to compare the results in (Asteris et al., 2016), obtained by applying the algorithms to MovieLens datasets at five times and its average running time (sec.). The first column is the average value of agreements presented in (Asteris et al., 2016), which implies that our implementations are correct by comparing with the second column. Note that the value of disagreements is $|E|$ minus the value of agreements.

Table 2: The average value of agreements obtained by applying the algorithms to MovieLens dataset and its average running time.

algorithms	100K	1M	10M
PBC_A	46,134	429,277	5,008,577
PBC	46,160	429,589	5,011,629
time (sec.)	2.17	90.19	11,154
PBCE	98,497	986,577	9,780,291
time (sec.)	4.36	274.36	55,135
DPBC	45,772	427,138	4,999,434
time (sec.)	2.12	91.64	9,096
DPBCE	99,555	997,882	9,943,548
time (sec.)	2.70	156.60	18,104

Table 2 shows that the value of agreements of PIVOTBICLUSTER (*resp.* DETPIVOTBICLUSTER) is larger than that of PIVOTBICLUSTEREDGE (*resp.* DETPIVOTBICLUSTEREDGE), where DETPIVOTBICLUSTEREDGE has the largest number. Also the value of agreements of each of the randomized algorithms is similar as that of the corresponding deterministic versions.

On the other hand, the algorithm PIVOTBICLUSTEREDGE occupies the largest running time and the algorithm DETPIVOTBICLUSTEREDGE does the next largest running time.

Table 3 illustrates the average value of disagreements obtained by applying the algorithms to datasets of CM, SX, AC, J1 and YG at five times and its average running time (sec.).

Table 3 shows that the algorithm PIVOTBICLUSTEREDGE gives the smallest value of disagreements for all the datasets, and the algorithm PIVOTBICLUSTER gives the smallest running time. Also, whereas the algorithm PIVOTBICLUSTEREDGE is slowest for the MovieLens datasets in Table 2, the algorithm DETPIVOTBICLUSTEREDGE is slowest for the datasets of KONECT in Table 3.

In particular, for the J1 dataset, the value of disagreements is extremely larger than other datasets.

Table 3: The average value of disagreements obtained by applying the algorithms to CM, SX, AC, J1 and YG and its average running time.

algorithms	CM	SX	AC	J1	YG
PBC	669	32,788	31,943	2,136,009	237,400
time (sec.)	0.15	11.17	53.48	2.48	199.96
PBCE	87	4,505	4,775	1,106,915	55,717
time (sec.)	0.19	39.13	68.62	20.25	679.02
DPBC	836	32,039	31,940	1,780,884	2,557,781
time (sec.)	0.63	171.68	167.38	7264.45	6,577.75
DPBCE	217	5,148	6,938	1,360,870	88,858
time (sec.)	0.28	65.31	84.84	67.14	1,536.84

One of the reason is that almost biclusters tend to be stars, that is, bipartite graphs such that either L or R is a singleton, since $|R|$ is much smaller than $|L|$ as represented in Table 1.

As summary of Figures 3 and 4 and Tables 2 and 3, whereas the algorithm PIVOTBICLUSTEREDGE is slower than the algorithm PIVOTBICLUSTER, the former gives smaller value of disagreements or larger value of agreements than the later. In particular, except the MovieLens datasets in Table 2, the algorithm PIVOTBICLUSTEREDGE gives the smallest value of disagreements and each of randomized algorithms are faster than the corresponding deterministic version.

Finally, to analyze the extracted biclusters, Table 4 illustrates the average number (*num*) and the average cardinality (*crd*) of extracted biclusters from the small datasets of CM, SX and AC. Here, "w.s." means that "without singletons."

Table 4: The average number and the average cardinality of extracted biclusters from CM, SX and AC.

	CM		SX		AC	
algorithms	*num*	*crd*	*num*	*crd*	*num*	*crd*
PBC	519	2.67	9,513	1.76	13,803	2.81
(w.s.)	281	4.07	1,907	4.79	6,297	4.96
PBCE	441	4.30	5,769	7.60	11,588	5.44
DPBC	670	2.06	9,177	1.82	12,398	3.12
(w.s.)	162	5.37	2,141	4.53	4,315	7.10
DPBCE	419	4.49	5.756	7.68	10,816	5.85

Table 4 shows that, whereas PIVOTBICLUSTER and DETPIVOTBICLUSTER extract larger number of smaller biclusters, PIVOTBICLUSTEREDGE and DETPIVOTBICLUSTEREDGE extract smaller number of larger biclusters. Also PIVOTBICLUSTER and DETPIVOTBICLUSTER extract many singletons. Without singletons, DETPIVOTBICLUSTER extracts larger clusters for CM and AC but DETPIVOTBICLUSTER does for SX.

5 CONCLUSION

In this paper, we have formulated the problem BIEGCOCLUST of bipartite edge correlation clustering and designed the algorithm PIVOTBICLUSTEREDGE to solve this problem, by improving the algorithm PIVOTBICLUSTER (Alion et al., 2012), with designing the deterministic versions of them. Then, we have given experimental results to evaluate the algorithms by using artificial data and real data such as MovieLens datasets[1] and datasets from KONECT[3].

First of all, concerned with the intractability results for BICOCLUST and BIEGCOCLUST, it is an important work whether or not the problem BIEGCOCLUST is NP-hard and is non-approximable. Then, it is a future work whether or not the algorithm PIVOTBICLUSTEREDGE is an approximation algorithm for the problem BIEGCOCLUST, in particular, it guarantees either approximation ratio as similar as (Amit, 2004) or probabilistic ratio as similar as (Alion et al., 2012).

Concerned with Section 4, it is a future work to analyze not only the value of disagreements (or agreements) but also the number and the cardinality of biclusters for other datasets in Table 4 and the density and the diameter of biclusters. It is also a future work to apply the algorithm PIVOTBICLUSTEREDGE to real data for community detection and evaluate the algorithm.

It is a future work to extend the problem BICOCLUST with the maximum agreement (Asteris et al., 2016) to the problem BIEGCOCLUST with the maximum agreement. Furthermore, since the running time of all the algorithms is quadratic, they are not efficient to large datasets, so it is a future work to design a faster algorithm by introducing some heuristics.

In this paper, we evaluate the results of PIVOTBICLUSEREDGE by using the number of disagreements, as same as PIVOTBICLUSER. On the other hand, the purpose of the problem BICOCLUST is different from that of BIEGCOCLUST. Hence, it is an important future work to introduce a more approproate new criterion to evaluate the results of PIVOTBICLUSTEREDGE, for example, the number of crossing edges (Ahmad and Khokhar, 2007), and then investigate whether or not the problem BIEGCOCLUST with the new criterion is intractable.

REFERENCES

Ahmad, W. and Khokhar, A. (2007). cHawk: An efficient biclustering algorithm based on bipartite graph crossing minimization. In *VLDB Workshop on Data Mining in Bioinformatics*.

Alion, N., Avigdor-Elgrabli, N., Liberty, E., and van Zuylen, A. (2012). Improved approximation algorithms for bipartite correlation clustering. *SIAM J. Comput.*, 41:1110–1121.

Amit, N. (2004). *The bicluster graph editing problem*. Master Thesis, Tel Aviv University.

Asteris, M., Kyrillidis, A., Papailiopoulos, D., and Dimakis, A. G. (2016). Bipartite correlation clustering: Maximizing agreements. In *Proc. AISTATS'16*, pages 121–129.

Chalermsook, P., Heydrich, S., Holm, E., and Karrenbauer, A. (2014). Nearly tight approximability results for minimum biclique cover and partitions. In *Proc. ESA'14 (LNCS 8737)*, pages 235–246.

Chandran, S., Issac, D., and Karrenbauer, A. (2016). On the parameterized complexity of biclique cover and partition. In *Proc. IPEC'16*, pages 11:1–11:13.

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Proc. ISBM'00*, pages 93–103.

Jiang, T. and Raviunar, B. (1993). Minimal NFA problems are hard. *SIAM J. Comput.*, 22:1117–1141.

Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE Trans. Comput. Bio. Bioinfo.*, 1:24–45.

Oghabian, A., S. Kilpinen, S. H., and Czeizler, E. (2014). Biclustering methods: Biological relevance and application in gene expression analysis. *PLOS ONE*, 9:e90801.

Orlin, J. (1977). Containment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae*, 80:406–424.

Pio, G., Ceci, M., D'Elia, D., Loglisci, C., and Malerba, D. (2013). A novel biclustering algorithm for the discovery of meaningful biological correlations between microRNAs and their target genes. *BMC Bioinformatics*, 14:Suppl. 77.

Pio, G., Ceci, M., Malerba, D., and D'Elia, D. (2015). CombiRNet: a web-based system for the analysis of miRNA-gene regulatory networks. *BMC Bioinformatics*, 16:Suppl. 9.