J. Basic. Appl. Sci. Res., 2(5)5324-5334, 2012

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Mining Dense Structures by Enumerating Weighted and Multi-level Pseudo-Bicliques

Zareen Alamgir[1], Saira Karim[1] and Syed Husnine[2]

[1]Department of Computer Science, National University of Computer and Emerging Sciences, Block B, Faisal Town Lahore, Pakistan
[2] Department of Mathematics , National University of Computer and Emerging Sciences, Block B, Faisal Town Lahore, Pakistan

ABSTRACT

Pseudo-bicliques model various problems encountered in bio-informatics, data mining and networks. They relax the rigid connectivity requirement of bicliques to cater missing and noisy data. In this paper, we consider the weighted density based model of pseudo-biclique. This model defines pseudo-biclique as a bipartite subgraph such that the ratio of the number of its edges to the number of edges in biclique of same size is no less than a given threshold value. The weighted model of pseudo-bicliques better fits the real-world situations and give much more flexibility to researchers. We propose an algorithm based on reverse search to generate all weighted pseudo-bicliques in a given graph. This is computationally non-trivial task as simple straightforward branch-and-bound and back-tracking schemes involve an NP complete problem. Furthermore, we proposed the use of multi-level pseudo-bicliques to discover knowledge at multiple levels and extend our algorithm to enumerate all multi-level pseudo-bicliques. We introduce various enhancements to our algorithm based on the structure of pseudo-bicliques and underlying bipartite graph. We evaluated the performance of our algorithms on random graphs and real-world problems. The results are quite promising and show that average linear time is incurred to generate each pseudo-biclique.

Keywords: combinatorial generation, graph theory, bipartite graph, biclique, data mining, algorithm, stock and financial ratios.

1.0 INTRODUCTION

Exhaustive generation of discrete combinatorial objects is of fundamental interest in computer science. The solution to various problems encountered in computing, networks, bio-informatics, chemistry, data mining, etc. commences by generating all possibilities that can arise. With the advent of the high speed digital computer, it is possible to treat vast amounts of data in a practical amount of time. In areas, such as genome science and data mining the problems are often vaguely defined, and researchers have to find meaningful structure in huge data set. In these areas, enumeration is being widely used to obtain optimal or nearly optimal objects.

The dense structures in a graph are of significant importance as they represent groups of similar objects or deeply related objects. Nowadays, due to increase in computational power it is possible to efficiently generate lists of dense structures. Biclique, a dense structure, is used to model various real-world problems: discovery of web communities, document and words co-clustering, images and features co-clustering and protein interaction discovery [8]. Due to the rigid all-verses-all connectivity requirement of biclique, it is unsuitable for dealing with incorrect or missing data. If an edge in a biclique is removed it is no more a biclique. However, it still is a dense structure and represents a more natural interaction in many real-life situations.

In this paper, we propose the use of weighted pseudo-bicliques instead of biclique to capture natural interactions in real world data. The pseudo-bicliques are more practical as they relax the rigid connectivity requirement of bicliques and thus, cater for the missing data [8]. Here, we consider the density based model of pseudo-biclique. This model defines pseudo-biclique as a bipartite subgraph such that the ratio of the number of its edges to the number of edges in biclique of same size is no less than a given threshold value. The use of weighted version of pseudo-bicliques gives a lot of flexibility to the researchers looking for similar patterns in huge data-set. We extend the definition of pseudo-bicliques to multiple levels. This allows us to find effectively fault-tolerant multi-level patterns.

The generation of density based pseudo-biclique is a non-trivial task because straightforward back-tracking and branch-and-bound schemes involve a NP-complete problem [7]. Secondly, the monotone property does not hold in the family of density based weighted pseudo-bicliques. Therefore, we cannot say that every subset of a pseudo-biclique is also a pseudo-biclique. Because of the anti-monotone property, the various techniques used in literature to enumerate combinatorial objects are not applicable to density based pseudo-bicliques.

In this paper, we suggest the application of our algorithms in stock market and social network. We apply

***Corresponding Author:** Zareen Alamgir, Department of Computer Science, National University of Computer and Emerging Sciences, Email: saira.karim}@nu.edu.pk, Block B, Faisal Town Lahore, Pakistan.

the concept of weighted and multi-level pseudo-bicliques to group similar stocks on the bases of financial ratios and vice versa. The financial ratios are valuable indicators of a company's financial situation and future performance. The ratios are compared with the historical values of the same company or with the ratios of similar companies to extract meaningful information. For this reason, financial analysts usually cluster stocks (or any other security) on the bases of financial ratios [3]. The clusters are then carefully examined to discover the related stocks on the bases of financial ratios. This helps to gain understanding of the company current financial situation and predict its future behavior. Apart from this, we also suggest the application of pseudo-bicliques in finding users with similar interests in social networks. We conduct experiments on a social network of movie ratings by users.

Now, we briefly outline the contributions of our research work. In this paper, we design efficient algorithm for listing all weighted pseudo-bicliques in a given graph G. The framework of our algorithm is based on reverse search [1]. We extend our generation scheme to enumerate weighted multi-level pseudo-bicliques. We evaluated the performance of our algorithm on randomly generated bipartite graphs and real-world problems. We considered two real-world data sets: stock and financial ratios set and social network for movie ratings. The results are very promising and show that average linear time is incurred for generating a pseudo-biclique.

The rest of the document is organized as follows: Section 2 describes the related work. Section 3 introduces the basic concepts used in this paper. In Section 4, we develop a generation algorithm for listing all pseudobicliques. Section 5 presents multi-level pseudo-bicliques and generation algorithm to enumerate them. In section 6, we describe the applications of our algorithm. In Section 7, we gives details of our computational experiments and results. Finally, we conclude the paper in Section 8.

2.0 RELATED WORK

Generation of bicliques and maximal bicliques has attracted a lot of attention in the last decade. Researchers have developed many polynomial delay algorithms to list down bicliques [11]. However, the biclique cannot handle missing data because of its rigid connectivity requirements. Therefore, researchers are now considering pseudo-bicliques to model more natural interactions in real world problems.

There are many ways to define pseudo-biclique, one possible model is a subgraph that is acquired from a complete bipartite graph by removing fixed number of edges. Another model is to define a pseudo biclique in terms of density. In this case, pseudo-biclique is a subgraph that has density greater than a given threshold value. Here, the density is the ratio of the number of edges in pseudo biclique to the number of edges in a complete bipartite graph of the same size. In the first model, certain number of edges is removed from graphs of any size. Thus, larger subgraphs can lose only a small portion of the edges, and many trivial vertex sets will become pseudo-bicliques. However, in the case of density based definition the restraint on the number of edges, changes with the size of the subgraph. Furthermore, small subgraphs are classified as pseudo bicliques only if they are bicliques.

Some schemes have been devised to enumerate pseudo-bicliques using the first model [8], [12] but no significant work is conducted using the density based definition of the pseudo-bicliques. David Gibson [2] proposed an algorithm for finding as many disjoint dense subgraphs in a given graph as possible. As his algorithm generates an incomplete list, thus, it can skip some useful dense graphs. In [6], the scheme to find quasi cliques in a given graph is extended to deal with quasi bicliques, but it can only list balanced quasi bicliques. The proposed schemes to list pseudo-bicliques are mostly limited to bipartite graphs. In [8], the author deals with this issue by converting a given graph to an equivalent bipartite graph. However, in the process he doubles the number of vertices and this increase the computation time considerably. Apart from this their algorithm lists duplicate pseudo-bicliques and requires a post process to remove them, thus increasing the computation time.

The problem of generating all pseudo-bicliques can be compared with the data mining techniques to list pseudo frequent item sets in a transactional database. Some mining algorithms [9, 12] deal with the missing item set by considering it a fault if an item of the item set is missing in a transaction. These approaches vary just one partition of the biclique that is they consider only frequent item set and do not deal with the frequent transactions [5]. However, in many applications related to chemistry and biology such as discovering protein-protein interactions, it is desirable to vary both the partition of the pseudo-bicliques. Our proposed algorithm deals with this issue and generates a complete list of pseudo-bicliques in average linear time per object. This is significantly better than the previously known related algorithms [5] as it gives an improvement by a factor of V, here V is the total number of vertices in the given graph G(V, E).

We apply our pseudo-biclique generation algorithm to co-cluster stock and financial ratios data set, which has a considerable amount of missing data. A technique called subspace clustering is also used to find clusters within different subspaces of high-dimensional data sets [15]. However, existing subspace and co-clustering algorithms donot handle missing data. Some researchers have proposed the use of self-organizing maps (SOMs) [3] to group stocks based on their financial ratios. SOM has its own limitations, in SOM it is difficult for user to

define clear clusters of entities because the boundaries of the clusters are hard to tell apart. In [4], authors resolve this issue and give some well-defined stock clusters, but still they cannot tell the financial ratios in which a cluster of stocks is highly related.

3.0 BASIC DEFINITIONS

A graph G = (V, W) comprises of a set of vertices V and weight matrix W to keep edge weights. The weight of the edge between vertex i and j is denoted by ***wi,j*** . We assume that graph G is simple, undirected and weighted (without loss of generality). Also, all the edge weights are less or equal to 1 that is $0 < w_{i,j} < 1$. A graph G is a bipartite if its vertex set V can be partitioned into two disjoint nonempty sets V1 and V2, such that each edge in G has one endpoint V1 and the other in V2. A bipartite graph G is often denoted as G = (V1∪V2,W).

A graph $H = (V_h, W_h)$ is a subgraph of a graph G if $V_h \subseteq V$ and $W_h \subseteq W$. We say that H is a biclique subgraph, if every vertex in the set $V_h \cap V1$ is connected to every vertex in $V_h \cap V2$. Furthermore, we say H is maximal biclique subgraph if no other biclique contains H.

Now, we introduce the definitions of weighted degree, density and pseudo-biclique subgraphs.

Definition 3.1. Weighted degree of a vertex v in a weighted graph G(V, W) is defined as $\deg(v) = \sum_{u \in \{V\}} w_{u,v}$

We define density of a bipartite subgraph: as the ratio of number of its edges to the number of edges in biclique of the same size.

Definition 3.2. For a weighted bipartite graph G(V1,V2,W), the density β(G) is given by

$$\beta(G) = \frac{\sum_{i \in V1} \deg(i)}{|V1| \times |V2|} = \frac{\sum_{i \in V2} \deg(j)}{|V1| \times |V2|} \qquad \textbf{(Eq. 1)}$$

For unweighted graphs, the weight matrix would be a binary matrix where 1 represents existence of the edge and 0 represents its absence. Whereas, the density relation remains the same.

Definition 3.3. A pseudo-biclique B_{U_1,U_2} is a bipartite subgraph of graph G, if $\beta(B) \geq \theta$, where $0 < \theta \leq 1$. We denote the degree of vertex v in a subgraph B by degB(v), the maximum degree by $\Delta(B_{U_1,U_2})$ and minimum degree by $\delta(B_{U_1,U_2})$.

4.0 ENUMERATION OF WEIGHTED PSEUDO-BICLIQUES

In this paper, we address the problem of finding dense bipartite subgraphs using an enumerative approach. We develop an algorithm to list weighted pseudo-bicliques in a given graph G using the reverse search technique. The reverse search, originally developed by Avis and Fukuda [1], [7], is a sophisticated depth-first search type scheme for generation. It is widely used in the field of combinatorial generation because it is a simple and efficient technique. In reverse search, we construct a tree-shaped traversal route on the family of the combinatorial object under consideration. In order to form the tree, we define a parent for each element and ensure that definition of the parent is unique and acyclic, that is, each element is not a proper ancestor of itself. The parent-child relation forms an enumeration tree; a spanning tree on the set of elements to be generated. The reverse search algorithm traverses the tree in a depth- first. One benefit of this technique is that it does not memorize the visited elements in memory space.

We need to establish an adjacency relation on the set of weighted pseudo-bicliques to enumerate all such structures. We observe that the removal of a vertex with a minimum weighted degree from a bipartite subgraph does not decrease the density of the resultant subgraph. Ties are broken lexicographically; if there are more than one minimum weighted degree vertices then we consider the one with minimum index. We use this observation to define a parent-child relationship on the set of weighted pseudo-bicliques.

The following lemma establishes parent-child relationship on the set of weighted pseudo-bicliques.

Fact 4.1. Let v be the minimum weighted degree vertex in G =(V1 ∪ V2,W), without loss of generality assume that v ∈ V1, then

$$\sum_{i \in V1} \deg G(i) \geq |V1| \deg G(v) \qquad \textbf{(2)}$$

Lemma 4.1. Let B_{U_1,U_2} be a weighted pseudo biclique, and vertex v ∈ (U1 ∪ U2). If $\deg B_{U_1,U_2}(v) = \delta(B_{U_1,U_2})$

then $\beta(B_{U_1,U_2} \setminus v) \geq \beta(B_{U_1,U_2})$.

Proof. We want to proof that the density of $B_{U_1,U_2} \setminus v$ is no less than the density of B_{U_1,U_2}. Without loss of generality we assume that $v \in U2$. Now, we show that

$$\beta(B_{U_1,U_2} \setminus v) - \beta(B_{U_1,U_2}) \geq 0$$

By substituting the value of density in the left side of above equation we get

$$\frac{\Sigma i \in U1 \ \deg_{BU1,U2}(i) - \deg_{BU1,U2}(v)}{|U1| \times |U2 - 1|} - \frac{\Sigma i \in U1 \ \deg_{BU1,U2}(i)}{|U1| \times |U2|}$$

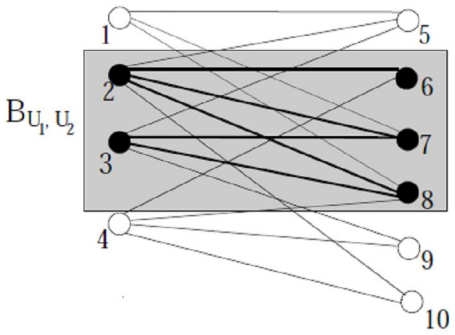
$$= \frac{\Sigma i \in U1 \ \deg_{BU1,U2}(i) - |U2| \times \deg_{BU1,U2}(v)}{|U1| \times |U2| \times |U2 - 1|}$$

$$\geq \frac{|U2| \times \deg_{BU1,U2}(v) - |U2| \times \deg_{BU1,U2}(v)}{|U1| \times |U2| \times |U2 - 1|} = 0$$

Using the above lemma we can establish that each pseudo-biclique has density no more than its parent, thus a parent $Prt(B_{U_1,U_2})$ is a pseudo-biclique if B_{U_1,U_2} is a pseudo-biclique. In other words, we can say that for any pseudo-biclique B_{U_1,U_2}, $B_{U_1,U_2} \setminus v$ will also be a pseudo-biclique if v is a minimum degree vertex in B_{U_1,U_2}. Now, we define a unique parent-child relationship among pseudo-bicliques.

Definition 4.1. Let B_{U_1,U_2} be a pseudo-biclique, then $B^* = (B_{U_1,U_2} \cup v)$ is a child of B_{U_1,U_2} if for every $u \in \{U1 \cup U2)\}$ one of the two conditions holds

1 $\deg B^*(v) < \deg B^*(u)$
2 $\deg B^*(v) = \deg B^*(u)$ and label of v is less than label of u



	5	6	7	8	9	10
1	0.6	0	0.7	0.2	0	0
2	0.3	1	0.9	0.8	0	0.9
3	0.6	0	0.4	0.5	0.8	0
4	0	0.7	0	0.8	0.3	0.7

Fig. 1: An illustrative example for Pseudo-biclique enumeration and weight matrix W

In Fig. 1, we show an example of pseudo-biclique B_{U_1,U_2} comprising of vertices {2, 3, 6, 7, 8}. According to our defined parent-child relationship only the addition of vertex 1, 5 or 9 can yield children of B_{U_1,U_2}. Vertex 4 cannot be added because $\deg_{B_{U_1,U_2}}(4) > \delta(B_{U_1,U_2})$. Although the degree of vertex 10 is equal to $\delta(B_{U_1,U_2})$., but it is lexicographically larger than vertex 3 (minimum degree vertex of $\deg B_{U_1,U_2}$).

4.1 Algorithm

In this section, we outline our algorithm design. First, we present the main idea and then suggest some improvements to reduce the computation time of the algorithm. Using computational experiments we prove that our suggested efficiencies significantly improve the time bounds, and it takes linear time to generate each pseudo-biclique.

The generation algorithms that are based on reverse search start with the trivial structure (an object of minimal size) and find all its children recursively. In our case, each vertex can be thought of a pseudobiclique. However, we want to avoid trivial pseudo-bicliques that have all vertices in the same partition. To achieve this we start our algorithm with an edge of the given graph G(V1 ∪ V2,W), and then recursively search its children. This pruning step does not affect the result because all the descendants of such subgraphs are the subgraphs with zero edges. Furthermore, this also prunes the bicliques that have two or more zero degree vertices. We develop a

preprocessing routine that calls GenPseudoBiclique for each edge to enumerate all the pseudo-biclique in that branch of the enumeration tree.

Mostly, we are not even interested in small sized or star shape graphs. To avoid such structures, we propose to start the algorithm with a subset of vertices that forms a pseudo-biclique. We can set a constraint on the minimum number of vertices allowed in each partition. However, there is one drawback in this preprocessing step, that is we may skip some of the pseudo-bicliques for lower density thresholds. So we have to choose a value for subset size that gives us efficiency and non-trivial pseudo-bicliques without losing any important structure. In our computational experiments on real-world data sets, we run the algorithms on subsets that are pseudo-bicliques having two vertices in each partitions.

As stated earlier every pseudo-biclique has a unique parent, this adjacency relation spans all possible pseudobicliques and forms the enumeration tree. Depth-first traversal of the enumeration tree will visit each node exactly once, which ensures that the result set is complete without any duplication. In our algorithm, we traverse the search space in a way that allows straightforward pruning of non-dense pseudo-bicliques. According to the defined adjacency relationship, a non-dense pseudo-biclique will has non-dense descendants. During the depth-first traversal, we prune the path whenever the density check fails at a node. Thus, we avoid the generation of non-dense descendants.

Algorithm 1 GenPseudoBiclique(B_{U_1,U_2})

Require: Graph $G(V_1 \cup V_2, W)$, density threshold θ

```
1: for each v ∈ {V1 ∪ V2} \ {U1 ∪ U2} do
2:   if ρ(B_{U1,U2} ∪ v) ≥ θ then
3:     if B_{U1,U2} ∪ v is a child of B_{U1,U2} then
4:       Output B_{U1,U2}
5:       GenPseudoBiclique(B_{U1,U2} ∪ v)
6:     end if
7:   end if
8: end for
```

Algorithm 1: Algorithm for generating Pseudo-bicliques

Two basic operations are performed before inclusion of each vertex: computation of minimum weighted degree vertex and density. The graphs are represented using adjacency lists. In simple straight forward implementation of the algorithm, time to compute the density of a pseudo biclique takes O(V) time. Similarly finding minimum degree vertex of a pseudo biclique takes O(V) time. Both these operations are performed at most V times in an iteration of the algorithm, thus time to compute a pseudo biclique is $O(V^2)$. The space requirement of the algorithm is quadratic since no additional space is used other than the weight matrix of the graph.

The above algorithm can be used with slight modification to list all maximal pseudo-bicliques. A pseudo-biclique B_{U_1,U_2} is maximal if and only if there does not exist a vertex v such that v/∈ U1 ∪U2 and $\beta(B_{U_1,U_2} \cup \{v\}) \geq$ e. The existence of v ensures that B_{U_1,U_2} is not maximal, and hence it is not listed by the algorithm in the output. In Algorithm 1 B_{U_1,U_2} is not maximal if the condition at line number 2 holds for any iteration. Hence, by addition of a simple constraint we can list only the maximal pseudo-bicliques of the given graph G(V1 ∪ V2,W).

4.2 Improvements for Efficient Computation

The time requirements of the algorithm can be improved by maintaining information about the minimum degree vertex and degrees of all vertices of G in B_{U_1,U_2} under consideration. We keep an array to record the degree of each vertex in B_{U_1,U_2}. For any v, if we know deg B_{U_1,U_2}(v) and the degree sum of all the vertices in B_{U_1,U_2} then the density of $B_{U_1,U_2} \cup \{v\}$ can be computed in a constant amount of time. Time to verify the parent-child relationship can also be improved. In most cases, only a comparison between deg B_{U_1,U_2} (v) and $\delta(B_{U_1,U_2})$ is sufficient to do the job. We can subdivide the task of determining child of B_{U_1,U_2} in one of the following three cases.

1. If deg B_{U_1,U_2}(v) is strictly less than $\delta(B_{U_1,U_2})$, then $B_{U_1,U_2} \cup \{v\}$ is a child of B_{U_1,U_2}
2. If deg B_{U_1,U_2} (v) is greater than $\delta(B_{U_1,U_2})$ + Wv,$\delta(B_{U_1,U_2})$, then it is not a child of B_{U_1,U_2}
3. Otherwise one of the two possibilities can occur
   (a) If v is connected to minimum degree vertex of B_{U_1,U_2}, then verify the parent-child relationship
   (b) If v is not connected to minimum degree vertex of B_{U_1,U_2}, then a comparison between label of v and minimum degree vertex of B_{U_1,U_2}complete the task

In all the above cases except 3(a), verification of child can be done in constant time. Only the case 3(a) takes O(V) time.

Now we show that the cost incurred on constant checkings can be distributed to pseudo-bicliques as overhead. When a pseudo-biclique is generated, it takes O(V). This is because when a vertex is added to B_{U_1,U_2}, the degrees of all of its adjacent vertices in the array are updated. This operation takes $O(\Delta(G))$ time. For each B_{U_1,U_2}, the number of constant checkings that does not yield any child are at most O(V). The overhead incurred for constant checking can be included in the generation cost of B_{U_1,U_2}. Hence, we can say that overhead of constant checkings does not affect the asymptotic time bounds of the algorithm. In section 7, we estimate non constant checkings using computational experiments. We found that the total number of non constant checkings is O(f(G)), where f(G) is the total number of pseudo bicliques in G.

Another improvement is to avoid trivial pseudo bicliques that have all vertices in the same partition. We start our enumeration algorithm with an edge. This allows us to prune all the subgraphs that have all vertices in the same partition. Pruning these bicliques from the enumeration tree do not affect the result. This is because all the descendants of such nodes are the subgraphs with zero edges. Furthermore, we also prune the bicliques that have two zero degree vertices.

4.3 Generating Weighted Pseudo-bicliques in General Graphs

Now we extend our pseudo-biclique generation algorithm described in previous section for general graphs. In general graph G, we wish to enumerate all non-induced pseudo bicliques.

Previously, some researchers [8] have dealt with generation in general graph by using a graph transformation scheme. In this scheme, the general graph G =(V, E) is transformed to an equivalent bipartite graph $G'(VUV',E')$, where $V' = \{v'|vi \in V\}$ and $E' = \{(vi,vj')|(vi,vj) \in E\}$. Now we can say that B_{V_1,V_2} is a pseudo-biclique in G if and only if B'_{V_1,V_2} is a pseudo-biclique in G', here V1 ⊆ V , V2 ⊆ V , and $V2' \subseteq V'$. We have not used this scheme because it double the size of the problem and, thus doubles the computation time and space. Enumeration is an exhaustive procedure, and it is highly desirable that generation of each object is extremely fast.

The nature of our algorithm allows us to extend it easily to the general graphs without increasing the size of the input graph G. Note that the lemma 4, proved in section 4 also holds for general graph. Therefore, we use the same adjacency relation on the set of pseudo-bicliques in general graph G that is described in the section 4. As every pseudo-biclique has a unique parent, the graph induced by this parent-child relationship forms a tree.

The Algorithm 1 GenPseudoBiclique can be adapted to work for general graphs. Note that, when input graph G is bipartite, then the partition of each vertex in generated pseudo-bicliques is specified by G. But this is not the case when G is a general graph. In this scenario, we have to check for two possible pseudo-bicliques against each vertex v. We introduce an array in Algorithm 1 to hold the partition id of each vertex in G. We first set partition id of v to U1 partition of B_{U_1,U_2} and perform steps 2-7 of Algorithm 4. We repeat this process after setting partition id of v to U2 partition. This will generate the complete list of all non-induced pseudo-bicliques in general graphs.

5.0 MULTILEVEL PSEUDO-BICLIQUES

The existing algorithms for pseudo-biclique enumeration deal with mining knowledge at single concept levels. However, mostly it is desired to discover knowledge at multiple concept levels. For example, besides clustering stocks on the bases of financial ratios, it will be more interesting to show analyst that stocks clusters are 80 percent similar in liquidity ratios and 50 percent related in debt ratios, etc. Secondly, in many real-world applications, multiple dimensions may be associated with one clustering entity. As in the case of social network, for example, movie rating network, we have different genres of movies. Incorporating dimension information into the mining process can produce patterns with more detailed knowledge. As described before, the existing subspace clustering algorithm cannot deal with this situation as they cannot handle missing data. We propose multi-level pseudo-biclique enumeration algorithm to handle multi-level clustering with missing data.

We model the data with multiple concept levels using a bipartite graph G(V1 ∪ V2,W), where V2 is further partitioned into non-overlapping vertex sets to represent different concept levels. For instance, consider the authors and books network; we model it as a bipartite graph, where the set of authors are represented as vertices in V1 and books as vertices in V2. An edge symbolizes the connection between an author and a book. To capture the concept at multiple levels, we partition V2 to represent the categories of the books. We are considering only the literary work, so we categorize the books as novels, poetry, drama and short stories. A multi-level pseudo-biclique in this graph corresponds to a group of authors that have coauthored at-least given number of books in each category.

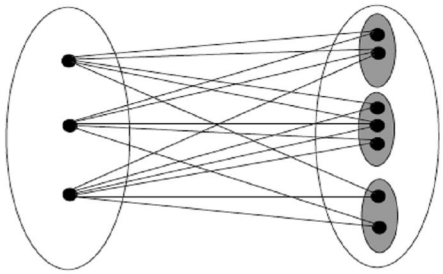


Fig. 2: A Multilevel Pseudo-biclique

Now, we give our definition of a multilevel bipartite graph and a multilevel pseudo-biclique. A multilevel bipartite graph G(V1 $\cup\{V2_1 \cup V2_2 \cup \cdots \cup V2_k\}$,W) is a bipartite graph, where partition V2 is divided into k non-overlapping sets and $V2_i$is the i^{th} set. A multilevel pseudo-biclique B(U1 $\cup\{U2_1 \cup U2_2 \cup \cdots \cup U2_k\}$) is a subgraph of G(V1 $\cup\{V2_1 \cup V2_2 \cup \cdots \cup V2_k\}$,W) such that U1 $\subseteq$ V1, $U2_i \subseteq V2_i$ and $\beta(B_{U_1,U_{2i}}) \geq \theta i$ for all $1 \leq i \leq k$. Here, θi is the density threshold for pseudo-biclique $B_{U_1,U_{2i}}$. The value of θi can be different for all, $1 \leq i \leq k$. Figure 2 shows an example of multilevel pseudo-biclique. Note that every multi-level pseudo-biclique with a same θ threshold for each level is a θ weighted pseudo-biclique.

5.1 Enumeration Algorithm for Multilevel Pseudo-Bicliques

In this section, we enhance our basic algorithm for pseudo-bicliques to enumerate all multilevel pseudo-bicliques. The following lemma establishes adjacency relationship on the set of weighted multilevel pseudo-bicliques.

Lemma 5.1. Let B(U1 $\cup\{U2_1 \cup U2_2 \cup \cdots \cup U2_k\}$) be a weighted multilevel pseudo-biclique, and v be the minimum degree vertex in B then β (B(U1 $\cup\{U2_1 \cup U2_2 \cup \cdots \cup U2_k\}) \setminus v) \geq \beta($ B(U1 $\cup\{U2_1 \cup U2_2 \cup \cdots \cup U2_k\}))$

Proof. We have to consider two cases; v $\in$ U1 or v $\in U2_i$. Note that we can represent the multilevel pseudo biclique B(U1 $\cup\{U2_1 \cup U2_2 \cup \cdots \cup U2_k\}$) as a collection of k pseudo-bicliques B(U1 $\cup U2_i$), $1 \leq i \leq k$. Let v$\in U2_i$, then removal of v effects only the density of pseudo-biclique B(U1 $\cup U2_i$). Therefore, by Lemma 4, $\beta(B(U1 \cup \{U2_i \setminus v\})) \geq \beta(B(U1 \cup U2_i)) \geq \theta i$. On the other hand, when v $\in$ U1, then it is the minimum degree vertex in all k pseudo-bicliques. Thus, repeated application of Lemma 4 ensures that removal of v does not decrease the density of any of the k pseudo-bicliques. Hence, $\beta(B(\{U1 \setminus v\} \cup U2_i)) \geq \beta(B(U1 \cup U2_i))$ for all $1 \leq i \leq k$.

We use the reverse search to enumerate all multilevel pseudo-bicliques based on the parent-child relation defined in Lemma 5.1. The multilevel pseudo-biclique can have different density thresholds for individual pseudo-bicliques. So we make sure that all density constraints are fulfilled after addition of a new vertex in current multilevel pseudo-biclique. Given a multilevel bipartite graph G, with k partitions of set V2, and k density thresholds the Algorithm 2 recursively lists all maximal multilevel pseudo-bicliques in G.

In order to generate all multilevel pseudo-bicliques, we repeatedly invoke the Algorithm 2 for each edge in the given graph. The Algorithm 2 produces a complete list as our parent-child relationship defines a unique parent for each multilevel pseudo-biclique, and we traverse each node in the enumeration tree exactly once. The computation time to output a multilevel pseudo-biclique increase by a factor of k as compare to the Algorithm 4. The additional time is required for computation of density and verifying density thresholds for all k pseudo-bicliques.

Algorithm 2 GenMultiPseudoBiclique($B(U_1 \cup \{U_{2_1} \cup U_{2_2} \cup \cdots \cup U_{2_k}\})$)

Require: Graph $G(V_1 \cup \{V_{2_1} \cup V_{2_2} \cup \cdots \cup V_{2_k}\}, W)$, density thresholds $\theta_1, \theta_2, \cdots, \theta_k$

```
1: isMaximal := true
2: for each v ∉ U1 ∪ U2_1 ∪ U2_2 ∪ ··· ∪ U2_k do
3:    isDense := true
4:    if v ∈ V1 then
5:       for each i := 1 to k do
6:          if ρ(B({U1 ∪ v} ∪ U2_i)) < θ_i then
7:             isDense:=false
8:          end if
9:       end for
10:   else if v ∈ V2 then
11:      i :=partition[v]
12:      if ρ(B(U1 ∪ {U2_i ∪ v})) < θ_i then
13:         isDense:=false
14:      end if
15:   end if
16:   if isDense then
17:      isMaximal:=false
18:      if B(U1 ∪ {U2_1 ∪ U2_2 ∪ ··· ∪ U2_k}) ∪ v is a child of B(U1 ∪ {U2_1 ∪ U2_2 ∪ ··· ∪ U2_k}) then
19:         GenMultiPseudoBiclique(B(U1 ∪ {U2_1 ∪ U2_2 ∪ ··· ∪ U2_k}) ∪ v)
20:      end if
21:   end if
22: end for
23: if isMaximal then
24:   output B(U1 ∪ {U2_1 ∪ U2_2 ∪ ··· ∪ U2_k}) ∪ v
25: end if
```

Algorithm 2: Algorithm for generating Multi-level Pseudo-bicliques

6.0 APPLICATIONS: STOCK AND FINANCIAL RATIOS CO-CLUSTERING AND SOCIAL NETWORKS

In this section, we describe the application of our pseudo-biclique generation algorithms and, in the next section we present the results of our computational experiments on real-world data set.

We apply the concept of weighted and multi-level pseudo-bicliques to group similar stocks on the bases of financial ratios and vice versa. The financial ratios are valuable indicators of a company's financial situation and future performance. They are used to examine trends and compare the financials of the companies. In some cases, financial ratio analysis can be used to foresee future bankruptcy. A study of financial ratios is very crucial for the financial analyst conducting a fundamental analysis of a business. Fundamental analysis is the cornerstone of investing as it provides insight on a company's prospects and potentials. It involves delving into the financial statements, gathering real data, calculating financial ratios to evaluate a security's value.

One of the drawbacks of fundamental analysis is that it involves too many parameters: economic indicators and extensive macroeconomic data. This can greatly complicate the situation for the analysts and investors. Apart from this, a reference point is needed for ratio analysis. This indicates that to dig out meaningful information, one should compare the ratios with the historical values of same corporation or with the ratios of other comparable corporations. For this reason, financial analysts usually cluster stocks (or any other security) on the bases of financial ratios [3]. Then the clusters are carefully examined to discover the related stocks on the bases of financial ratios. This helps to gain understanding of the company current financial situation and predict its future behavior.

We propose the use of weighted pseudo-bicliques to co-cluster stock and financial ratios. The weighted model of pseudo-bicliques provides greater flexibility to the analyst examining the financial ratios. The ratio analysis is used for various purposes like value investment, analyzing company current and future performances and to predict bankruptcy. In the analysis, the analyst considers only those financial ratios that are relevant to the topic of the study under-consideration. Moreover, depending on the study some ratios plays more important role than others. For example, for value investing the investment valuation ratios are very important but role of other ratios like cash flow indicator ratios cannot be completely ignored. In this scenario, it would be highly beneficial for the analyst to assign weights to ratios according to their relevance to the purpose and then cluster them. Furthermore, to provide another level of abstraction we propose the use of multi-level pseudo-bicliques. In this case, we exploit the fact that financial ratios have multiple classification levels like liquidity ratios, efficiency ratios, debt ratios etc. Thus, they can be considered as multi-level entities to extract more information.

The pseudo-bicliques generation is quite helpful in finding peoples with similar interests in social networks. Our generation algorithms, give a list of pseudo-bicliques of desired density that represent some interconnected group of users. We have conducted experiments on a network of movie ratings by users. The output pseudo-bicliques can be quite useful for social networking sites to recommend movies to their users.

7.0 COMPUTATIONAL EXPERIMENTS

We have conducted computational experiments to evaluate the performance of our algorithm. We have used randomly generated bipartite graphs and real-world data-sets in our experiments. The results are quite promising and show that average linear cost is incurred for generating each pseudo-biclique. We carry out our experiments on Windows 7 environment, using Intel(R) Core(TM)i7 CPU (1.6GHz) with 6GB RAM. We implement our algorithm in C++ using Boost Graph Library [10]. The results of the experiments are plotted in log scale.

7.1 Random Graphs

In this experiment, we estimate the ratio of the total number of pseudo-bicliques in a graph to the number of non-constant checkings. The edges in randomly generated bipartite graphs are uniformly distributed according to the given edge density. The number of pseudo-bicliques depends on the given density threshold or on the graph size. For this purpose, we have evaluated the performance of our algorithm on three parameters: density threshold, number of vertices, and edge density. We carry out three different experiments to estimate the desired ratios. In each experiment, we fix two parameters and compute the ratio for various values of the third parameter. In the first experiment, we estimate the ratio for different edge densities. Second experiment examines the effect of various density thresholds on the ratio. In the third experiment, we vary the number of vertices in the graph. The results of these experiments are shown in Figure 3. In these experiments, we invoke the Algorithm 4 for each edge in the randomly generated graph.

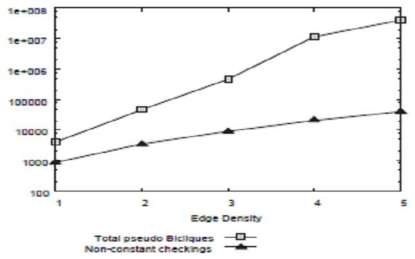
We have made two observations from the above experiments. First, the number of non-constant checkings is less than the number of pseudo bicliques generated in all three experiments. This observation leads us to infer that cost of non-constant checking can be distributed to pseudo-bicliques generated and an amount of work done per pseudo biclique is O(V). Secondly, the growth rate of non-constant checking is far less than the growth rate of pseudo-bicliques when graph size is increased or density threshold is decreased. From this, we can deduce that the average cost of computing a pseudo biclique decreases as the search space of algorithm increases.

7.2 Experiments on Real-World Datasets

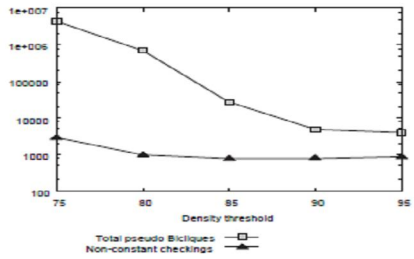
We conduct experiments on two real-world problems: co-clustering stocks and financial ratios and movie rating social network.

7.2.1 Stock Market: Stocks and Financial Ratios

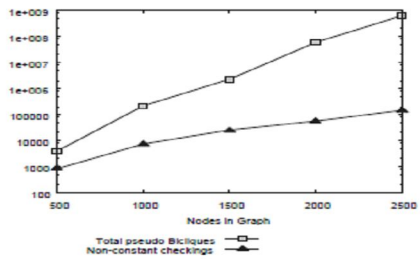
We group stocks and financial ratios using the two approaches developed in this paper, namely enumeration of weighted pseudo-bicliques and multi-level pseudo-bicliques. The existing algorithms [8], [9], [12] cannot be used as they are incapable of finding our defined maximal multilevel pseudo-bicliques. We obtained various stocks and financial ratios data sets for the year 2007 − 2008 from Portfolio123 [17]. The data sets are for medium and large sized technology firms. We have performed experiments on firms of different sizes separately. This is mainly because; researchers have discovered that significant differences exist in financial ratios between large public and small private firms [16], [14]. Thus, it is meaningless to compare financial ratios of the different sized firms.



(a) For different edge densities (%)



(b) For different density thresholds(%)

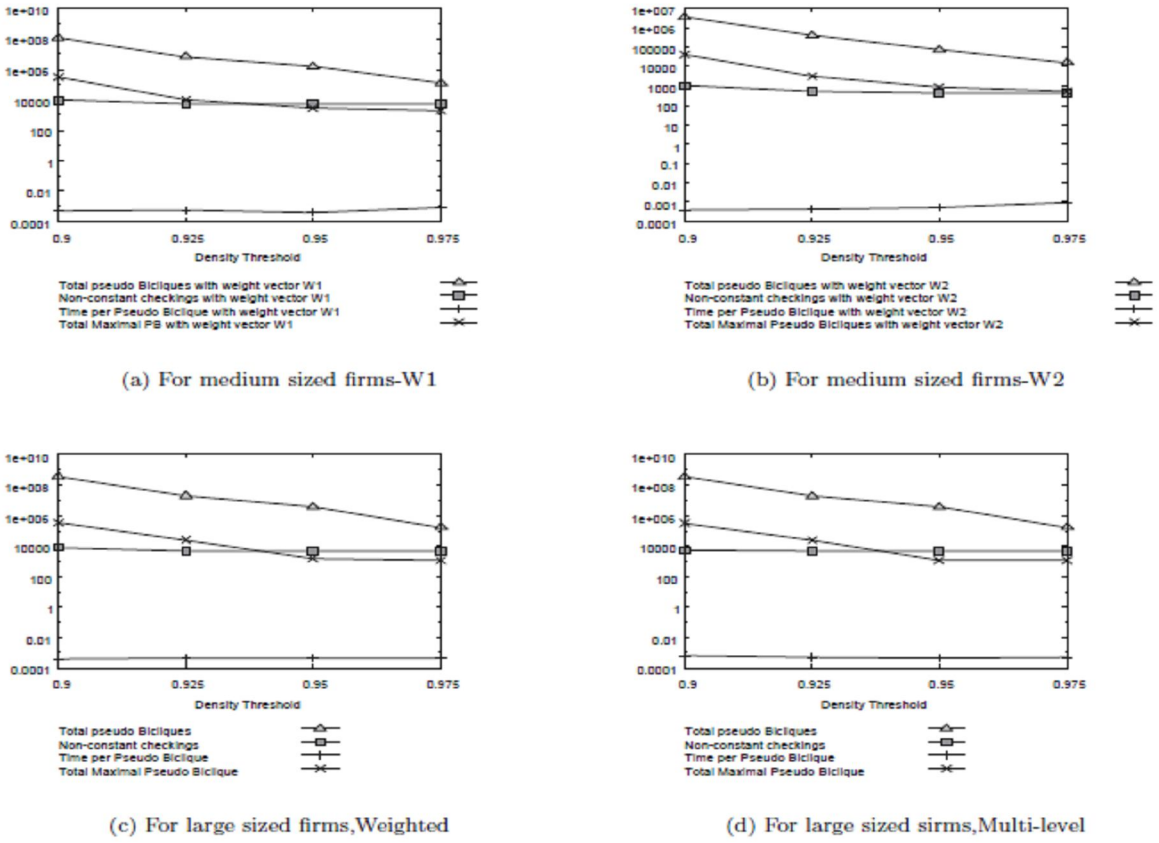


(c) For different nodes

Fig. 3: Results obtained from experiments on random graphs

The choice of the ratios plays an important role and greatly depends on the purpose of the analysis which can be value investment, trend analysis or bankruptcy prediction. In our algorithm, analyst is given the flexibility to assign weights to the ratios according to their importance and relevance. Thus, he can play with the weights to get the desired grouping of stocks and financial ratios. In our experiments, we have considered the following basic categories of ratios: valuation, liquidity, financial strength, profitability and growth. In the valuation category, we have included dividend yield(DY), price earnings ratio(PE), price/earnings to next year growth rate, price to book ratio, price to cash flow per share ratio and price to sales ratio. In order to assess liquidity, the current ratio(CR) was the primary ratio examined. For assessing the financial strength, we have included total debt to total equity and payout ratio(PR) annually. Examination of the return on investment(ROI) and return on equity(ROE) ratios were used to provide an assessment of profitability. Growth was measured by EPS growth rate, gross margin(GM), net income(NI) and sales growth rate(SGR). We have considered five year growth rates.

The financial ratios consist of continuous values. We deploy hierarchical clustering algorithm to divide the range of each financial ratio into discrete clusters. There are various hierarchical clustering methods, we employ the pairwise centroid-linkage clustering. We used C-Clustering library implementation



(a) For medium sized firms-W1

(b) For medium sized firms-W2

(c) For large sized firms,Weighted

(d) For large sized sirms,Multi-level

Fig. 4: Results obtained from experiments on stock and financial ratio data sets

We represent the data sets as bipartite graphs. As mentioned earlier, we have considered two data sets. The bipartite graph for the large-sized firms consists of 401 vertices(101 stocks and 300 financial ratio intervals) and for the medium-sized firm consists of 603 vertices(228 stocks and 375 financial ratio intervals). In this experiment, we have used 15 financial ratios mentioned above. We have used two different weight vectors, namely W1 and W2, to examine the effect of weights on the number of pseudo-bicliques and clustering. In W1, we assign 0.75 weight to PR and ROI, and other ratios are set to one. While in W2, we set price to book ratio and price to cash flow to 0.8 and CR, SGR and NI to 0.5 and rest are same as W1. For multi-level pseudo-bicliques, we group the ratios into three groups: valuation ratios, growth ratios and a combined group for liquidity, profitability and financial strength ratios. In the experiment for Algorithm 2, we have used the same density threshold for all levels and assign weight one to all the ratios.

In this experiment, we run the algorithms on subsets that are pseudo-bicliques having two vertices in each partition. Furthermore, we observe that pseudo-bicliques that have multiple edges between stocks and discretized cluster of one financial ratio provide no useful information. So we prune these pseudo-bicliques and thus, reduce the search space. It is clear, from Figure 4 that the number of the pseudo-bicliques is very large, so

it would be more interesting to output only the maximal pseudo-bicliques for analyzing the stocks' behavior. As described above, with slight modification our algorithm can list the maximal pseudo-bicliques.

The results of the above experiments with different data sets are shown in the Fig. 4. We scrutinize the output set to find out the effect of weight vectors W 1 and W 2. The weight vectors greatly help in pruning the unnecessary pseudo-bicliques. With weight vector W 2, we get much less pseudo-bicliques as compare to W 1, this is evident from the graphs in Figure 4(a) and Figure 4(b). The W 2 vector assign less weight to some ratios and this helps to get a restricted set of pseudo-bicliques.

We examine the pseudo-bicliques generated by both the approaches. As mentioned above, every multi-level pseudo-biclique with a same e threshold for each level is a e weighted pseudo-biclique. However, if we are considering only the maximal pseudo-bicliques set then this is not true. A maximal multi-level pseudobiclique may not belong to the maximal weighted pseudo-biclique set. We observe that output set of maximal multi-level pseudo-cliques provide the financial analyst with specific information regarding financial ratios as it has maintained density for each type of ratios. Moreover, in case of multi-level, we observe that the edges in a pseudo-biclique are evenly distributed; especially in-case the value of e is large. However, this is not true for weighted pseudo-bicliques.

In this experiment, we compare the number of non-constant-checkings with total pseudo-bicliques generated. It is interesting to note that non-constant checkings are far less than the total pseudo-bicliques as in the case of random graphs. Thus, we can distribute the overhead of non-constant-checkings to pseudo-bicliques.

7.2.2 Social-Network: Movie Ratings by Viewers

We have also conducted an experiment on a social network for the movie ratings by viewers. The data set for this experiment was obtained from MovieLens Data Sets [13]. It was gathered by the "GroupLens Research Project" conducted by University of Minnesota. The data set that we have considered consists of 10000 ratings for 1682 movies by 943 users. The users have rated the movies on the scale of 1 to 5. We construct a bipartite graph for this data set with 943 users in the first partition and 8410 rankings of 1682 movies in the other. The graph has 9353 edges. The results of this experiment are shown in the Fig. 5.

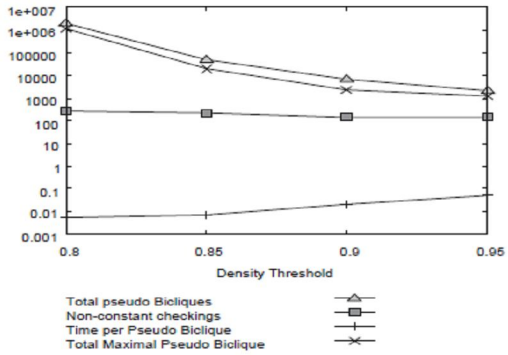


Fig. 5: Results obtained from experiment on movie rating social network

The generated maximal pseudo-bicliques provide groups of users who are interested in similar movies. This can be quite useful for social networking sites to recommend movies to their users. Furthermore, the results confirmed that the non-constant checkings are far less than the number of pseudo-bicliques generated. Therefore, each pseudo-biclique takes average linear time to generate.

8.0 CONCLUDING REMARKS

The pseudo-biclique enumeration is an interesting problem to be studied as it models various real life problems in chemistry, data mining and biology. This paper is set out to investigate the ways for the efficient generation of weighted pseudo bicliques in a given graph. We used density based model to define pseudo biclique and devise a scheme for their exhaustive generation based on reverse search. Our algorithm is optimal in the sense it operates in average linear time and linear space. We extended our algorithm to list multi-level pseudobicliques that helps to discover knowledge at multiple levels of abstraction. We carry out computational experiments and use theoretical bounds to show that our algorithm takes linear time on average to generate each pseudo biclique. Furthermore, we have conducted experiments on real-world data sets namely movie rating social network and stock and financial ratios data sets to evaluate the performance of our algorithms. Usually, it is not desirable to have small sized or asymmetric dense substructures in the output set. This problem can be resolved by adding a constraint on partition size and connectivity of individual vertices. As future work, we want to explore ways that can eliminate small sized and asymmetric pseudo bicliques. Furthermore, we want to

bound the non-constant checkings performed in our algorithm using theoretical bounding techniques.

REFERENCES

[1] D. Avis and K. Fukuda, "Reverse search for enumeration", Discrete Applied Mathematics, Vol. 65, No. 1, 65:21 – 46, March 1996, pp. 21-46.

[2] D. Gibson, R. Kumar and A. Tomkins, "Discovering large dense sub-graphs in massive graphs", in international conference on very large databases VLDB, Norway, September 2005, pp. 721-732.

[3] T. Eklund, B. Back, H. Vanharanta, and A. Visa, "Assessing the feasibility of self-organizing maps for data mining financial information". In Proceedings of the ECIS, Gdansk,Poland, 2002, pp. 528-537.

[4] S. Wu and T. Chow, "Self organizing map based clustering using a local clustering validity index". Neural Processcing Letter, Vol. 17, 2003, pp. 253–271.

[5] T. Uno and H. Arimura, "Ambiguous frequent item-set mining and polynomial delay enumeration", Advances in Knowledge Discovery and Data Mining, Vol. 5012, No.1, May 2008, pp.357-368.

[6] J.Abello, M.G.C.Resende, and S.Sudarsky, "Massive quasi-clique detection", in American Symposium on Theoretical Informatics, Mexico, April 2002.

[7] T.Uno. "An efficient algorithm for solving pseudo clique enumeration problem". Algorithmica, Vol.56, January 2010, pp. 3-16.

[8] J. Li, K. Sim, G. Liu, and L. Wong, "Maximal quasi-bicliques with balanced noise tolerance: concepts and co-clustering applications", in SIAM Conference on Data Mining, Atlanta, USA, April 2008.

[9] J.Liu, S.Paulsen, W.Wang, and A.Nobel, "Mining approximate frequent itemsets from noisy data". In Proceedings of the Fifth IEEE International Conference on Data Mining. Houston, Texas, USA, 27 November 2005.

[10] "Boost Graph Library", open source graph library, web-site "http://www.boost.org/".

[11] V.M.F.Dias, C.M.H.Figueiredo, and J.L.Szwarcfiter, "On the generation of bicliques of a graph". Discrete Applied Mathematics, Vol. 155, September 2007, pp. 1826-1832.

[12] N. Mishra, D. Ron, and R. Swaminathan, " A new conceptual clustering framework". Machine Learning, Vol.56, No.1, 2004, pp.115-151.

[13] "University of Minnesota MovieLens data sets collected by the GroupLens Research Project". web-site "http://www.grouplens.org".

[14] J. Osteryoung, R. L. Constand, and D. Nast, "Financial ratios in large public and small private firms ". Journal of Small Business Management, Vol. 30, No. 3, 1992, pp. 35-46.

[15] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data". SIGKDD Explorer Newsletter, Vol.6, 2004, pp. 90–105.

[16] M.D. Phillips, J. X. Volker, and S. J. Anderson. "A behavioral comparison of financial ratios for different size privately-held retail and service businesses". Journal of Behavioral Studies in Business, Vol.1, No.1, July 2009, pp. 5-12.

[17] "Portfolio123". Web-site, "http://www.portfolio123.com".