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

Mikio Mizukami¹, Kouich Hirata¹ and Tetsuji Kuboyama²
¹Kyushu Institute of Technology, Kawazu 680-4, Iizuka 820-8502, Japan
²Gakushuin 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 biclusters 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 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...
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 \textit{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 \textsc{PivotBiCluster}, we design the randomized algorithm \textsc{PivotBiClusterEdge} of the bipartite edge correlation clustering, which outputs no singletons. Also, in this paper, we design the deterministic versions of \textsc{PivotBiCluster} and \textsc{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\textsuperscript{1} discussed in (Asteris et al., 2016) but also Crime (MC)\textsuperscript{2}, Sexual escorts, opshal-collaboration, jester1, youtube-groupmemberships.

\section{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 \textit{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 $(\{l\}, \emptyset)$ or $(\emptyset, \{r\})$ is always a biclique.

\textbf{Definition 1.} Let $C = \{C_1, \ldots, C_k\}$ be a collection of bicliques in $G$. We say that $C$ is a \textit{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 $\bigcup_{i=1}^{k} E_i$.

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

$L = \{1, 2\}, R = \{a, b\},$

$E = \{(1, a), (1, b), (2, b)\},$

$C_1 = \{\{1\}, \{a, b\}, \{(1, a), (1, b)\}\},$

$C_2 = \{\{2\}, \emptyset, \emptyset\},$

$C_3 = \{\{1\}, \{a\}, \{(1, a)\}\},$

$C_4 = \{\{2\}, \{b\}, \{(2, b)\}\}.$

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 \textit{labeled complete bipartite graph}. Let $C = \{C_1, \ldots, C_k\}$ be a biclique partition of $G$, where $C_i = (L_i, R_i, E_i)$. Also let $E^+_C = \bigcup_{i=1}^{k} E_i$ and $E^-_C = E \setminus E^+_C$.

Then, the \textit{disagreement} $d_a(C)$ of $C$ in $G$ is defined as follows.

\begin{equation*}
    d_a(C) = |\{e \in E^+_C \mid l(e) = -1\}| + |\{e \in E^-_C \mid l(e) = 1\}|.
\end{equation*}

\textbf{Definition 2 (Amit (Amit, 2004)).} The problem BiCoCUST of bipartite correlation clustering (or bicluster graph editing) is defined as follows.

\textbf{BiCoCUST} \\
\textsc{Instance:} A labeled complete bipartite graph $G = (L, R, E)$.
\textsc{Solution:} Find a biclique partition $C$ of $G$ such that $d_a(C)$ is minimum.

We can formulate the problem BiCoCUST by using a non-labeled non-complete bipartite graph $G = (L, R, E)$. When $G$ is given as an instance the problem BiCoCUST, 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 BiCoCUST is NP-hard. Furthermore, Alion et al. (Alion et al., 2012) have designed both a deterministic and a randomized algorithms of BiCoCUST 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 \textsc{PivotBiCluster} (Alion et al., 2012) which guarantees probabilistic 4-approximation of the problem of BiCoCUST.

As related works to the problem BiCoCUST, Asteris et al. (Asteris et al., 2016) have discussed the
procedure PIVOTBICLUSTER(G) /* G = (L,R,E): bipartite graph */
Γ ← ∅; /* Γ: set of clusters */
while L ̸= ∅ do
    select l₁ ∈ L uniformly random;
    C ← {l₁} ∪ N(l₁); /* C: cluster */
    L′ ← L \ {l₁}; R′ ← R \ N(l₁);
    foreach l₂ ∈ L′ do
        with probability min \( \frac{|R_1|}{|R_2|}, 1 \) do
            begin
                if |R₁| ≥ |R₂| then C ← C ∪ {l₂};
                else Γ ← Γ ∪ {{l₂}};
            end
            L ← Γ ∪ C; L′ ← L′; R ← R′;
        output Γ;

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

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 Ravivuara, 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 C of G such that \( d_{GC}(C) \) 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 ̸= ∅ in while loop in line 2, which is the condition that L ̸= ∅ in the algorithm PIVOTBICLUSTER. Also it works nothing when |R₁| < |R₂| in line 10, and deletes l₂ from L’ when N(l₂) is empty after deleting edges in E₁₂ in line 12. As same as PIVOTBICLUSTER, PIVOTBICLUSTEREDGE always works nothing when R₁₂ = ∅.

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

Then, by selecting l₁ as \{1, 5, 6, 3\} in this order, the algorithm constructs a cluster \( C_1 \) and transforms to a graph \( G_i \) in the i-th while loop (1 ≤ i ≤ 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.

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 biclique edge correlation clustering.

Definition 3. Let \( C = \{C₁, C₂, ..., Cₖ\} \) be a collection of bicliques in G such that \( C_i = (L_i, R_i, E_i) \). Then, we say that \( C \) is an edge biclique partition of G if

1. \( \bigcup_{i=1}^{k} E_i = E \)
2. \( E_i \cap E_j = \emptyset \) for every \( i, j \) (1 ≤ i, j ≤ k, i ≠ j).

It is possible that \( L_i \cap L_j = \emptyset \) and \( R_i \cap R_j = \emptyset \).

Example 2. Consider the bipartite graph G and the bicliques \( C_i \) (1 ≤ i ≤ 4) in Example 1. Also let \( C_3 \) and \( C_6 \) be the following bicliques.

\( C_3 = \{\{2\}, \{b\}, \{(2,b)\}\} \),
\( C_6 = \{\{1\}, \{b\}, \{(1,b)\}\} \).

Then, both \( \{C_1, C_5\} \) and \( \{C_3, C_4, C_6\} \) are edge biclique partitions of G.
procedure PivotBiClusterEdge(G)
    /* G = (L, R, E): bipartite graph */
    Γ ← ∅; /* Γ : set of clusters */
    while E ≠ ∅ do
        select l₁ ∈ L uniformly;
        E₁ ← {(l₁, r) | r ∈ N(l₁)}; C ← E₁;
        E ← E \ E₁; /* C: cluster */
        L' ← L \ {l₁}; R' ← R \ N(l₁);
        foreach l₂ ∈ L \ {l₁} do
            R₁ ← N(l₁) \ N(l₂);
            R₁,₂ ← N(l₁) ∩ N(l₂);
            With probability min \{\frac{|R₁|}{|R₂|}, 1\} do
                if |R₁| ≥ |R₂| then
                    E₁,₂ ← {(l₂, r) | r ∈ R₁,₂};
                    C ← C ∪ E₁,₂; E ← E \ E₁,₂;
                end
                if N(l₂) = ∅ then
                    L' ← L' \ {l₂};
                end
            end
        end
        Γ ← Γ ∪ {C}; L ← L'; R ← R';
    output Γ;

Algorithm 2: PivotBiClusterEdge.

First, we replace the random selection of l₁ ∈ L in line 3 in Algorithm 1 and line 2 in Algorithm 2 with the following statement.

select l₁ ∈ argmax\{|N(l)| \ l ∈ L\};

Here, when the candidates of l₁ exist more than two, we select the minimum index of l₁.

Next, we improve the algorithms to execute just when |R₁| ≥ |R₂|, that is, the probability of min\{\frac{|R₁|}{|R₂|}, 1\} is 1. For PivotBiCluster, we replace the statements from lines 7 to 11 in Algorithm 1 with the following statements.

if |R₁| > 0 and |R₁| ≥ max\{|R₁|, |R₂|\} then
    C ← C ∪ \{l₂\};
    L' ← L' \ \{l₂\};
end

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 |R₁| > 0 and |R₁| ≥ max\{|R₁|, |R₂|\} then
    E₁,₂ ← {(l₂, r) | r ∈ R₁,₂}; C ← C ∪ E₁,₂;
    E ← E \ E₁,₂;
    if N(l₂) = ∅ then L' ← L' \ \{l₂\};
end

We denote this algorithm by DetPivotBiClusterEdge. For example, when applying the algorithm DetPivotBiClusterEdge to G in Figure 1, we obtain just Γ₅ in Figure 2, after selecting l₁ as \{6, 1, 3\} 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 discussed in (Asteris et al., 2016) but also datasets of Crime (MC), Sexual escorts (SX), arXiv cond-mat (AC), Jester 100 (J1) and YouTube (YG).

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 \(D_n\) 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).

- \(D_{eo}\) (exclusive row and column biclusters):
  \[[1, 1; 100], [101, 101; 100], [201, 201; 100], [301, 301; 100]\] and \([401, 401; 100]\).

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

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

- \(D_{oe}\) (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 \(D_n\) \((s, t \in \{e, o\}\) 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
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.

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 with comparing the result in (Asteris et al., 2016) and datasets of CM, SX, AC, J1 and YG from KONECT. 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.
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>
<thead>
<tr>
<th>algorithms</th>
<th>100K</th>
<th>1M</th>
<th>10M</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PBC_A$</td>
<td>46,134</td>
<td>429,277</td>
<td>5,008,577</td>
</tr>
<tr>
<td>$PBC$</td>
<td>46,160</td>
<td>429,589</td>
<td>5,011,629</td>
</tr>
<tr>
<td>$PBCE$</td>
<td>98,497</td>
<td>986,577</td>
<td>9,780,291</td>
</tr>
<tr>
<td>$DPBC$</td>
<td>45,772</td>
<td>427,138</td>
<td>4,999,434</td>
</tr>
<tr>
<td>$DPBCE$</td>
<td>99,555</td>
<td>997,882</td>
<td>9,943,548</td>
</tr>
</tbody>
</table>

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.

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

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.

<table>
<thead>
<tr>
<th>algorithms</th>
<th>CM</th>
<th>SX</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PBC$</td>
<td>519</td>
<td>2.67</td>
<td>9,513</td>
</tr>
<tr>
<td>(w.s.)</td>
<td>281</td>
<td>4.07</td>
<td>1,907</td>
</tr>
<tr>
<td>$PBCE$</td>
<td>441</td>
<td>4.30</td>
<td>5,769</td>
</tr>
<tr>
<td>$DPBC$</td>
<td>670</td>
<td>2.06</td>
<td>9,177</td>
</tr>
<tr>
<td>(w.s.)</td>
<td>162</td>
<td>5.37</td>
<td>2,141</td>
</tr>
<tr>
<td>$DPBCE$</td>
<td>419</td>
<td>4.49</td>
<td>5,756</td>
</tr>
</tbody>
</table>

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 PIVOTBiCLUSTEREDGE by using the number of disagreements, as same as PIVOTBiCLUSTER. 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 appropriate 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


