Mining Maximal Quasi-Bicliques: Novel Algorithm and Applications in the Stock Market and Protein Networks

Kelvin Sim, Jinyan Li, Vivekanand Gopalkrishnan, and Guimei Liu

***Abstract*— Several real world applications require mining of bicliques, as they represent correlated pairs of data clusters. However, the mining quality is adversely affected by missing and noisy data. Moreover, some applications only require *strong* interactions between data members of the pairs, but bicliques are pairs that display *complete* interactions. We address these two limitations by proposing *maximal quasi-bicliques*. Maximal quasi-bicliques tolerate erroneous and missing data, and also relax the interactions between the data members of their pairs. Besides, maximal quasi-bicliques do not suffer from skewed distribution of missing edges that prior quasi-bicliques have. We develop an algorithm $MQBminer$, which mines the complete set of maximal quasi-bicliques from either bipartite or non-bipartite graphs. We demonstrate the versatility and effectiveness of maximal quasi-bicliques to discover highly correlated pairs of data in two diverse real world datasets. Firstly, we propose to solve a novel financial stocks analysis problem by using maximal quasi-bicliques to co-cluster stocks and financial ratios. Results show that the stocks in our co-clusters usually have significant correlations in their price performance. Secondly, we use maximal quasi-bicliques on a mining protein network problem and we show that pairs of protein groups mined by maximal quasi-bicliques are more significant than those mined by maximal bicliques.**

I. Introduction

Biclique subgraphs have been mined in diverse applications such as finding large interacting pairs of protein groups [1], discovering web communities which contain a group of webpages and a group of users [2], words and documents co-clustering [3], etc. A biclique subgraph consists of two disjoint vertex sets, where all vertices from one set are connected to every vertex from the other. To reduce the redundancies in the biclique subgraphs of a graph, Li *et al.* [4] and Alexe *et al.* [5] proposed to mine biclique subgraphs that are *maximal*. A biclique subgraph of a graph is maximal if and only if it is not a proper subset of any other biclique subgraph of the graph.

However, maximal biclique subgraphs exhibit two weaknesses. Firstly, real world data are prone to contain erroneous or missing values. These missing or erroneous values have an adverse effect on the quality of maximal biclique subgraphs mined. Secondly, the all-to-all (complete) relation between the two vertex sets of maximal biclique subgraphs may be too strict, as some applications may require *most-to-most* relation instead.

Kelvin Sim is with Institute for Infocomm Research, Singapore. Email: shsim@i2r.a-star.edu.sg

Jinyan Li and Vivekanand Gopalkrishnan are with the School of Computer Engineering, Nanyang Technological University, Singapore. Email: {jyli, asvivek}@ntu.edu.sg

Guimei Liu is with the School of Computing, National University of Singapore, Singapore. Email: liugm@comp.nus.edu.sg

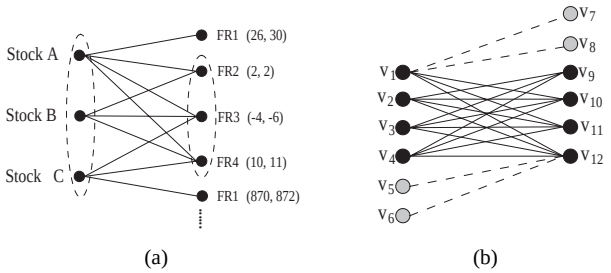


Fig. 1. (a) Vertices labeled with {Stock A, B, C} and {FR2(2, 2), FR3(-4, -6), FR4(10, 11)} form a maximal quasi-biclique subgraph. (b) A skewed quasi-biclique graph where missing edges are not *balanced.*

In this paper, we propose to overcome these two weaknesses by introducing *maximal quasi-biclique subgraphs*. A maximal quasi-biclique subgraph consists of two disjoint set of vertices, X and Y , such that every vertex in X is allowed to disconnect with up to ϵ vertices in Y and vice versa. ϵ is the error tolerant threshold defined by the user. For example in Figure 1(a), vertices labeled with {Stock A, B, C} and {FR2(2, 2), FR3(-4, -6), FR4(10, 11)} are two disjoint vertex sets forming a maximal quasi-biclique subgraph at $\epsilon = 1$. In this subgraph, every vertex is disconnected with up to one vertex from the other vertex set.

This simple, yet elegant definition of maximal quasi-biclique subgraphs can be used to effectively overcome the two weaknesses of maximal biclique subgraphs. Firstly, the attempt to reduce the negative impact of missing or erroneous data is achieved by using ϵ, as each vertex in a maximal quasi-biclique subgraph can tolerate up to ϵ number of errors. Secondly, varying ϵ allows the user to control the strictness of the most-to-most relation of the maximal quasi-biclique subgraphs.

Biclique subgraphs tolerating missing or erroneous data have been studied recently [6]–[9]. However, their definitions do not have a good constraint on the vertices to have a balanced error tolerance, thus they have a skewed distribution of the missing edges. For example, Figure 1(b) is a quasi-biclique subgraph qualified in [6], [7], [9], but the vertices $v_5, \cdots, v_8$ each has a very low connectivity compared to the other vertices. By our definition of maximal quasi-biclique subgraphs, this skewness can be avoided, as the error tolerance is required to be evenly distributed in the subgraph. A detailed

Stock	FR1	FR2	FR3	FR4	FR5	FR6
A	26	**2**	**-4**	**10**	58	-58
B	-2	**2**	**-6**	**11**	**20**	**16**
C	872	**N.A.**	**-5**	**10**	**22**	**14**
D	-17	30	1	**999**	**21**	**15**

Fig. 2. A financial ratios dataset. FR1 to FR6 are financial ratios, e.g. FR1 can be Return on Equity. There are two overlapping co-clusters, {Stock A, B, C} {FR2, FR3, FR4} and {Stock B, C, D} {FR4, FR5, FR6}

comparison between the competing approaches is presented in Section 3.

We develop an algorithm $MQBminer$ to enumerate the complete set of maximal quasi-biclique subgraphs, which is shown to be more efficient and scalable than our previous algorithm $CompleteQB$ [10]. Our algorithm can take either a bipartite graph or non-bipartite graph as input.

To demonstrate the strength and versatility of maximal quasi-biclique subgraphs, we apply them in two radically different real world applications. The first application is proposed by us to solve a long standing financial problem.

Application 1: Co-clustering stocks and financial ratios for fundamental analysis. Careful examination on the *financial ratios* of the companies is an integral part of fundamental analysis [11], [12]. Financial ratios reflect the "health" status of the stock issuing company, hence if a company possesses a healthy status of financial ratios, it is often believed that its fundamentals are strong and it has a high potential to be profitable. Therefore, the price of the company's stock would rise in the long run [11], [12].

Fundamental analysts usually group companies (and consequently, stocks) that have similar financial health status by clustering them based on their financial ratios [13]–[15]. Once clusters are obtained, it is useful to understand which financial ratios the cluster of stocks have close similarities in, so that analysts can investigate the reasons behind it.

Figure 2 shows a financial ratios dataset, with the financial ratios labeled from FR1 to FR6. In this dataset, stocks A, B, C have high similarity in FR2, FR3, FR4, while stocks B, C, D have high similarity in FR4, FR5, FR6. We can consider them as co-clusters of stocks and financial ratios, {Stock A, B, C} {FR2, FR3, FR4} and {Stock B, C, D} {FR4, FR5, FR6}. Subspace clustering algorithms [16] can be used to co-cluster the stocks and financial ratios but the co-clusters found do not overlap, hence co-cluster {Stock A, B, C} {FR2, FR3, FR4} in Figure 2 may become {Stock A, B, C} {FR2, FR3} due to its overlapping with {Stock B, C, D} {FR4, FR5, FR6}, resulting in information loss. Co-clustering algorithms [17], [18] can also be used to co-cluster the stocks and financial ratios. However, neither subspace clustering algorithms nor co-clustering algorithms tolerate missing or erroneous data. Therefore, they are unable to discover these two co-clusters, assuming that FR2 of Stock C is missing and FR4 of Stock D is erroneous.

We propose to use maximal quasi-biclique subgraphs to co-cluster stocks and financial ratios for fundamental analysis. In our method, stocks and their financial ratio values are represented by a bipartite graph. A bipartite graph consists of two disjoint sets of vertices, and edges exist only between pairs of vertices spanning the two disjoint sets. Here we use one set of the vertices to represent the stocks, and the other set to represent the financial ratio values. An example of this representation is shown in Figure 1(a). Since the financial ratio values are continuous, we propose to use hierarchical clustering algorithm with a new scoring function iir (intra-inter ratio) for the discretization of financial ratios into intervals, which are then represented by vertices. An edge exists between a stock vertex s and a financial ratio vertex r_range, if the financial ratio value for the stock s falls in the interval r_range of this financial ratio. We call such a bipartite graph a *StoR* graph.

Maximal quasi-biclique subgraphs are then used to mine co-clusters of stocks and financial ratios from the StoR graph. Thus a maximal quasi-biclique subgraph of an StoR graph corresponds to a co-cluster of stocks and financial ratios. It can be seen that the stocks are clustered based on their similarities in financial ratios and concurrently, these financial ratios are implicitly clustered according to their occurrences in the stocks.

Application 2: Mining protein networks. Li *et al.* [1] transform the protein-protein interactions (ppi) dataset into a non-bipartite graph, where the proteins are represented by vertices and an edge connects two proteins if they have interaction. Maximal biclique subgraphs are then mined from the ppi dataset, and interacting pairs of protein groups represented by maximal biclique subgraphs are shown to be biologically significant.

However, Li *et al.* [1] observe two important characteristics of current ppi datasets that impede the usage of maximal biclique subgraphs. (1) Not all pairs of protein groups exhibit all-to-all interactions. Using maximal bicliques to mine pairs of protein groups will filter off significant pairs of protein groups that exhibit most-to-most interactions. In fact, pairs of protein groups generally exhibit most-to-most interactions and those exhibiting all-to-all interactions are rarities [19]. (2) ppi datasets are incomplete, are constantly updating, and are known to be noisy and of low quality [20]. Thus, the quality of pairs of protein groups mined by maximal biclique subgraphs suffers.

Thus, we propose to mine maximal quasi-biclique subgraphs from ppi dataset and we show that pairs of protein groups mined from maximal quasi-biclique subgraphs are more significant than those mined from maximal biclique subgraphs.

The rest of the paper is organized as follows. Section 2 gives a formal definition of our maximal quasi-biclique subgraphs. Section 3 discusses the related work. Section 4 presents the algorithm $MQBminer$ and the discretization method. Section 5 reports the experiment results and Section 6 concludes the paper.

II. Problem Definition

An undirected graph G consists of a set of vertices denoted by $V(G)$ and a set of edges denoted by $E(G) = \{\{u, v\} | u \neq v \wedge u, v \in V(G)\}$. Vertices $u, v \in V(G)$ are *adjacent* to each

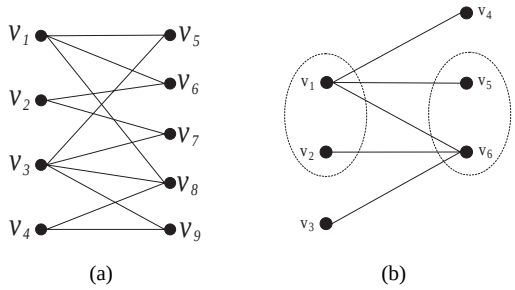


Fig. 3. (a) A bipartite graph G, which contains a maximal quasi-biclique subgraph g, with $V(g) = \{\{v_1, v_2, v_3\}, \{v_5, v_6, v_7\}\}$, at $ms = 3, \epsilon = 1$. (b) A bipartite graph G containing a maximal quasi-biclique subgraph G that does not contain any maximal biclique subgraphs.

other if there is an edge $\{u, v\}$ connecting them. Throughout the rest of the paper, we assume that all graphs are undirected.

The *neighborhood* of v in a graph G is denoted as $\Gamma(v) = \{u | \{v, u\} \in E(G) \wedge u \in V(G)\}$. Let $V \subset V(G)$ and v be a vertex in $V(G) \setminus V$. We denote the set of vertices in V that is adjacent to v as $\Gamma_V(v) = \{u | \{v, u\} \in E(G) \wedge u \in V\}$.

A graph g is a subgraph of a graph G if $V(g) \subseteq V(G)$ and $E(g) \subseteq E(G)$. Graph g is a proper subgraph of G if g is a subgraph of G, and $g \neq G$. A graph G is a bipartite if its vertex set consists of two disjoint subsets of vertices V_x and V_y, and its edge set $E(G)$ consists of only those edges $\{v, u\}$, where $v \in V_x$ and $u \in V_y$. A bipartite graph is complete if $E(G) = \{\{v, u\} | \forall v \in V_x \wedge \forall u \in V_y\}$. For brevity, a complete bipartite graph (or subgraph) is also called a biclique (or biclique subgraph). A complete bipartite subgraph of a graph G is *maximal* if it is not a proper subgraph of any other complete bipartite subgraphs of G. Next, we introduce our definition of maximal quasi-biclique subgraphs.

Definition 1 (Quasi-biclique): A bipartite graph G is a quasi-biclique if $V(G)$ consists of two disjoint sets of vertices V_x and V_y such that $\forall v \in V_x$, $|V_y| - |\Gamma_{V_y}(v)| \leq \epsilon$, and $\forall v \in V_y$, $|V_x| - |\Gamma_{V_x}(v)| \leq \epsilon$, where the error tolerant threshold ϵ is an integer.

Definition 2 (Maximal quasi-biclique): A quasi-biclique subgraph g of an undirected graph G is maximal if and only if there does not exist a quasi-biclique subgraph g' of G such that g is a proper subgraph of g'.

Small maximal quasi-biclique subgraphs may not be practically useful, and enumerating all of them may be computationally expensive since there are potentially a large number of them. Thus, it is desirable to enumerate only maximal quasi-biclique subgraphs whose sizes are larger than a minimum size threshold ms, with the requirement that $ms > \epsilon$. That is, a maximal quasi-biclique g with $V(g) = \{V_x, V_y\}$ is of our interest if $|V_x| \geq ms, |V_y| \geq ms$.

Figure 3(a) shows a bipartite graph G with $V(G) = \{V_x, V_y\}$, $V_x = \{v_1, \ldots, v_4\}$ and $V_y = \{v_5, \ldots, v_9\}$. At $\epsilon = 1$ and $ms = 3$, there is a maximal quasi-biclique subgraph g in G, with $V(g) = \{X, Y\}$, $X = \{v_1, v_2, v_3\}$ and $Y = \{v_5, v_6, v_7\}$. We can see that $\forall v \in Y$, v satisfies the constraint $|X| - |\Gamma_X(v)| \leq \epsilon$, as $|\Gamma_X(v_5)| = 2$, $|\Gamma_X(v_6)| = 2$, $|\Gamma_X(v_7)| = 2$. Similarly $\forall v \in X$, v satisfies the constraint $|Y| - |\Gamma_Y(v)| \leq \epsilon$, as $|\Gamma_Y(v_1)| = 2$, $|\Gamma_Y(v_2)| = 2$, $|\Gamma_Y(v_3)| = 2$.

As the error tolerant threshold ϵ is an integer, it nicely sets an upper bound on the number of missing edges each vertex in a maximal quasi-biclique can tolerate with respect to the size of the maximal quasi-biclique. For example, if $ms = 3, \epsilon = 1$, then each vertex in a maximal quasi-biclique can tolerate up to 33.33% of missing edges that connect it to its counterpart vertex set.

Note that one subset of V_x may form quasi-bicliques with more than one subsets of V_y. For example, in Figure 3(a), $X = \{v_1, v_2, v_3\}$ can form a maximal quasi-biclique with $Y_1 = \{v_5, v_6, v_7\}$ and $Y_2 = \{v_6, v_7, v_8\}$ respectively, but X cannot form a maximal quasi-biclique with the union of Y_1 and Y_2 because v_2 is disconnected to both v_5 and v_8.

III. Comparison to Literature Work

A. Graph

On the error tolerance of quasi-bicliques, we use two notions, *symmetrical* and *balanced*, to characterize them. The error tolerance is symmetrical in a biclique if vertices in the both sides of the quasi-biclique can tolerate missing edges. It is balanced, if every vertex in the quasi-biclique can tolerate up to the same threshold of missing edges. The rationale of defining a quasi-biclique whose error tolerance is symmetrical and balanced is to ensure that each vertex is closely related to all vertices in the counterpart vertex set. Without this constraint, the error distribution will be skewed as roughly explained in the Introduction section.

The definition of quasi-bicliques by Abello *et al.* [6] is density based—A subgraph H is dense if all edges in H divided by the total number of vertices in H exceeds a threshold. Therefore, the error tolerance is not balanced though symmetrical. Mishra *et al.* [8] defined ϵ-bicliques in a way such that its error tolerance is neither symmetrical nor balanced. Specifically, a bipartite subgraph G with $V(G) = \{V_l, V_r\}$ suffices to be a ϵ-biclique if every vertex in V_r is adjacent to $(1 - \epsilon)$ of the vertices in V_l. But every vertex in V_l is not required to be adjacent to at least $(1 - \epsilon)$ of the vertices in V_r, thus the error tolerance of ϵ-biclique is not balanced. The error tolerance of ϵ-biclique is not symmetrical as there is no error tolerant requirement on vertices in V_l. Using Figure 1(b) as an example, at $\epsilon = 0.6$, G is a ϵ-biclique subgraph where $V(G) = \{\{v_1, \ldots, v_4\}, \{v_7, \ldots, v_{12}\}\}$. Thus, the concept of ϵ-bicliques is prone to skewed error distributions.

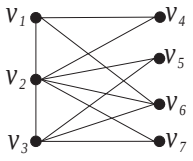
Yan *et al.* [9] introduced α-quasi-bicliques, which are maximal and their error tolerance is symmetrical, but not balanced. An α-quasi-biclique has $V(G) = \{V_l \cup V_{e1}, V_r \cup V_{e2}\}$ where $\{V_l, V_r\}$ forms a maximal biclique and $\{V_{e1}, V_{e2}\}$ its maximal $\alpha-$extension. Every vertex in V_{e1} is adjacent to at least $\alpha\%$ of the vertices in V_r and every vertex in V_{e2} is adjacent to at least $\alpha\%$ of the vertices in V_l. We can see that the tolerance is relative to V_l or V_r, but not relative to the vertex sets of the α-quasi-biclique, therefore its error tolerance is not balanced. For example at $\alpha = 0.25$, Figure 1(b) shows an α-quasi-biclique with $V(G) = \{V_l \cup V_{e1}, V_r \cup V_{e2}\}$, $V_l = \{v_1, \ldots, v_4\}$, $V_r = \{v_{10}, v_{11}, v_{12}\}$, $V_{e1} = \{v_5, v_6\}$, $V_{e2} = \{v_7, v_8, v_9\}$.

To enumerate the complete set of α-quasi-bicliques, all maximal biclique subgraph are first enumerated by using any

Definition	Type	Symmetrical	Balanced	Algorithm
Ours	Maximal	Yes	Yes	Complete
γ-biclique [6]	Density	Yes	No	Greedy
Bu *et al.* [7]	Non-maximal	Yes	Yes	Heuristic
ϵ-biclique [8]	Non-maximal	No	No	Greedy
α-quasi-biclique [9]	Maximal	Yes	No	Complete

TABLE I

COMPARISON OF DIFFERENT TYPES OF QUASI-BICLIQUES AND THEIR ALGORITHMIC APPROACH



(a)

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$\Gamma(v_1)$	0	1	0	1	0	1	0
$\Gamma(v_2)$	1	0	1	1	1	1	1
$\Gamma(v_3)$	0	1	0	0	1	1	1
$\Gamma(v_4)$	1	1	0	0	0	0	0
$\Gamma(v_5)$	0	1	1	0	0	0	0
$\Gamma(v_6)$	1	1	1	0	0	0	0
$\Gamma(v_7)$	0	1	1	0	0	0	0

(b)

Fig. 4. (a) A non-bipartite graph G which contains two maximal quasi-bicliques, g_1, with $V(g_1) = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}\}$ and g_2, with $V(g_2) = \{\{v_1, v_2, v_3\}, \{v_4, v_6, v_7\}\}$, when $ms = 3, \epsilon = 1$. (b) The binary matrix representation of the non-bipartite graph.

algorithm of [4], [5], [21], and then every maximal biclique subgraph (deemed as a 'core') is expanded to obtain α-quasi-bicliques. However, this approach cannot enumerate the complete set of our defined maximal quasi-bicliques. We use the graph G in Figure 3(b) to illustrate the reason. The two vertex sets $\{v_1, v_2\}$ and $\{v_5, v_6\}$ in G form a maximal quasi-biclique subgraph g' where $\epsilon = 1$. However, g' does not contain any maximal biclique subgraphs since the only two maximal biclique subgraphs in the graph are not a subset of g'—one maximal biclique subgraph has vertex sets $\{v_1\}$ and $\{v_4, v_5, v_6\}$, and the other has vertex sets $\{v_6\}$ and $\{v_1, v_2, v_3\}$. Thus a maximal quasi-biclique subgraph may not always contain a maximal biclique subgraph.

Bu *et al.* [7] mine quasi-biclique subgraphs in the ppi data set, where each vertex of its quasi-biclique can be disconnected up to a certain percentage of vertices in its counterpart vertex set. Hence, its noise tolerance is balanced and symmetrical. However, their quasi-biclique subgraphs are not maximal. Bu *et al.* [7] use spectral analysis to mine quasi-bicliques, but it is not clear how their algorithm works, since only a general description of it is given. To mine quasi-bicliques, the eigenvectors of the adjacency matrix of the input graph are calculated, and each eigenvector corresponds to a vertex of the graph that is an 'intrinsic characteristic of interactions' [7], but this claim is not proven. The top 10% of the vertices in the graph with the highest negative eigenvectors are selected, and quasi-bicliques are mined from them. Thus, they are using a heuristic approach which does not mine the complete set of their defined quasi-bicliques.

Table I summarizes the differences among the various types of quasi-bicliques, and the different types of algorithmic approaches (fifth column in the table) to mine them. In fact, if the application requires unbalance error tolerance in quasi-bicliques, our maximal quasi-bicliques can easily handle it by setting error tolerance of one side of the quasi-biclique to be large and the other side to be small.

In [22], we introduce an alternate version of maximal quasi-biclique whose error tolerance is percentage based. As this alternate version does not have anti-monotone property, there is no efficient algorithm to mine it.

B. The "Quasi" Concept

Recently, Pei *et al.* [23] proposed cross-graph quasi-clique, which is a set of graphs and each graph has vertex set V. In each graph, each vertex connects to at least $\gamma.(|V| - 1)$ other vertices in V, thus, their error tolerance is balanced. Although the "quasi" concept is used, their error tolerance is percentage based, while maximal quasi-biclique's is absolute based. Besides this difference, our graph of interests are also different. Cross-graph quasi-clique is a set of closely connected vertices (representing entities of one kind) across a set of non-bipartite graphs, while maximal quasi-biclique subgraph is two sets of closely connected vertices (representing entities of two kinds, e.g. stocks and financial ratios) in a bipartite graph.

Another area related to maximal quasi-bicliques is frequent itemsets that tolerate errors. Yang *et al.* [24] raised the idea of mining error tolerant frequent itemsets (ETIs). ETIs and its variants [25], [26] are a general form of frequent itemsets, which allow some errors in the frequent itemsets.

Approximate frequent itemset (AFI) [26] is a stricter variation of ETI, as it has error tolerant constraint on both the itemset and its transaction set. One may misunderstood that the problem of mining maximal quasi-biclique can be solved by considering the binary matrix representation of the graph as a transaction dataset, and using the AFI mining algorithm to mine AFIs, where each AFI and its transaction set form a quasi-biclique subgraph. To clear this misunderstanding, we need to explain in details the main differences between these two works.

(1) We mine maximal quasi-biclique subgraphs from both bipartite and non-bipartite graph. Figure 4(a) shows an example of a non-bipartite graph G which has two maximal quasi-biclique subgraphs g_1, with $V(g_1) = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}\}$ and g_2, with $V(g_2) = \{\{v_1, v_2, v_3\}, \{v_4, v_6, v_7\}\}$, when $ms = 3, \epsilon = 1$. The binary matrix of this graph G is shown in 4(b), and if we mine AFIs with minimum support of 0.4 and $\epsilon_r = \epsilon_c = 1/3$ from it, the following AFIs will be generated: $\{v_2, v_3\}, \{v_2, v_4, v_5\}, \{v_2, v_4, v_6\}, \{v_2, v_5, v_6\}, \{v_2, v_6, v_7\}$, $\{v_4, v_5, v_6\}$, $\{v_4, v_6, v_7\}$, $\{v_2, v_4, v_5, v_6\}$, $\{v_2, v_4, v_6, v_7\}$. These are useful error tolerant frequent itemsets, but they do not represent quasi-biclique subgraphs.

(2) A closed itemset and its transaction set form a biclique [21], but a AFI and its transaction set do not form a quasi-biclique, due to its error tolerance characteristic. For example, AFI $\{v_2, v_4, v_5\}$ with its transaction set $\{\Gamma(v_1), \Gamma(v_2), \Gamma(v_3)\}$ do not form a quasi-biclique.

(3) AFIs are not maximal, so it is possible that exponential number of AFIs which are subsets of each other are generated. For example in the result of Figure 4(b), 4 AFIs are subsets of $\{v_2, v_4, v_5, v_6\}$.

(4) The error tolerance of ETI and AFI are percentage-based, which means they do not have anti-monotone property. This poses a critical issue in efficient mining of ETIs and AFIs, and currently there are no existing algorithms that mine the complete set of ETIs or AFIs. For example, the AFI breadth-first mining algorithm [26] does not mine some cases of AFIs. In Figure 4(b), AFI $\{v_1, v_2, v_3\}$ with transaction set $\{\Gamma(v_2), \Gamma(v_4), \Gamma(v_5), \Gamma(v_6)\}$ is not mined although it satisfies the settings mentioned above. The algorithm considers $\{\Gamma(v_2), \Gamma(v_4), \Gamma(v_5), \Gamma(v_6), \Gamma(v_7)\}$ as the transaction set of $\{v_1, v_2, v_3\}$, as each transaction in the transaction set fulfills the ϵ_r constraint, but this transaction set fails the ϵ_c constraint.

Besson *et al.* [27] introduced error tolerance into formal concepts by proposing DR-bi-sets, which are bi-sets tolerating errors. DR-bi-sets are defined by two properties: a dense property and a relevant property. Although maximal quasi-bicliques and DR-bi-sets are from two different fields, both approaches can obtain the same output under certain constraints—when the graph (which is represented by a binary matrix) does not contain any self-loops and the relevant property of DR-bi-sets is disregarded.

C. Subspace Clustering and Co-Clustering

Due to the curse of dimensionality, subspace clustering is proposed to discover clusters within different subspaces of high dimensional datasets [16]. On the other hand, Cheng and Church [17] proposed to co-cluster (also known as bicluster) genes and conditions, where in a co-cluster, the set of genes have similar set of conditions. Although subspace clustering [16] and co-clustering [17] are motivated by different problems, they are actually solving similar problems. A co-cluster containing a set of genes and a set of conditions can be viewed as a subspace cluster defined by the same set of genes and the same set of conditions.

Subspace clustering and co-clustering algorithms thus can be applied to the stocks and financial ratios dataset, where the stocks are the objects and the financial ratios are the dimensions. However, none of the existing subspace and co-clustering algorithms tolerate missing or erroneous data, which are common in financial datasets.

D. Self-Organising Map (SOM)

Self-organizing maps (SOMs) [13]–[15] have been previously proposed to group stocks based on their financial ratios. SOM is a visualization tool which allows users to see how entities are clustered together, but it is hard for users to define clear clusters of entities because the boundaries of the clusters are difficult to distinguish. A recent method called *clustering on SOM* [28], [29] can be used to remedy this problem, where some well-defined hierarchical or partitive clusters can be obtained. However, the clusters of stocks are determined by the whole set of financial ratios, so analysts cannot determine specifically which financial ratios a cluster of stocks are highly similar in.

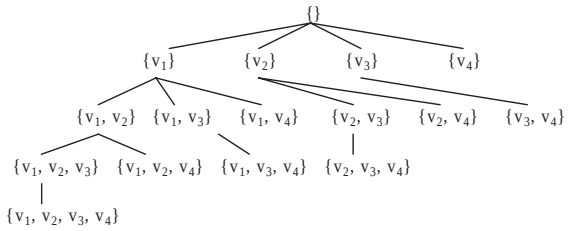


Fig. 5. The search space tree ($V_X = \{v_1, v_2, v_3, v_4\}$)

IV. Mining Maximal Quasi-Biclique Subgraphs: $MQBminer$

In this section, we present our maximal quasi-biclique subgraphs mining algorithm $MQBminer$. We first describe the algorithm in the context of bipartite graphs, and then discuss how to handle non-bipartite graphs.

Given a bipartite graph G with two disjoint vertex sets V_x and V_y, any subset of V_x (V_y) may form maximal quasi-biclique subgraphs with one or more subsets of V_y (V_x), so the search space is the power set of V_x and V_y. $MQBminer$ picks one vertex set as the primary enumeration vertex set, let it be V_x, and enumerates the subsets of V_x that have the potential to form quasi-clique subgraphs. Then for each generated subset of V_x, denoted as X, $MQBminer$ enumerates the subsets of V_y that can form quasi-biclique subgraphs with X. The main challenge here is how to identify and prune those subsets of V_x and V_y that cannot form maximal quasi-biclique subgraphs. Note that once one side of a quasi-biclique subgraph is fixed, the search space of the other side is greatly limited. Therefore, the size of the primary enumeration vertex set has a bigger impact on the efficiency of the algorithm than that of the other vertex set. $MQBminer$ always picks the smaller vertex set as the primary vertex set. In the remaining of this section, we assume that V_x is picked as the primary enumeration vertex set.

The power set of V_x can be represented as a set-enumeration tree. Figure 5 shows the set-enumeration tree when $V_x = \{v_1, v_2, v_3, v_4\}$. Each node in the tree represents a subset of V_x. The vertices in the set-enumeration tree are sorted according to some order. For every vertex set X in the tree, only vertices after the last vertex of X can be used to extend X. This set of vertices are called *candidate extensions* of X, denoted as $cand\_exts(X)$. For example, in Figure 5, vertices are sorted based on their subscripts, so vertex v_4 is in $cand\_exts(\{v_1, v_3\})$, but vertex v_2 is not a candidate extension of $\{v_1, v_3\}$ because vertex v_2 is before vertex v_3 in the order. $MQBminer$ explores the set-enumeration tree in depth-first order. It first enumerates all the vertex sets containing v_1, and then enumerates all the vertex sets containing v_2 but not containing v_1, and so on. The vertex set $\{v_4\}$ is enumerated last.

Given a subset X of V_x, we use $N(X)$ to denote the set of vertices in V_y that are connected to at least $|X| - \epsilon$ vertices in X, that is, $N(X) = \{u | u \in V_y \wedge \Gamma_X(u) \geq |X| - \epsilon\}$. It is easy to see that if X and Y can form a quasi-biclique subgraph, then we have $Y \subseteq N(X)$. During the mining process, for

every X explored, $MQBminer$ maintains its $N(X)$, and uses the following lemmas to prune the search space.

Lemma 1: Given a vertex v and two vertex sets Y and Y' such that $Y \subseteq Y'$, we have $|Y| - |\Gamma_Y(v)| \leq |Y'| - |\Gamma_{Y'}(v)|$.

Proof $|Y'| - |\Gamma_{Y'}(v)| = |Y| + |Y' - Y| - (|\Gamma_Y(v)| + |\Gamma_{Y'-Y}(v)|) = |Y| - |\Gamma_Y(v)| + (|Y' - Y| - |\Gamma_{Y'-Y}(v)|) \geq |Y| - |\Gamma_Y(v)|$.

Lemma 2: Given a vertex set $X \subseteq V_x$, if $|N(X)| < ms$, then for every superset X' of X, we have $|N(X')| < ms$.

Proof For every vertex $u \in N(X')$, we have $|X| - |\Gamma_X(u)| \leq |X'| - |\Gamma_{X'}(u)| \leq \epsilon$ based on Lemma 1. Thus we have $u \in N(X)$, which implies that $N(X') \subseteq N(X)$. Therefore, if $|N(X)| < ms$, then we have $|N(X')| < ms$.

The above lemma states that subsets of V_x have the anti-monotone property. For every subset X of V_x, $MQBminer$ checks whether $|N(X)| < ms$ is true, if it is, then there is no need to extend X further. The proof of the above lemma shows that $N(X')$ is a subset of $N(X)$ if $X \subset X'$. $MQBminer$ utilizes this property to save mining cost by generating $N(X')$ from $N(X)$.

Lemma 3: Let $\{X, Y\}$ be a quasi-biclique subgraph with respect to ϵ, and $|X| \geq ms$, $|Y| \geq ms$. For every vertex $v \in X$, we have $|\Gamma_{N(X)}(v)| \geq ms - \epsilon$.

Proof Based on the definition of quasi-biclique, for every vertex $v \in X$, we have $|\Gamma_Y(v)| \geq |Y| - \epsilon \geq ms - \epsilon$. Since $Y \subseteq N(X)$, we have $|\Gamma_{N(X)}(v)| \geq |\Gamma_Y(v)| \geq ms - \epsilon$.

Based on the above lemma, $MQBminer$ removes a vertex v from $cand\_exts(X)$ if $|\Gamma_{N(X)}(v)| < ms - \epsilon$ because v cannot appear in any valid quasi-biclique subgraph containing X. $MQBminer$ also checks whether there exists a vertex $v \in X$ such that $|\Gamma_{N(X)}(v)| < ms - \epsilon$. If such v exists, then there is no need to extend X further because no quasi-biclique subgraphs can be generated from X.

Lemma 4: Let $\{X, Y\}$ be a quasi-biclique subgraph with respect to ϵ, and $|X| \geq ms$, $|Y| \geq ms$. For every pair of vertices $v_1, v_2 \in X$, we have $|\Gamma_{N(X)}(v_1) \bigcap \Gamma_{N(X)}(v_2)| \geq ms - 2\epsilon$.

Proof Based on the definition of quasi-biclique subgraphs, for every vertex $v \in X$, we have $|\Gamma_Y(v)| \geq |Y| - \epsilon$. Therefore, $|\Gamma_Y(v_1) \bigcap \Gamma_Y(v_2)| = |\Gamma_Y(v_1)| + |\Gamma_Y(v_2)| - |\Gamma_Y(v_1) \bigcup \Gamma_Y(v_2)| \geq |\Gamma_Y(v_1)| + |\Gamma_Y(v_2)| - |Y| \geq 2(|Y| - \epsilon) - |Y| = |Y| - 2\epsilon \geq ms - 2\epsilon$. Since $Y \subseteq N(X)$, we have $|\Gamma_{N(X)}(v_1) \bigcap \Gamma_{N(X)}(v_2)| \geq ms - 2\epsilon$.

Based on the above lemma, $MQBminer$ checks every pair of vertices $v_1, v_2 \in X$. If $|\Gamma_{N(X)}(v_1) \bigcap \Gamma_{N(X)}(v_2)| < ms - 2\epsilon$, then there is no need to extend X further. $MQBminer$ also removes from $cand\_exts(X)$ those vertices u such that there exists vertex $v \in X$ and $|\Gamma_{N(X)}(u) \bigcap \Gamma_{N(X)}(v)| < ms - 2\epsilon$.

Algorithm 1 shows the pseudo code of $MQBminer$. When Algorithm 1 is first called on graph G with vertex sets V_x and V_y, X is set to $\{\}$, $N(X)$ is set to V_y and $cand\_exts(X)$ is set to V_x. At line 4, $MQBminer$ generates $N(X')$ from $N(X)$ based on the anti-monotone property. Before extending vertex set X', $MQBminer$ first checks whether X' is extendable based on Lemma 3 and Lemma 4 (line 5). When generating $cand\_exts(X')$, $MQBminer$ also uses Lemma 3 and Lemma 4 to remove those vertices that cannot be added to X' to form quasi-biclique subgraphs (line 8).

Algorithm 1 Algorithm $MQBminer$

Input:
X is a subset of V_x that is currently being explored;
$N(X)$ is the set of vertices in V_y that are connected to at least $|X| - \epsilon$ vertices in X;
$cand\_exts(X)$ is the set of candidate extensions of X;
ms is the minimum size threshold;
ϵ is the error tolerant value;

Description:
1: **for all** $v \in cand\_exts(X)$ **do**
2: $X' = X \cup \{v\}$;
3: $cand\_exts(X) = cand\_exts(X) - \{v\}$;
4: $N(X') = \{u | u \in N(X) \wedge |\Gamma_{X'}(u)| \geq |X'| - \epsilon\}$;
5: **if** $|N(X')| \geq ms$ AND for every $v \in X'$, $|\Gamma_{N(X')}(v)| \geq ms - \epsilon$ AND for every pair of vertices $v_1, v_2 \in X'$, $|\Gamma_{N(X')}(v_1) \bigcap \Gamma_{N(X')}(v_2)| \geq ms - 2\epsilon$ **then**
6: **if** $|X'| \geq ms$ **then**
7: Generate all $Y \subseteq N(X')$ such that $|Y| \geq ms$ and $\{X', Y\}$ is a maximal quasi-biclique subgraph;
8: $cand\_exts(X') = \{u | u \in cand\_exts(X) \wedge |\Gamma_{N(X')}(u)| \geq ms - \epsilon \wedge \forall v \in X', |\Gamma_{N(X')}(u) \bigcap \Gamma_{N(X')}(v)| \geq ms - 2\epsilon\}$;
9: **if** $|X'| + |cand\_exts(X')| \geq ms$ **then**
10: $MQBminer(X', N(X'), cand\_exts(X'), ms, \epsilon)$;

A. Generating maximal quasi-biclique subgraphs and maximality checking

For every vertex set $X \subseteq V_x$ explored during the mining process, all the vertices in $N(X)$ satisfy the error constraint with respect to X, but it is possible that some of the vertices in X do not satisfy the error constraint with respect to $N(X)$. That is, there exists some vertex $v \in X$ such that $|N(X)| - |\Gamma_{N(X)}(v)| > \epsilon$. In this case, $MQBminer$ needs to search for the subsets of $N(X)$ that can form quasi-biclique subgraphs with X (line 7), and these subsets of $N(X)$ must be maximal with respect to X. Here we say a subset Y of $N(X)$ is maximal if there does not exist another vertex set $Y' \subseteq N(X)$ such that $Y \subset Y'$ and $\{X, Y'\}$ is also a quasi-clique.

$MQBminer$ generates the maximal subsets of $N(X)$ that can form quasi-biclique subgraphs with X as follows. It first identifies the set of vertices in X that do not satisfy the error constraint with respect to $N(X)$, denoted as $\bar{X} = \{v | v \in X \wedge |N(X)| - |\Gamma_{N(X)}(v)| > \epsilon\}$. Then $MQBminer$ identifies the set of vertices in $N(X)$ that are connected to all the vertices in $\bar{X}$, denoted as $\bar{Y} = \{u | u \in N(X) \wedge \forall v \in \bar{X}, u \; is \; connected \; to \; v\}$. Vertex set $\bar{Y}$ should be included in all the maximal subsets of $N(X)$ that can form quasi-biclique subgraphs with X.

Lemma 5: Given $Y \subseteq N(X)$, if $\{X, Y\}$ is a quasi-biclique subgraph, then $\{X, Y \bigcup \bar{Y}\}$ is also a quasi-biclique subgraph.

Proof For every vertex $u \in Y \bigcup \bar{Y}$, u satisfies the error constraints based on the definition of $N(X)$. For every vertex $v \in X$, there are two cases. The first case is that $v \in \bar{X}$. In this case, $|Y \bigcup \bar{Y}| - |\Gamma_{Y \bigcup \bar{Y}}(v)| = |Y \bigcup \bar{Y}| - |\Gamma_Y(v)| - |\Gamma_{\bar{Y}-Y}(v)| = |Y \bigcup \bar{Y}| - |\Gamma_Y(v)| - |\bar{Y} - Y| = |Y| - |\Gamma_Y(v)| \leq \epsilon$. The other case is that $v \in X - \bar{X}$. In this case, we have

$|N(X)| - |\Gamma_{N(X)}(v)| \leq \epsilon$ based on the definition of $\bar{X}$. Since $Y \bigcup \bar{Y} \subseteq N(X)$, we have $|Y \bigcup \bar{Y}| - |\Gamma_{Y \bigcup \bar{Y}}(v)| \leq |N(X)| - |\Gamma_{N(X)}(v)| \leq \epsilon$ based on Lemma 1. Therefore $\{X, Y \bigcup \bar{Y}\}$ is a quasi-biclique subgraph.

Now the problem is reduced to finding the subsets Y of $N(X) - \bar{Y}$ such that $\{X, Y \cup \bar{Y}\}$ is a quasi-biclique subgraph, and $Y \cup \bar{Y}$ is maximal. The subsets of $N(X) - \bar{Y}$ can also be represented as a set-enumeration tree, and $MQBminer$ uses the depth-first order to explore the set-enumeration tree. The subsets of $N(X)$ also have the anti-monotone property. Based on this property, $MQBminer$ extends a subset of $N(X) - \bar{Y}$ only if the subset can form a quasi-biclique subgraph with X.

Lemma 6: If X cannot form a quasi-biclique subgraph with Y, then for every superset Y' of Y, X cannot form a quasi-biclique subgraph with Y'.

Proof The only reason that X cannot form a quasi-biclique subgraph with Y is that there exists $v \in X$ such that $|Y| - |\Gamma_Y(v)| > \epsilon$. Since $Y \subseteq Y'$, we have $|Y'| - |\Gamma_{Y'}(v)| \geq |Y| - |\Gamma_Y(v)| > \epsilon$ based on Lemma 1. Therefore, X cannot form a quasi-biclique subgraph with Y'.

The remaining problem is how to check the maximality of Y with respect to X and the maximality of X with respect to Y. There are two typical existing approaches. One is to store all the quasi-biclique subgraphs that have been previously generated, and then for each newly generated quasi-biclique subgraph g, we check whether there exists an existing quasi-biclique subgraph g' such that g' is a super graph of g. The drawback of this approach is that the stored quasi-biclique subgraphs can be very large, which not only consumes lot of memory, but also slows down the checking operation. The other approach is to utilize the graph itself to check whether g is maximal.

Here we adopt the second approach. To check whether Y is maximal with respect to X, we check whether there exists a vertex $u \in (N(X) - Y)$ such that $\{X, Y \cup \{u\}\}$ is a quasi-biclique subgraph. If such u exists, then Y is not maximal. If Y is maximal with respect to X, then we check whether X is maximal with respect to Y by checking whether there exists some vertex $v \in exts(X)$ such that v can be added to X to form a quasi-biclique subgraph with Y, where $exts(X) = \{v | v \in V_x - X \wedge |\Gamma_{N(X)}(v)| \geq ms - \epsilon \wedge \forall u \in X, |\Gamma_{N(X)}(u) \bigcap \Gamma_{N(X)}(v)| \geq ms - 2\epsilon\}$, and it is derived based on Lemma 3 and Lemma 4.

B. Example

We use the example graph shown in Figure 3(a) to demonstrate how $MQBminer$ mines maximal quasi-biclique subgraphs from a bipartite graph. In the example graph, $V_x = \{v_1, v_2, v_3, v_4\}, V_y = \{v_5, v_6, v_7, v_8, v_9\}$. The mining parameters are set as follows: $ms = 3$ and $\epsilon = 1$. Figure 6 shows how $MQBminer$ traverses the search space of G.

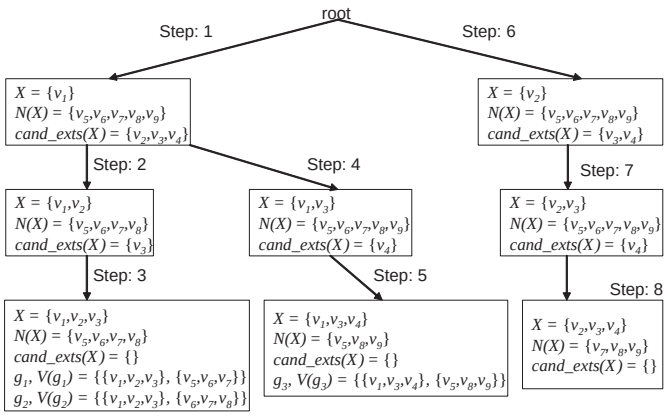


Fig. 6. The traversal of the search space by $MQBminer$ on graph shown in Figure 3(a).

Step 1: $MQBminer$ starts from vertex set $X = \{v_1\}$. Here $N(X) = V_y$ and $cand\_exts(X) = \{v_2, v_3, v_4\}$.

Step 2: $MQBminer$ extends X by adding v_2 to X. Vertex v_9 is removed from $N(X)$ because it is disconnected from more than one vertex in X. Vertex v_4 is pruned from $cand\_exts(X)$ as it is connected to only one vertex in $N(X)$, which is less than $ms - \epsilon$=2.

Step 3: $MQBminer$ adds v_3 to X, and no vertices in $N(X)$ can be removed. The size of X satisfies the size constraint. However, $|X|$ cannot form a valid quasi-biclique subgraph with $N(X)$ because v_2 is disconnected from both v_5 and v_8. $MQBminer$ needs to search for the subsets of $N(X)$ to form quasi-biclique subgraphs with X. It first identifies $\bar{X}$ and $\bar{Y}$ and gets $\bar{X} = \{v_2\}$, and $\bar{Y} = \{v_6, v_7\}$, then it enumerates the subsets of $N(X) - \bar{Y} = \{v_5, v_8\}$ and add them to $\bar{Y}$. Two quasi-biclique subgraphs are generated. There is no need to extend X further because $cand\_exts(X) = \{\}$.

Step 4: $MQBminer$ backtracks to step 1, and extends X to $X = \{v_1, v_3\}$. Now $N(X) = V_y$ and $cand\_exts(X) = \{v_4\}$.

Step 5: $MQBminer$ adds v_4 to X. Vertices v_6 and v_7 are removed from $N(X)$. Here both X and $N(X)$ satisfy the size constraint and the error constraint. A maximal quasi-biclique subgraph is generated.

Step 6: $MQBminer$ returns to the root and starts to enumerate vertex sets not containing v_1 but containing v_2.

Step 7: $MQBminer$ extends X to $X = \{v_2, v_3\}$. $N(X)$ is still equal to V_y.

Step 8: $MQBminer$ adds v_4 to X. Vertices v_5 and v_6 are removed from $N(X)$. Both X and $N(X)$ satisfy the size constraint, but vertex $v_2 \in X$ is connected to only one vertex in $N(X)$. Hence no maximal quasi-biclique subgraphs are generated in this step.

Step 9: $MQBminer$ returns to root and stops as $|X| + |cand\_exts(X)| < ms$.

C. Mining Maximal Quasi-Biclique Subgraphs from Non-Bipartite Graphs

In the case that graph G is not a bipartite graph, every vertex in $V(G)$ can be on either side of a maximal quasi-biclique subgraph. In Algorithm 1, V_x and V_y are replaced by V, and in $N(X)$, we remove vertices that are in X. This may result in duplicated maximal quasi-biclique subgraphs being enumerated. A simple post-processing step is implemented to remove the duplicates.

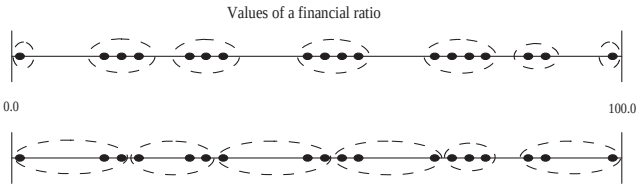


Fig. 7. Two examples of how the values of a financial ratio can be clustered into intervals. Each dot on the line indicates the value of a stock financial ratio. The desired intervals are shown on the first line and equidepth binning method is applied with three values in an interval.

D. Discretization of Data Containing Continuous Values

In our previous work [10], we apply a simple discretization technique known as equidepth binning [30] to partition the range of continuous values into intervals such that each interval has n number of value, where n is set by the user. The weakness of this technique is apparent in Figure 7, which shows an example of discretization of the continuous values of a financial ratio into intervals. The range is from 0 to 100. Applying equidepth binning with $n = 3$ results in the intervals shown on Figure 7 second line, which are of poor quality because many values far apart are in the same intervals.

The desired intervals are shown on the first line of Figure 7, where values close together in relative to the range are in the same intervals. We attempt to achieve this 'natural' partitions with minimum user interference, by adopting the agglomerative hierarchical clustering (AGNES) algorithm [31]. AGNES consists of a series of iterations. At each iteration, two closest clusters are merged together based on the unweighted pair-group average method. The algorithm starts with singletons as clusters and ends with all values in one cluster. By applying AGNES on the continuous values, a cluster corresponds to an interval and each iteration of AGNES gives a discretization result.

We then introduce a scoring method intra-inter ratio (iir) to score the clusters obtained in each iteration of AGNES, and the iteration that minimizes the score is selected as the best discretization result. iir uses the same concept as the multi representation clustering validity index [29], [32]. In this index, the quality of the partitioning is based on the intra distance of the clusters and the inter distance between clusters. Optimum partition is achieved by minimizing the intra distance of the clusters and maximizing the inter distance between clusters. However, the formulation of iir is much simpler than the multi representation clustering validity index. iir is defined as

$$iir(C) = \min_{C_s \in C} \left(\frac{Intra(C_s)}{Inter(C_s)} \right)$$

where C is the complete set of clusters obtained from the iterations of AGNES on the continuous values. C_s is the set of clusters obtained from an iteration of AGNES. The functions $Intra(C_s)$ and $Inter(C_s)$ are defined as

$$Intra(C_s) = \sum_{c_i \in C_s} f(c_i)/|c_i|$$

$$f(c_i) = \begin{cases} 1 & \text{if } |c_i| = 1 \\ \sum_{x \in c_i} \frac{|x - \mu_{c_i}|}{|c_i|} & otherwise \end{cases}$$

$$Inter(C_s) = \sum_{c_i \in C_s} \left(\sum_{c_j \in C_s, c_i \neq c_j} \frac{|\mu_{c_i} - \mu_{c_j}|}{|C_s| - 1} \right) / |C_s|$$

where c_i is a cluster in C_s, μ_{c_i} is the centroid of cluster c_i.

Silhouette Coefficient [31] and SSE [33] are two alternative scoring methods but they are sensitive to outliers. When outliers exist in the data, the optimal score obtained from them normally coincide to a discretization result where the number of partitions can be very large or small. iir is a heuristic method proposed with the aim of obtaining the 'natural' partitions and to reduce the sensitivity towards outliers. Using AGNES may be computationally slow when the dataset is large and there may be lack of memory space to handle it. In such situation, we can use the memory-constrained UPGMA algorithm [34] instead of AGNES, which handles large datasets.

V. Experimental Results

We conducted six experiments on maximal quasi-bicliques: (1) We compared three different scoring methods for the AGNES discretization method. (2) We evaluated the quality of different quasi-bicliques mined from noisy data, by comparing how well they are able to recover maximal bicliques mined from the original data. (3) We investigated the efficiency of the algorithm $MQBminer$ by testing it on 3 graph datasets. (4) We conducted case studies on the real stock market to examine the usefulness of maximal quasi-bicliques. (5) We explored the potential of using maximal quasi-bicliques as input vectors and dimensions selection for SOM. (6) We used maximal quasi-bicliques to mine the protein networks, and show that their results are better than maximal bicliques. Our experiments were performed on Windows XP environment, using Intel Xeon CPU 3.4GHz with 2GB RAM. $MQBminer$ was coded in C++.

A. Graph datasets used

We used five graph datasets for our experiments. The first dataset contains the financial ratios belonging to 470 stocks of S&P 500 [35] from year 2001. This dataset was obtained from Compustat [36], and it contains 12 financial ratios of the 470 stocks. Table II shows the financial ratios, which are categorized into five different types of ratios. The growth ratios were obtained by calculating the percentage change from previous year's value to current year's value.

The second dataset is the yeast ppi (protein-protein interaction) dataset. The yeast ppi was downloaded from the protein information repository DIP (database of interacting proteins) [37]. This dataset is modeled by a non-bipartite graph with 4,919 vertices and 17,163 edges. The vertices of the graph are proteins and an edge between two vertices exists if the

Type	Ratio
Liquidity Ratio	Current Ratio (Cur)
Finance Ratio	Debt to Equity Ratio (DE)
Profitability Ratio	Return on Assets (ROA)
	Return on Equity (ROE)
Investment Ratio	Dividend Yield (DY)
	Price to Earnings Ratio (PE)
	Price to Book Ratio (PB)
	Price to Cashflow Ratio (PC)
	Price to Sales (PS)
Growth Ratio	Net Income Growth (NIG)
	Earnings Before Interest and Tax Growth (EBITG)
	Sales Growth (SG)

TABLE II

FINANCIAL RATIOS USED IN OUR DATASET.

Financial ratio	# of values	# of intervals by		
		SSE	Silhouette Coefficient	iir
Cur	380	4	379	4
ROA	380	6	378	41
ROE	387	3	385	11
DE	437	2	436	41
DY	340	4	339	15
PB	463	5	462	4
PC	357	4	356	28
PE	398	3	396	21
PS	460	4	459	7
NIG	205	3	203	17
SG	293	4	292	14
EBITG	232	4	225	13

TABLE III

OPTIMAL NUMBER OF PARTITIONS BASED ON DIFFERENT SCORING METHODS.

two corresponding proteins interact with each other. All self-looping edges are removed as they are superfluous for our purpose.

The third dataset is a benchmark dataset c-fat200-1, which was obtained from the 2^{nd} DIMACS Challenge benchmarks [38]. It is a non-bipartite graph with 200 vertices and 1,534 edges.

The fourth dataset is a synthetic bipartite graph containing 10,000 vertices in each of its disjoint vertex set. We embedded this graph with 50 maximal biclique subgraphs, where each maximal biclique subgraph contains 10 vertices in each of its disjoint vertex set. Thus, this synthetic bipartite graph has 5,000 edges. The fifth dataset is similar to the fourth dataset, but we randomly added 5,000 extra edges as noise in the dataset.

B. Discretization of the financial ratio dataset

The financial ratio values are in continuous values and every fundamental analyst has his own preference on how each ratio is to be partitioned. Hence, we adopted an unsupervised approach by using the AGNES algorithm with iir to partition the financial ratio values into intervals. This discretization method was applied separately to positive values and negative values, as generally there is a clear distinction between positive and negative values in fundamental analysis.

After discretization of the financial ratios, we represented the dataset as a StoR graph which contains 686 vertices (470 stocks and 216 financial ratio value intervals). In this dataset, 3.71% of the financial ratio values are either missing or unavailable, so there are only 5,431 edges (not $470 \times 12 = 5640$ edges) in this graph.

We also compared iir with the other two scoring methods used in discretization of continuous vales; SSE and silhouette coefficient. Table III shows the number of positive values in each financial ratio, and the optimal number of clusters (which corresponds to the number of partitions) obtained by the three scoring methods. For the SSE, the SSE values obtained from each iteration of AGNES was plotted in a graph. The iteration that produces a distinct knee in the graph is selected as the optimal number of clusters [33]. For the silhouette coefficient, the iteration of AGNES that maximizes the silhouette coefficient is selected as the optimal number of clusters [30].

We can see that the SSE scoring method is biased towards a very small number of clusters, whereas the silhouette coefficient scoring method is biased towards a very large number of clusters and most of the clusters are singletons. For the iir scoring method, its optimal number of clusters is not skewed towards a large or small number, thus it is more robust towards outliers.

C. Evaluation of the quality of different models of quasi-bicliques

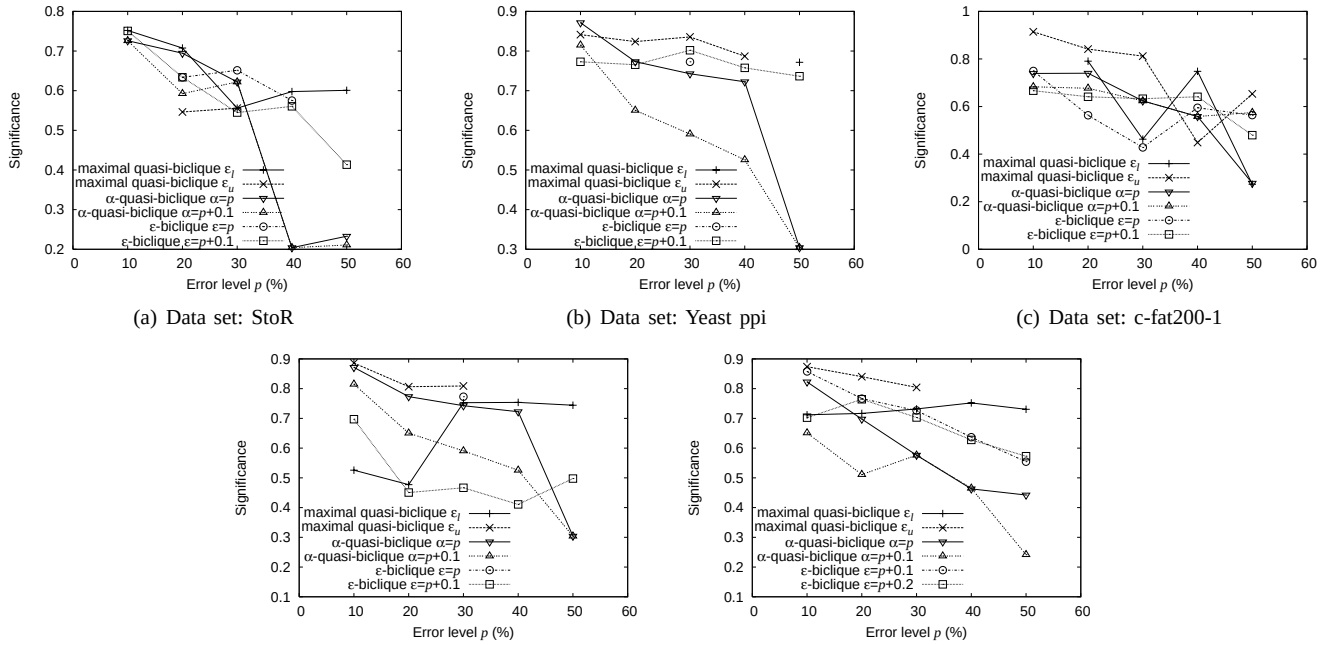
We evaluated how good the different models of quasi-bicliques are in tolerating errors. The evaluation procedure was conducted as follows: (1) We mined complete sets of maximal biclique subgraphs from different graphs using the algorithm in [4]. (2) Errors were introduced to the graphs by removing edges from the graphs randomly. Each edge in a graph has a probability p of been removed. (3) Different types of quasi-bicliques were mined from the erroneous graphs. (4) The different models of quasi-bicliques were evaluated on how well they are able to recover the original maximal biclique subgraphs.

Assume that there is a set of maximal biclique subgraphs $B = \{b_1, \ldots, b_{|B|}\}$ mined from a graph, and a set of quasi-biclique subgraphs $Q = \{q_1, \ldots, q_{|Q|}\}$ mined from the graph with errors. The following measures were used to evaluate the quality of the quasi-bicliques, which were modified from the error tolerant itemset evaluation measures proposed by Gupta *et al.* [39].

1) Recoverability: Recoverability measures the ability of recovering the original maximal bicliques based on a set of quasi-biclique subgraphs. Let $V(b) = \{X, Y\}$ and $|V(b)| = |X| + |Y|$. Let $r(b) = max\{|X \cap X'| + |Y \cap Y'||V(b) = \{X, Y\}, V(q) = \{X', Y'\}, q \in Q\}$. $r(b)$ is the largest number of common vertices found in a quasi-biclique subgraph q and a maximal biclique subgraph b. The recoverability of a set of quasi-biclique subgraphs Q is

$$Recoverability\ R = \sum_{b \in B} \frac{r(b)}{|V(b)|}$$

2) Spuriousness: Quasi-biclique subgraphs may have high recoverability because they are large to the extend that they contain all vertices of the maximal biclique subgraphs by



(a) Data set: StoR (b) Data set: Yeast ppi (c) Data set: c-fat200-1

(d) Data set: Synthetic. No extra edges (e) Data set: Synthetic. With 5000 extra edges

Fig. 8. Maximal quasi-bicliques, ϵ-bicliques and α-quasi-bicliques are mined from erroneous graphs, and they are used to recover maximal bicliques of these graphs without errors. The quality of the recovery is evaluated by the significance measure.

chance. So spuriousness is used to measure how many spurious or redundant vertices are in the quasi-biclique subgraphs.

To measure the spuriousness of a quasi-biclique q, we find a maximal biclique subgraph b such that both q and b have the most number of common vertices. We then count the number of vertices in q which is not in b, which quantifies the spuriousness of q. Let $s(q) = |V(q)| - max\{|X \cap X'| + |Y \cap Y'||V(q) = \{X', Y'\}, V(b) = \{X, Y\}, b \in B\}$. The spuriousness of a set of quasi-bicliques Q is

$$Spuriousness\ S = \sum_{q \in Q} \frac{s(q)}{|V(q)|}$$

3) Significance: Significance measures the trade-off between the recoverability and spuriousness of a set of quasi-biclique subgraphs Q,

$$Significance = \frac{2R(1-S)}{R + (1-S)}$$

The higher the significance of Q, the closer is the quality of Q to the set of maximal biclique subgraphs B.

We set ms to 5, 12, 6, 10, 10 and obtained 7, 4, 6469, 50, 50 maximal biclique subgraphs from the StoR, yeast ppi, c-fat200-1, synthetic and synthetic with 5000 extra edges graphs respectively. ms was set at the highest level which maximal biclique subgraphs can still be found from the graph. The error probability p is varied from 0.1 to 0.5 in each graph.

In this experiment, we compared our maximal quasi-bicliques with α-quasi-bicliques [9] and ϵ-bicliques [8]. We used the algorithm in [4] to mine maximal bicliques which are the 'cores' used to obtain α-quasi-bicliques, and we coded the approximate maximum biclique algorithm [8] which mines ϵ-bicliques.

As each quasi-biclique model has its own parameter settings, we need to find their optimal parameter settings, so that high quality quasi-bicliques can be mined. Let us assume that we are finding their optimal parameter settings to mine quasi-biclique subgraphs from a graph G with error probability p. For these three quasi-biclique models, we tried to set their error tolerant thresholds close to the noise probability p of the graph G. For maximal quasi-bicliques, we set the same ms used in mining the maximal biclique subgraphs from the graph G without errors. We also set two error tolerant thresholds ϵ_l, ϵ_u, such that $\frac{\epsilon_l}{ms} \leq p < \frac{\epsilon_u}{ms}$. For example, when $ms = 12$ and $p = 0.5$, we set $\epsilon_l = 6, \epsilon_u = 7$, so that $\frac{\epsilon_l}{ms} \leq p < \frac{\epsilon_u}{ms} \Rightarrow \frac{6}{12} \leq 0.5 < \frac{7}{12}$.

For the α-quasi-bicliques [9], we set $\alpha = p$ and $p + 0.1$. We need to mine maximal biclique subgraphs from the graph G, which are the 'cores'. These 'cores' are smaller than the original maximal biclique subgraphs mined from G without errors, since these 'cores' are mined from G with errors. We set ms of these 'cores' at the highest level where the number of 'cores' is at least as much as the number of original maximal biclique subgraphs.

The ϵ-biclique [8] model requires more effort in finding its optimal settings. The approximate maximum biclique algorithm randomly picks vertices to form three vertex sets that are used to find ϵ-bicliques and it requires users to define the sizes of these three vertex sets, which are denoted as $\widehat{m}, m$ and t. To find the appropriate settings, Mishra *et al.* state that analysis has to be conducted to determine them [8]. After trying different settings, we set $\widehat{m} = 2, m = 20, t =$size of the graph, which gives us good results in reasonable time. Approximate maximum algorithm also requires the user to define the number of ϵ-bicliques to be mined and we set it

to the number of original maximal biclique subgraphs mined from the graph G without errors. Lastly, we set $\epsilon = p$ and $p + 0.1$. However, in the synthetic graph with 5000 edges, no ϵ-biclique subgraphs can be found for this ϵ setting, so we increase ϵ to $p + 0.1$ and $p + 0.2$.

Figure 8 presents the significance measures of the different models of quasi-bicliques. There were no results for some experiments as they could not finish running within six hours (we limit each experiment running time to six hours) or no quasi-bicliques were mined from these experiments. Across the five graphs, our maximal quasi-bicliques have the highest significance in all settings, except in the StoR graph at $p = 0.3$ and yeast ppi graph at $p = 0.1$. This demonstrates the strength of maximal quasi-bicliques in recovering the original maximal biclique subgraphs from the graphs, even when the error probability p in the graphs is as high as 0.5. However, careful selection of ϵ is required as a ± 1 difference in ϵ can lead to fluctuations in the significance scores, as shown in Figure 8(c).

For the α-quasi-bicliques, their quality drops drastically as p increases. α-quasi-bicliques are highly dependent on their 'cores', and since the 'cores' are not noise tolerant, the quality of α-quasi-bicliques drops as p increases. The quality of ϵ-bicliques is lower than those of maximal quasi-bicliques in most of the experiments across the five graphs, which could be due to its noise tolerance being not symmetrical and balanced. Moreover, there are many experiments which the α-quasi-biclique model could not complete running after six hours, as the randomness nature of approximate maximum algorithm restricts its efficiency.

From these experiments, we can see that setting ϵ of maximal quasi-biclique at a threshold where $\frac{\epsilon_l}{ms} \leq p < \frac{\epsilon_u}{ms}$ gives good quality maximal quasi-bicliques, provided that the noise probability p of the dataset is known. If p is unknown, then the user should set the appropriate ms and ϵ to generate the required number of maximal quasi-bicliques.

D. Efficiency of $MQBminer$

We compared the performance of our proposed algorithm $MQBminer$ with $CompleteQB$. The existing algorithms [6]–[9] were not evaluated because they are incapable of finding our defined maximal quasi-bicliques. The efficiency of $MQBminer$ was evaluated on the StoR, yeast ppi and c-fat200-1 graphs.

The sub figures in the first row of Figure 9 show the number of maximal quasi-biclique subgraphs mined from the three graphs, and the sub figures in the second row show the time taken by $MQBminer$ and $CompleteQB$ to generate them.

From Figure 9, we can see that $MQBminer$ outperforms $CompleteQB$ in all situations, except in the StoR graph, at $ms = 6, 7$ and $\epsilon = 1$, but the difference in their running time is only less than 10 seconds. This clearly demonstrates that $MQBminer$ is highly efficient in traversing the search space of the graph. In some cases where ms are low, $CompleteQB$ could not even complete the mining task within 24 hours. Although $CompleteQB$ also exploits the anti-monotone property of maximal quasi-bicliques to perform the mining task, our experiment results show that this is not sufficient, and aggressive pruning techniques of $MQBminer$ are needed to speed up the running time.

We studied three factors that affect the running time of $MQBminer$, namely the minimum size threshold ms, the error tolerant threshold ϵ and the density of the graphs.

1) Effect of minimum size threshold (ms): On the same graph, we compared the running time and the number of maximal quasi-biclique subgraphs mined to study the effect of ms. Observe that the running time of $MQBminer$ scales up in a polynomial way when ms decreases across the three graphs; meanwhile, the number of maximal quasi-biclique subgraphs also scales up almost in the same polynomial way. This indicates that the running time of $MQBminer$ is roughly linear to the number of maximal quasi-biclique subgraphs mined.

2) Effect of error tolerant threshold (ϵ): We noted that the number of maximal quasi-biclique subgraphs increases when the error tolerant threshold rises. In fact, the running time of $MQBminer$ increases at an even higher rate when we increase ϵ but decrease ms. Therefore, it is computationally very expensive to mine small maximal quasi-biclique subgraphs that allow large number of errors.

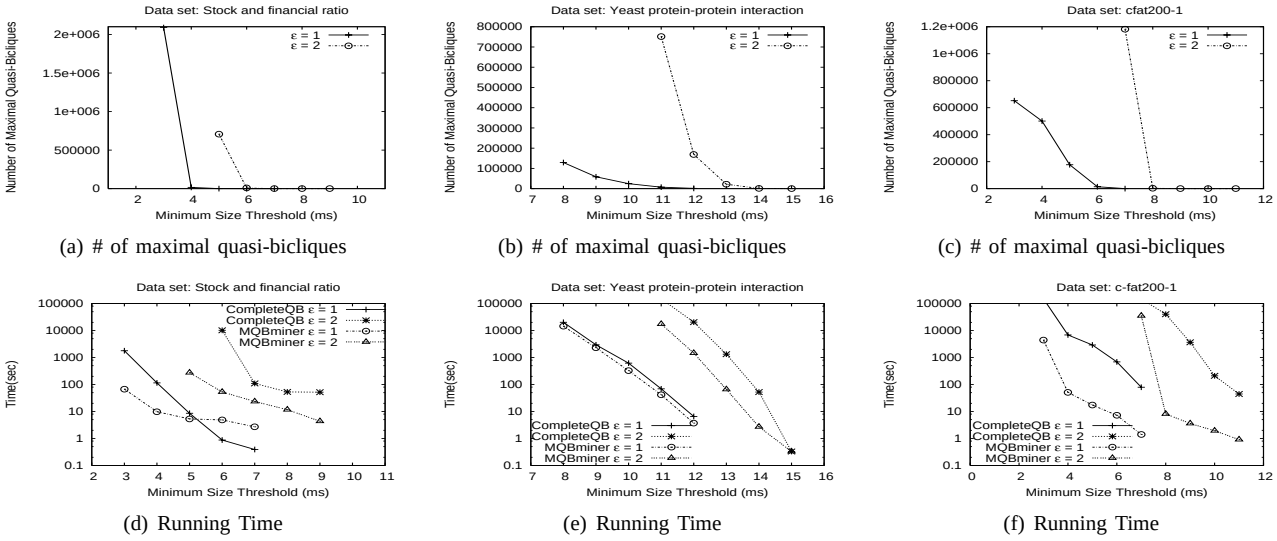
3) Effect of density of graph: The density of the graph affects the number of maximal quasi-biclique subgraphs mined, which in turn affects the running time of $MQBminer$. In a graph, we calculate the ratio between the number of maximal quasi-biclique subgraphs at a ms level over the number of maximal quasi-biclique subgraphs at the $ms + 1$ level. We then took the average of the ratios of each graph. At $\epsilon = 1$, the average ratios for the StoR, yeast ppi and c-fat200-1 graphs are 32.94, 8.87 and 20.67 respectively. And the edge density [1] for these graphs are 2.16%, 0.142% and 0.77% respectively. We can see that for graphs with higher density, the number of maximal quasi-bicliques increases more considerably as ms decreases. And since the running time of $MQBminer$ is roughly linear to the number of maximal quasi-biclique subgraphs, therefore, for a dense graph, the running time will increase substantially as ms decreases.

Summarizing our results, $MQBminer$ runs approximately linear to the number of maximal quasi-bicliques enumerated, which means that $MQBminer$ is sensitive to the number of outputs. To reduce the running time of $MQBminer$, user can select a high ms and low ϵ, so that a small number of maximal quasi-bicliques are generated, which will result in faster running time.

E. Mining Co-clusters from the Stock Market

In stock picking, a widely accepted assumption is that prices of stocks will rise in the long run if the stocks possess superior financial ratios [11], [12]. We generalize this hypothesis by studying whether stocks having similar financial ratios will have similar price performances in the stock market. Since stocks in a co-cluster have similarities in the financial ratios of the co-cluster, we examine if the price performances of the stocks in a co-cluster are similar. This study was conducted

[1] Edge density = (number of edges in the graph)/$(n(n-1)/2)$ for an n-vertex graph.



(a) # of maximal quasi-bicliques (b) # of maximal quasi-bicliques (c) # of maximal quasi-bicliques

(d) Running Time (e) Running Time (f) Running Time

Fig. 9. Running time and the number of maximal quasi-biclique subgraphs mined from the graphs.

with a limited data of 12 financial ratios and 470 stocks, as this is the largest amount of data we managed to obtain. Hence, part of our future work is to use a bigger set of data.

We used maximal quasi-bicliques, α-quasi-bicliques and ϵ-bicliques to mine co-clusters from the StoR graph. For the α-quasi-bicliques, we mined maximal biclique subgraphs from StoR graph at $ms = 4$ and used them as the 'cores'. For the ϵ-bicliques, we used the same parameter settings described in Section V-C but we varied α. Under different α, we ran the approximate maximum biclique algorithm [8] and outputted its result after six hours.

In a co-cluster C, we calculated the price performance of each stock s in C, which is denoted as $d(s) = \frac{p(s,2002)-p(s,2001)}{p(s,2001)}$, where $p(s, 2001)$ and $p(s, 2002)$ are the closing prices of s in 31^{st} December 2001 and 2002 respectively. We then calculate the *standard deviation* of the price performances of stocks in co-cluster C, denoted as

$$\sigma(C) = \sqrt{\frac{1}{|C|}\sum_{s \in C}(d(s) - \mu(C))^2}$$

where $\mu(C) = \frac{1}{|C|}\sum_{s \in C} d(s)$ is the mean price performance of stocks in co-cluster C.

The standard deviation of the price performance of stocks in a co-cluster (for brevity, we termed it as standard deviation of the co-cluster) measures the dispersion of the price performance of the stocks. Thus, a low standard deviation means that the price performance of these stocks are highly similar.

Table IV shows the number of co-clusters mined by the different quasi-biclique models and the average standard deviation of the co-clusters. The two highest average standard deviations of co-clusters were obtained by using maximal quasi-bicliques with $ms = 4, \epsilon = 1$ and $ms = 5, \epsilon = 2$. A large number of co-clusters were mined using these two settings, so there is a high possibility that erroneous co-clusters that do not contain stocks with similar price performances were also mined. Hence, the average standard deviations of these two sets of co-clusters are higher. However, the average

TABLE IV

STANDARD DEVIATIONS OF THE PRICE PERFORMANCES OF THE CLUSTERS OF STOCKS OBTAINED BY DIFFERENT TYPES OF QUASI-BICLIQUES

Types of quasi-biclique	Num	Standard deviation
Maximal quasi-biclique ($ms = 4, \epsilon = 1$)	3528	0.389
Maximal quasi-biclique ($ms = 5, \epsilon = 1$)	34	0.278
Maximal quasi-biclique ($ms = 6, \epsilon = 2$)	387	0.306
Maximal quasi-biclique ($ms = 5, \epsilon = 2$)	72950	0.37
α-quasi-bicliques ($ms = 3, \alpha = 0.1$)	190	0.354
α-quasi-bicliques ($ms = 3, \alpha = 0.2$)	190	0.354
α-quasi-bicliques ($ms = 3, \alpha = 0.3$)	190	0.354
α-quasi-bicliques ($ms = 4, \alpha = 0.1$)	11	0.328
α-quasi-bicliques ($ms = 4, \alpha = 0.2$)	11	0.328
α-quasi-bicliques ($ms = 4, \alpha = 0.3$)	11	0.345
ϵ-biclique ($\epsilon = 0.1$)	11	0.345
ϵ-biclique ($\epsilon = 0.2$)	11	0.328
ϵ-biclique ($\epsilon = 0.3$)	11	0.356

standard deviation of the co-clusters mined using setting $ms = 5, \epsilon = 2$ is lower than the one under setting $ms = 4, \epsilon = 1$. This means that stocks with more similar financial ratios may lead to higher similarity in price performance. In settings $ms = 5, \epsilon = 1$ and $ms = 6, \epsilon = 2$, the average standard deviations of their co-clusters are the lowest in Table IV, which substantiate our observation that stocks with more similar financial ratios have higher similar price performance.

Table IV also shows that maximal quasi-bicliques are more effective in mining co-clusters with higher similar price movements (which translates to lower standard deviations), compared to the other quasi-biclique models. For the co-clusters mined from maximal quasi-bicliques under settings $ms = 5, \epsilon = 1$ and $ms = 6, \epsilon = 2$, although their numbers are more than the co-clusters of the other quasi-biclique models, they have the lowest average standard deviations.

We selected some co-clusters mined from maximal quasi-bicliques and studied them in detail. We categorized our findings into two types of co-clusters: good and poor co-clusters. The good co-clusters contain groups of financial ratios whose values are in the healthy range. Likewise for the poor co-clusters, they contain groups of financial ratios whose values in the poor range. Table V shows some co-clusters of

TABLE V

SOME CO-CLUSTERS OF STOCKS AND FINANCIAL RATIOS MINED BY $MQBminer$ FROM THE STOR GRAPH.

Co-cluster	Stock symbols	Financial ratios and their value intervals
1	APA, KSE, PEG, PGL	Cur(0.232,3.276) DY(5.090,5.137) PB(1.617,1.813) EBITG(0.112,4.778)
2	BBT GDW NFB STZ	PC(12.069,14.66) PE(9.919,17.675) NIG(40.146,55.423) EBITG(18.274,49.823)
3	EMC MU TLAB XLNX	Cur(3.388-6.534) ROA(-6.879,-4.374) ROE(-7.381,-5.553) EBITG(-104.96,-99.081)
4	AHC, COP, GR, LMT	Cur(0.232,3.276) DE(108.216,116.576) NIG(-12.153,-10.591) EBITG(-8.561,-3.186)

stocks and their financial ratios' intervals.

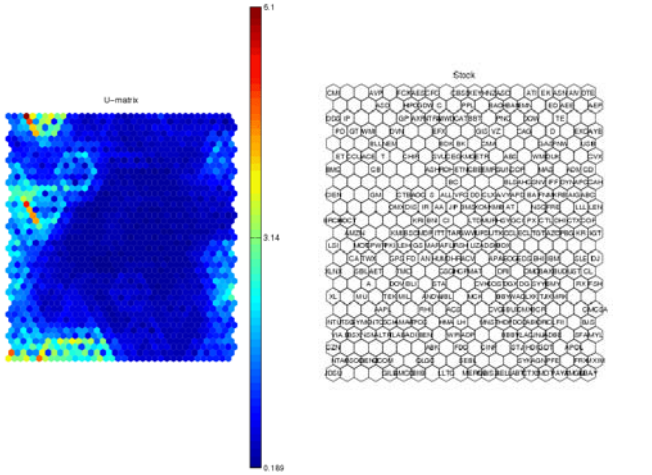
For simplicity of comparison, we say that stocks in a co-cluster have similar price performances if their prices all rise or fall together, by comparing their closing price of 31^{st} December 2001 to the closing price of 31^{st} December 2002. All price charts shown were taken from *MSN Money* [40].

Good co-clusters

- Co-cluster 1. The stocks in this co-cluster can be considered as undervalued stocks as they have very low PB, and at the same time, they have growth in their EBIT. Another attractive point is that they have a good DY of about 5%. Comparing these stock prices with the S & P 500 index of year 2002, we can see that three out of the four stocks in co-cluster 1 outperformed the S & P 500 index, as shown in Figure 10(a). Only PGL performed similarly with the S & P 500 index. The poor performance of PGL could be due to external factors which are not considered in our model. The positive note is that three stocks in the co-cluster performed much better than the S & P 500 index with an average of 6.67% increase in their prices, while the S & P 500 index dropped -22% for the year ended 2002.
- Co-cluster 2. Although the stocks have moderate PC and PE, they are good stocks due to their high NIG and EBITG. Figure 10(b) shows the price performances of these stocks for year 2002. Again, all the stock prices increased, unlike the dismal performance of S & P 500 index.

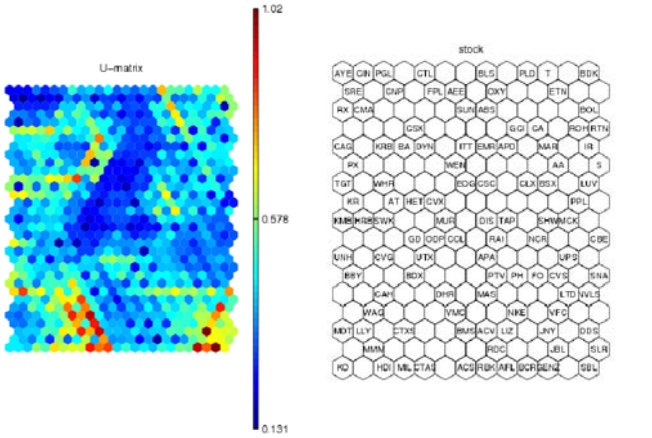
Poor co-clusters

- Co-cluster 3. This co-cluster has negative ROA and ROE, which indicate that the stocks were either making a loss for the financial year of 2001, or the stocks have negative shareholders' equities. If a stock has negative shareholders' equities, this implies that it has a larger amount of long term liabilities than fixed assets. A possible explanation on the high Cur may be due to the stocks having large amount of inventories or large amount of accounts receivable, which are signs of the companies in trouble. These stocks also have a large drop in their EBITG, thus this is a poor co-cluster that should be avoided. Figure 10(c) shows the price performance for these stocks in year 2002. We can see that all of their prices performed worse than the S & P 500 index, thus confirming that our model has correctly mined a poor co-cluster.



(a) (b)

Fig. 11. 470 stocks with 12 financial ratios as the input for the SOM. (a) U-matrix of an SOM. (b) The SOM where its neurons are labeled with the name of the stocks. Quantization error of SOM: 1.053. Topographic error of SOM: 0.03.



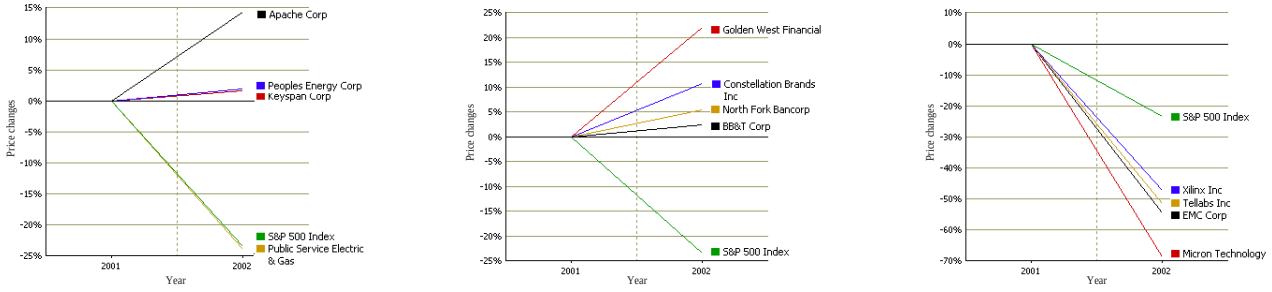
(a) (b)

Fig. 12. 139 stocks with 4 dimensions are obtained by using maximal quasi-bicliques as input vectors and dimensions selection, and they are used to train an SOM. (a) U-matrix of the SOM. (b) SOM in which its neurons are labeled with the stocks. Quantization error of SOM: 0.433. Topographic error of SOM: 0.022.

- Co-cluster 4. As this co-cluster has a high DE and negative NIG and EBITG, we consider it to be a poor co-cluster. The high DE can be attributed to the stocks having large amount of long term liabilities, as their Cur is normal. Thus, these stocks may be risky investment.

F. Using Maximal Quasi-bicliques as Input Vectors and Dimensions Selection for SOM

We explored the potential of using maximal quasi-bicliques to select stocks and financial ratios as input to SOM, for the purpose of improving the quality of SOM. We prepared a transaction dataset using the 3,528 maximal quasi-bicliques obtained from the stocks and financial ratios case study, under setting $ms = 4, \epsilon = 1$. In a transaction, the items are the



(a) Three out of the four stocks in Co-cluster 1 performed better than the S & P 500 index.

(b) All the stocks in Co-cluster 2 performed better than the S & P 500 index.

(c) All the stocks in Co-cluster 3 performed worse than the S & P 500 index.

Fig. 10. Price performances of stocks in co-clusters for the year 2002, in comparison with the S & P 500 index.

TABLE VI

NO. OF MAXIMAL QUASI-BICLIQUES/BICLIQUES MINED FROM YEAST PROTEIN-PROTEIN INTERACTION DATASET AND THEIR SIGNIFICANCE.

basic results			Group validation		pair validation	
ms	ϵ	pairs	Covered domains	Validated groups (rate)	iPfam pairs	Interdom pairs
11	0	53	386	92 (86.79%)	0	5
	1	7251	1657	12423 (85.66%)	128	420
12	0	4	24	7 (87.50%)	0	0
	1	1381	1509	2353 (85.19%)	28	115
13	1	79	318	104 (65.82%)	0	0
14	2	1150	1164	1961 (85.26%)	22	67
15	2	13	118	25 (96.15%)	0	0
16	3	12	87	17 (70.83%)	0	0
17	4	15	86	19 (63.33%)	0	0

financial ratios of a maximal quasi-biclique and the maximal quasi-biclique is the transaction identifier. Thus, we have a total of 3,528 transactions. A closed itemset mining algorithm LCM3 [41] was applied on the transaction dataset with minimum support 500 to obtain a set of frequent closed itemsets. Each closed itemset is a set of financial ratios. We took the closed itemset which has the highest number of occurrences as the selected dimensions for SOM, which corresponds to the set of financial ratios $\{Cur, DE, PB, EBITG\}$. We used 139 distinct stocks which are in the maximal quasi-bicliques that contain the financial ratios set $\{Cur, DE, PB, EBITG\}$ as the input vectors of SOM.

Figure 12(a) shows the U-matrix of the SOM and Figure 12(b) shows the SOM labeled with the stocks. This SOM was constructed using SOM Toolbox 2.0 [42] in Matlab 7.0 [43] environment. Figure 11 shows the SOM based on the original input dataset of 12 financial ratios of the 470 stocks from S & P 500. We can see that more distinct clusters are formed in Figure 10 than in Figure 11, and the quantization error and topographic error of SOM in Figure 10 are 0.433 and 0.022, which are better than 1.053 and 0.03 of SOM in Figure 11. Thus, using maximal quasi-bicliques for input vectors and dimensions selection can be useful for improving the quality of SOM.

G. Mining protein networks

We mined both maximal quasi-biclique subgraphs and maximal biclique subgraphs from the yeast ppi dataset. The aim of this experiment is to study if using maximal quasi-bicliques leads to more significant discoveries in protein networks than using maximal bicliques.

The maximal quasi-biclique subgraphs were mined using $MQBminer$ and the maximal biclique subgraphs were mined using the method in [1]. Table VI presents the number of maximal quasi-biclique subgraphs and maximal biclique subgraphs mined from the yeast ppi dataset, by varying the error tolerance threshold ϵ while maintaining a constant minimum support ms. The third column $pairs$ shows the number of maximal quasi-biclique/biclique subgraphs mined at a given ms and ϵ. The results with $\epsilon = 0$ were obtained with maximal bicliques, whereas the others were obtained with maximal quasi-bicliques. For $ms \geq 13$, no maximal biclique subgraphs were found but we are able to mine maximal quasi-biclique subgraphs by increasing ϵ. This demonstrates the strength of maximal quasi-bicliques, as large interacting pairs of protein groups can be obtained by relaxing the all-to-all relation of maximal biclique.

As mentioned in Section I, a maximal quasi-biclique/biclique subgraph represents a pair of protein groups. To validate if these discovered pairs of protein groups are significant, we use the validation techniques [1] – *Group validation (Covered domains, Validated groups)* and *Pair validation*. Details of these validation techniques are in [1].

Group validation checks if each protein group in a pair of protein groups can be mapped to domains in the domain databases. *Covered domains* indicates the number of domains in the domain databases which protein groups can be mapped to, and *Validated groups* indicates the number of protein groups that can be mapped to domains in the domain databases. At $ms = 11$ and 12, the number of *Covered domains* obtained by using maximal quasi-bicliques is 4.5 and 62.9 times more than the *Covered domains* obtained by using maximal bicliques. Similarly, at $ms = 11$ and 12, we are also able to obtain 135 and 336 times more *Validated groups* by using maximal quasi-bicliques, than by using maximal bicliques. The *Validated groups* rate in the fifth column indicates that a high percentage ($> 80\%$) of protein groups mined by maximal quasi-bicliques can be mapped to domains in the domain database.

Pair validation checks if pairs of protein groups can be mapped to pairs of domains. At $ms = 11$ and 12, by using

maximal bicliques, we can only find 5 pairs of protein groups that can be mapped to pairs of domains and they are only found in the Interdom database. By using maximal quasi-bicliques, we can map 691 pairs of protein groups to pairs of domains in both domain-domain interaction databases, iPfam and Interdom. Thus, by using maximal quasi-bicliques, we are able to discover more relations between pairs of protein groups and pairs of domains.

VI. Conclusion

We proposed maximal quasi-bicliques to overcome the weaknesses of maximal bicliques. Maximal quasi-bicliques can tolerate certain degrees of erroneous and missing data that are common in real world graphs, and the strictness of the connections between the two vertex sets forming a maximal quasi-biclique can be controlled. Our error tolerant definition of maximal quasi-bicliques is symmetrical and balanced, thus maximal quasi-bicliques do not have the problem of skewed distribution of missing edges, which is faced by prior quasi-bicliques. We developed an algorithm $MQBminer$, which mines the complete set of maximal quasi-bicliques from both bipartite and non-bipartite graphs. We also proposed to use the hierarchical clustering algorithm with a new scoring method iir for the discretization of continuous data. iir has been shown to be robust against outliers.

We showed that maximal quasi-bicliques are more robust than prior quasi-biclique models in recovering maximal bicliques from noisy graphs, and also show that the running time of $MQBminer$ is linear to the number of maximal quasi-bicliques mined. To demonstrate the versatility and effectiveness of maximal quasi-bicliques, we used them to solve a financial problem and a biology problem.

There are areas that need to be improved, which we leave as our future work. First, the error tolerance of maximal quasi-bicliques is absolute based, and having a percentage based error tolerance may be a more natural constraint, as the error tolerance is with respect to the size of the quasi-bicliques. Thus, we plan to develop an algorithm that mines maximal quasi-bicliques with percentage based error tolerance, but the absence of anti-monotone property in them makes this a difficult problem. Second, $MQBminer$ is not suitable for very large and dense graphs, as we have shown that its running time complexity is linear to the number of outputs, which can be in exponential. Hence, we plan to develop a heuristic based algorithm for mining maximal quasi-bicliques from large and dense graphs.

References

[1] H. Li, J. Li, and L. Wong, "Discovery motif pairs at interaction sites from protein sequences on a proteome-wide scale." in *Bioinformatics 22(8):989-996*, 2006.

[2] T. Murata, "Discovery of user communities from web audience measurement data." in *WI*, 2004, pp. 673–676.

[3] W. Peng, C. Ding, T. Li, and T. Sun, "Finding hotspots in document collection," in *ICTAI*, 2007.

[4] J. Li, G. Liu, H. Li, and L. Wong, "Maximal biclique subgraphs and closed pattern pairs of the adjacency matrix: A one-to-one correspondence and mining algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 12, pp. 1625–1637, 2007.

[5] G. Alexe, S. Alexe, Y. Crama, S. Foldes, P. L. Hammer, and B. Simeone, "Consensus algorithms for the generation of all maximal bicliques." *Discrete Applied Mathematics*, vol. 145, no. 1, pp. 11–21, 2004.

[6] J. Abello, M. G. C. Resende, and S. Sudarsky, "Massive quasi-clique detection," in *LATIN*, 2002, pp. 598–612.

[7] D. Bu and *et al.*, "Topological structure analysis of the protein-protein interaction network in budding yeast." in *Nucleic Acids Research 31(9):2443-2450*, 2003.

[8] N. Mishra, D. Ron, and R. Swaminathan, "A new conceptual clustering framework," *Mach. Learn.*, vol. 56, no. 1-3, pp. 115–151, 2005.

[9] C. Yan, J. G. Burleigh, and O. Eulenstein, "Identifying optimal incomplete phylogenetic data sets from sequence databases." in *Molecular Phylogenetics and Evolution 35(2005):528-535*, 2005.

[10] K. Sim, J. Li, V. Gopalkrishnan, and G. Liu, "Mining maximal quasi-bicliques to co-cluster stocks and financial ratios for value investor." in *ICDM*, 2006, pp. 1059–1063.

[11] B. Graham and D. Dodd, *Security Analysis*. McGraw-Hill Professional, 1934.

[12] P. Lynch and J. Rothchild, *One Up on Wall Street: How to Use What You Already Know to Make Money in the Market*. Simon & Schuster, 2000.

[13] B. Martín-del-Brío and C. Serrano-Cinca, "Self-organizing neural networks for the analysis and representation of data: Some financial cases." *Neural Computing & Applications.*, vol. 1, no. 3, pp. 193–206, 1993.

[14] T. Eklund, B. Back, H. Vanharanta, and A. Visa, "Assessing the feasibility of self-organizing maps for data mining financial information." in *ECIS*, 2002, pp. 528–537.

[15] C. Magnusson, A. Arppe, T. Eklund, A. Kloptchenko, B. Back, A. Visa, and H. Vanharanta, "Combining collocational networks and self-organizing maps in analyzing quarterly reports." *Information & Management.*, vol. 42, no. 4, pp. 561–574, 2005.

[16] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 90–105, 2004.

[17] Y. Cheng and G. M. Church, "Biclustering of expression data," in *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 2000, pp. 93–103.

[18] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *KDD*, 2001, pp. 269–274.

[19] A. H. Y. Tong and *et al.*, "A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules." *Science*, no. 295, pp. 321–324, 2002.

[20] T. Chiang, D. Scholtens, D. Sarkar, R. Gentleman, and W. Huber, "Coverage and error models of protein-protein interaction data by directed graph analysis." in *Genome Biology 8(9)*, 2007.

[21] J. Li, H. Li, D. Soh, and L. Wong, "A correspondence between maximal complete bipartite subgraphs and closed patterns." in *PKDD*, 2005, pp. 146–156.

[22] J. Li, K. Sim, G. Liu, and L. Wong, "Maximal quasi-bicliques with balanced noise tolerance: Concepts and co-clustering applications," in *SDM*, 2008, pp. 72–83.

[23] J. Pei, D. Jiang, and A. Zhang, "On mining cross-graph quasi-cliques," in *KDD '05*, 2005, pp. 228–238.

[24] C. Yang, U. Fayyad, and P. S. Bradley, "Efficient discovery of error-tolerant frequent itemsets in high dimensions," in *KDD*, 2001, pp. 194–203.

[25] J. Pei, A. K. H. Tung, and J. Han, "Fault-tolerant frequent pattern mining: Problems and challenges." in *DMKD*, 2001.

[26] J. Liu, S. Paulsen, X. Sun, W. Wang, A. B. Nobel, and J. Prins, "Mining approximate frequent itemsets in the presence of noise: Algorithm and analysis," in *SDM*, 2006.

[27] J. Besson, C. Robardet, and J. F. Boulicaut, "Mining a new fault-tolerant pattern type as an alternative to formal concept discovery," in *ICCS '06*, 2006, pp. 144–157.

[28] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Networks*, vol. 11, no. 3, p. 586, 2000.

[29] S. Wu and T. W. S. Chow, "Self-organizing-map based clustering using a local clustering validity index," *Neural Process. Lett.*, vol. 17, no. 3, pp. 253–271, 2003.

[30] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[31] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: an introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[32] M. Halkidi and M. Vazirgiannis, "Clustering validity assessment using multi representatives," in *SETN*, 2002.

[33] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison-Wesley, 2005.

[34] Y. Loewenstein, E. Portugaly, M. Fromer, and M. Linial, "Efficient algorithms for accurate hierarchical clustering of huge datasets: tackling the entire protein space," *Bioinformatics*, vol. 24, no. 13, pp. 41–49, 2008.
[35] "Standard and Poors," http://www.standardandpoors.com.
[36] "Compustat," http://www.compustat.com.
[37] "Database of interacting proteins," http://dip.doe-mbi.ucla.edu.
[38] "2^{nd} dimacs challenge benchmarks," ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique/.
[39] R. Gupta, G. Fang, B. Field, M. Steinbach, and V. Kumar, "Quantitative evaluation of approximate frequent pattern mining algorithms," in *KDD*, 2008.
[40] "MSN Money," http://moneycentral.msn.com.
[41] T. Uno, M. Kiyomi, and H. Arimura, "LCM ver.3: Collaboration of array, bitmap and prefix tree for frequent itemset mining," in *OSDM 2005, in conjunction with KDD*, 2005.
[42] "SOM Toolbox," http://www.cis.hut.fi/projects/somtoolbox.
[43] "MATLAB 7.0," http://www.mathworks.com/products/matlab.