A Game Theoretic Framework for Heterogeneous Information Network Clustering

Faris Alqadah*
Johns Hopkins University
733 North Broadway
Baltimore, MD 21205
faris.alqadah@jhu.edu

Raj Bhatnagar
University of Cincinnati
814 Rhodes Hall
Cincinnati, Ohio 45221
raj.bhatnagar@uc.edu

ABSTRACT

Heterogeneous information networks are pervasive in applications ranging from bioinformatics to e-commerce. As a result, unsupervised learning and clustering methods pertaining to such networks have gained significant attention recently. Nodes in a heterogeneous information network are regarded as objects derived from distinct domains such as 'authors' and 'papers'. In many cases, feature sets characterizing the objects are not available, hence, clustering of the objects depends solely on the links and relationships amongst objects. Although several previous studies have addressed information network clustering, shortcomings remain. First, the definition of what constitutes an information network cluster varies drastically from study to study. Second, previous algorithms have generally focused on non-overlapping clusters, while many algorithms are also limited to specific network topologies.

In this paper we introduce a game theoretic framework (GHIN) for defining and mining clusters in heterogeneous information networks. The clustering problem is modeled as a game wherein each domain represents a player and clusters are defined as the Nash equilibrium points of the game. Adopting the abstraction of Nash equilibrium points as clusters allows for flexible definition of reward functions that characterize clusters without any modification to the underlying algorithm. We prove that well-established definitions of clusters in 2-domain information networks such as formal concepts, maximal bi-cliques, and noisy binary tiles can always be represented as Nash equilibrium points. Moreover, experimental results employing a variety of reward functions and several real world information networks illustrate that the GHIN framework produces more accurate and informative clusters than the recently proposed NetClus and state of the art MDC algorithms.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

*Portions of work were completed while author was attending the University of Cincinnati

KDD'11, August 21–24, 2011, San Diego, California, USA.

General Terms

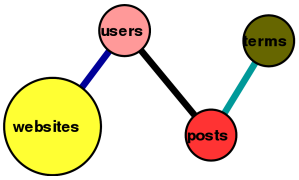
Algorithms

1. INTRODUCTION

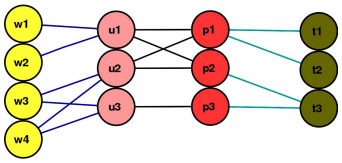
Heterogeneous information networks (HINs) are pervasive in applications ranging from bioinformatics to e-commerce. The nodes of a HIN are regarded as objects derived from distinct domains while the topology of the network is defined by how the domains are related. In many cases, feature sets characterizing the objects are not available, hence, clustering of the objects depends solely on the links and relationships amongst objects. For example, consider the domains of *users*, *websites*, *posts*, and *terms* as illustrated in figure 1. The topology of the network is defined by edges between the domains (figure 1(a)); each edge represents a relation table relating the objects between the pair of domains. In the example, the $websites\_users$ relation captures clicking patterns of users, the $users\_posts$ table specifies which users made postings on different blogs, and the $posts\_terms$ table is a document-terms dataset relating terms to postings. Clustering the nodes of a HIN advances knowledge discovery in two manners. First, hidden associations between objects from differing domains are unveiled; this leads to a better understanding of the hidden structure of the entire network. Second, local clusters within a domain are sharpened and put into greater context, resulting in more accurate local clustering. Consider the HIN in figure 1(b) once again. While no information is available relating websites to blog posts, the fact that $u1$ clusters with both $\{p1, p2\}$ and $\{w1, w2\}$ may suggest a hidden similarity between $\{p1, p2\}$ and $\{w1, w2\}$. Clustering *users* exclusively on clicking patterns leads to the cluster $\{u2, u3\}$; however, when considering the $users\_posts$ relation a strong case can also be made for the clustering $\{u1, u2\}$. Consequently, beyond inference of hidden links between objects, information network clustering revises and puts local clustering of objects within a domain into context. In addition to illustrating the benefits of information network clustering, the above example exposes the need to consider overlapping clusters. How does an algorithm decide between the $\{u1, u2\}$ and $\{u2, u3\}$ clusterings? Considering all the different relations, strong cases can be made for both groupings. This scenario occurs frequently in real world applications such as bioinformatics and text mining where a gene may encompass several functional groups or a document may relate to multiple categories such as "Politics" and "Religion".

1.1 Related Work

Few works have approached the task of clustering data in game theoretic terms. Most notably in [8] the authors propose a game theoretic approach to hypergraph clustering. This work is simi-



(a) Topology of the network



(b) The Full HIN

Figure 1: A tree-shaped heterogeneous information network with four domains

lar in philosophy to the GHIN framework proposed here; both approaches define clusters in terms of Nash equilibrium points and cultivate algorithms to enumerate these points. The works differ in the type of clustering problem addressed. The hypergraph clustering problem consists of homogeneous objects paired with similarity scores between objects that are known a-priori (although these similarities need not be pairwise and can be of higher order). As a result, the hypergraph clustering game and subsequent rewards are naturally defined only in terms of the given object similarities. In the HIN-clustering problem only relational information between heterogeneous objects is available; hence, in addition to determining a framework to define HIN clusters, appropriate relational reward functions must also be developed. Moreover, although the authors of [8] point out that a game-theoretic framework has the potential to mine overlapping clusters, the final adapted algorithm does not.

HIN-clustering has been addressed in the literature as "multi-way" [18, 17, 5, 6], "information-network" [26, 27], and "relational clustering" [7, 29, 30]. Additionally, bi-clustering [9, 19] algorithms may be viewed as information network clustering algorithms specified for single edge information networks. Nonetheless, the definition of an information network cluster varies from study to study. Multi-way clustering algorithms generalize bi-clustering approaches, hence adopting and generalizing the definition of the respective bi-clustering approach. The definition of a bi-cluster has varied significantly and includes: vertex cuts [12], maximal cliques [16, 28, 32], maximal sub-matrices that minimize variance [10, 15, 11], and maximal sub-matrices that maximize mutual information with respect to a pre-specified number of clusters [13]. Relational and information network clustering approaches utilize similarity measures such as PageRank, SimRank and other ranking measures defined over the information network to perform clustering [30, 26]. The premise of these similarity measures is the recursive definition that similarity between two objects depends on the similarities between the objects linked with them. Recently, NetClus was introduced [27] utilizing similarity measures and the assumption that "every net-cluster is corresponding to a generative model, according to which generative probabilities of every target object in each cluster can be calculated".

Additionally, the majority of algorithms in the literature do not allow for overlapping clusters, require a pre-specified number of clusters, and are limited by the topology of the information network. For example, multi-way clustering algorithms operate on any given topology, but are limited to non-overlapping clusters and require a pre-specified number of clusters. NetClus and other ranking-based approaches are only suited for star shaped networks and still require the number of clusters as a parameter.

1.2 Contributions

The principal contributions of this paper are two fold. First, the GHIN framework, based on game theory, is introduced as a general scheme for defining and mining clusters in HINs. The abstraction of a Nash equilibrium point in a game is employed to define a HIN-cluster while mining the clusters consists of enumerating the Nash equilibrium points. The premise of the framework is that a HIN-cluster constitutes an equilibrium among several possible competing local clusterings of objects in each domain. This idea was inspired by the observation in [24] (expanded upon in [1] to star shaped information networks) that a subspace cluster constitutes a trade-off between the number of data points and attributes admitted. The more objects a subspace cluster encompasses the fewer attributes it will admit and vice-versa. This observation naturally extends to a relational setting. Consider the HIN in figure 1 once again. The more $users$ included in a cluster naturally implies fewer $websites$ jointly visited by those users and fewer $posts$ collectively blogged on. Accordingly, a trade-off exists among objects included in a HIN-cluster based on the topology of the network. Moreover, as the discussion in the previous section depicted, different relations may influence the formation of clusters within a domain in a seemingly contradictory manner. Hence, a cluster that incorporates data from the entire information network should be viewed as an equilibrium point between competing clustering influences. This notion is captured precisely as the Nash equilibrium solution concept of a game involving two or more players. A game is in Nash equilibrium if each player has chosen a strategy and no player has anything to gain by changing only his own strategy unilaterally. In terms of HINs, we define a cluster as a collection of nodes from each domain in which the quality criterion or reward function in each domain cannot be improved by unilaterally changing the clustering of that domain.

The second primary contribution of the paper is the development of specific reward functions in conjunction with GHIN. These reward functions yield effective HIN-clustering algorithms as evidenced by an empirical study. In the sequel we formally introduce the GHIN framework by modeling the HIN-clustering problem as a game and defining the clusters as the Nash equilibrium points of that game. Section 3 establishes the framework for a single edge information network (bi-clustering) and shows that this approach encompasses well-established bi-clustering frameworks. In section 4 the GHIN framework is defined for a general tree-shaped HIN while section 5 proposes two reward functions to be used in conjunction with GHIN. Section 6 displays experimental results and section 7 offers concluding remarks and possible extensions.

2. PRELIMINARIES

We formulate the HIN-clustering problem in the vocabulary of Formal Concept Analysis (FCA). The field of FCA serves as a theoretical basis for association rule analysis, closed itemset mining [31], and bi-clustering [16][2]. Next, the basic terminology and conceptions of game theory are presented and related to HIN-clustering in section 3.

2.1 Heterogeneous Information Networks and FCA

A **context** $\mathbb{K}_{ij} = (G_i, G_j, I_{ij})$ consists of two sets G_i, G_j and a relation I_{ij} between them. We refer to the sets G_i and G_j as

domains. We refer to the elements of each domain G_i, G_j as **objects** and denote them as $g_i^1, \ldots, g_i^{|G_i|}$. A context may be depicted as a $|G_i| \times |G_j|$ binary matrix, denoted as $mat(\mathbb{K}_{ij})$, where $mat(\mathbb{K}_{ij})_{mn} = 1$ if $g_i^m I_{ij} g_j^n$ and 0 otherwise. Moreover, $\mathbb{K}_{ij}$ may also be viewed as a bipartite graph, denoted $grph(\mathbb{K}_{ij})$, with vertex set $G_i \cup G_j$, and edge set I_{ij}. Therefore, $mat(\mathbb{K}_{ij})$ is the adjacency matrix of $grph(\mathbb{K}_{ij})$. A **heterogeneous information network** (HIN) is a graph $\mathbb{G}_n = (\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ is a set of domains $\{G_1, \ldots, G_n\}$ and $(G_i, G_j) \in \mathbf{E}$ iff $\exists \mathbb{K}_{ij}$. A **subspace** of an information network $\mathbb{G}_n$ is an n-tuple of **object-sets** $(A_1 \subseteq G_1, \ldots, A_n \subseteq G_n)$. In general, the HIN clustering problem is to identify a set of subspaces that maximizes a quality criterion $\mathbf{Q}$ subject to a set of constraints $\mathbf{C}$. Clustering of a single edge HIN amounts to clustering in a singleton context; in this case FCA has been well established as a theoretical base. For object-set A_i define

$$\psi^j(A_i) = \begin{cases} \{g_j \in G_j | g_j I_{ij} g_i \quad \forall g_i \in A_i\} & \text{if } (G_i, G_j) \in \mathbf{E}, \\ \emptyset & \text{otherwise.} \end{cases}$$

In words, $\psi^j(A_i)$ identifies the objects of G_j common to the objects in A_i. If $(G_i, G_j) \in \mathbf{E}$ then generally $\psi^j(A_i)$ maybe computed as $\bigcap_{g \in A_i} \psi^j(g)$.

DEFINITION 1. *A **concept** of the context (G_i, G_j, I_{ij}) is a subspace (A_i, A_j) such that $\psi^j(A_i) = A_j$ and $\psi^i(A_j) = A_i$.*

The above definition can be shown to yield two closure systems on G_i and G_j which are dually isomorphic to each other [14]. For any object-set $A_i \subseteq G_i$, then$(\psi^i(\psi^j(A_i)), \psi^j(A_i))$ is always a concept. A **semi-concept** of a $\mathbb{K}_{ij}$ is a subspace (A_i, A_j) such that $\psi^j(A_i) = A_j$ or $\psi^i(A_j) = A_i$ and constitutes a relaxation on the stricter definition of a concept. Utilizing the binary matrix representation, a concept can be represented by a *maximal* sub-matrix of 1's under suitable permutations of the rows and columns. The term maximal indicates that no row or column can be added to the submatrix without the introduction of at least one zero. Concepts can also be thought of as the maximal bi-cliques of $grph(\mathbb{K}_{ij})$ [16]. This is important as it was illustrated in [16] that the maximal bi-cliques of a dataset correspond exactly to the bi-clusters (subspace clusters, co-clusters, projected clusters, closed patterns) of a binary dataset; thus, the bi-clustering problem is modeled by FCA. A natural approach to solving the HIN-clustering problem is to extend FCA-conceptions to the general case of a HIN. In section 3 it is proven that Nash-equilibrium points for a suitably defined game encapsulate the FCA definition of a bi-cluster.

2.2 Game Theory

A game consists of a set of players, positions, moves, and a reward function. The moves are defined by a set of rules and dictate how players may move between different positions. The situation at the start of the game is called the **initial** position. At each position, the rules indicate which player (or players) make a move from that position and what the allowable moves are. For every position p, there must be a sequence of moves from the initial position to p. Some positions are designated as **terminal** positions, and no moves are allowed from such a position, hence, the game ends when a terminal position is reached. A play of the game consists of a sequence of moves starting at the initial position and ending at a terminal position. Every terminal position determines a reward or pay-off to each of the players. A central notion of game theory is **strategy**. A strategy for a player is a specification that tells the player what to do in every situation that might arise during a game. Notice that once a strategy is chosen the player's moves are completely determined.

	Player 2 chooses 0	Player 2 chooses 1	Player 2 chooses 2
Player 1 chooses 0	(0,0)	(1,0)	(2,-2)
Player 1 chooses 1	(0,1)	(1,1)	(3,-2)
Player 1 chooses 2	(-2,2)	(-2,3)	(2,2)

Figure 2: Game displayed in normal form, with Nash equilibrium points highlighted

DEFINITION 2. *A finite, n-player, normal form game, $\mathcal{G}$, is a triple $\langle N, (M_i), (r_i) \rangle$ where*

- *$N = \{1, \ldots, n\}$ is the set of players*
- *$M_i = \{m_i^1, \ldots, m_i^{l_i}\}$ is the set of moves available to player i and l_i is the number of available moves for that player.*
- *$r_i : M_1 \times \cdots \times M_n \to \mathfrak{R}$ is the reward function for each player i. It maps a profile of moves to a value.*

Each player i selects a strategy from the set of all available strategies $\mathbb{P}_i = \{p_i : M_i \to [0, 1]\}$. The primary solution concept for a normal form game is that of a **Nash equilibrium**: a strategy profile in which no player has an incentive to unilaterally deviate [21, 20]. In other words, if each player has chosen a strategy and no player can benefit by unilaterally changing his or her strategy (assuming other players keep theirs unchanged) then the current set of strategy choices and the corresponding payoffs constitute a Nash equilibrium.

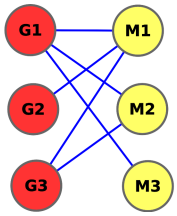
DEFINITION 3. *A strategy profile $p^* \in \mathbb{P}$ is a **Nash equilibrium** if :*

$$\begin{aligned} \forall i \in N, p_i \in \mathbb{P}_i \quad &: \\ r_i(p_1^*, \ldots, p_{i-1}^*, p_i, \ldots, p_n^*) \quad &\leq \quad r_i(p_1^*, \ldots, p_n^*) \end{aligned}$$

A fundamental theorem of game theory is that every finite game in which the players have perfect information has a Nash equilibrium [21, 20]. Normal form games maybe represented by an n-dimensional array of n-vectors by tabulating the function $r(p_1, \ldots, p_n) = (r_1(p_1, \ldots, p_n), \ldots, r_n(p_1, \ldots, p_n))$. In the case of a two-player game, this representation reduces to a matrix whose elements are pairs of real numbers. Figure 2 illustrates the matrix representation of a two-player normal form game in which two players simultaneously select an integer between 0 and 2. If the two players select the same integer the reward is the value of the selected integer. If one player selects a larger number then they give up that many points to the other player and the player with the smaller number is rewarded the value of their own number. The Nash equilibrium points of the game are highlighted in yellow.

3. THE BI-CLUSTERING GAME

The bi-clustering problem can be viewed as a game. Consider two party planners, P_1 and P_2, whose goal is to plan a party by inviting guests on their client lists. Each party planner is responsible for disjoint sets of clients, G_1 and G_2, and can only send out invitations to clients in their set. The party planners receive compensation based on the overall satisfaction of their clients. Party-goers from each set of clients only interact with party-goers of the other set and satisfaction of each individual party-goer is based on the amount of interaction at the party. Additionally, if party goers encounter guests that they do not care for then their satisfaction is negatively impacted. Both P_1 and P_2 are good at their jobs and know whose company each client enjoys and whose company they



(a) Client Preferences

	$M1$	$M1, M2$	$M1, M2, M3$	$M1, M3$	$M2$	$M2, M3$	$M3$
$G1$	(1,1)	(1,2)	(1,3)	(1,2)	(1,1)	(1,2)	(1,1)
$G1, G2$	(2,1)	(-1,-1)	(-2,-3)	(-1,-1)	(-4,-2)	(-4,-4)	(-4,-2)
$G1, G2, G3$	(3,1)	(0,0)	(-3,-3)	(-3,-2)	(-3,-1)	(-6,-4)	(-9,-3)
$G1, G3$	(2,1)	(2,2)	(0,0)	(-1,-1)	(2,1)	(-1,-1)	(-4,-2)
$G2$	(1,1)	(-2,-4)	(-3,-9)	(-2,-4)	(-5,-5)	(-5,-10)	(-5,-5)
$G2, G3$	(2,1)	(-1,-1)	(-4,-6)	(-4,-4)	(-4,-2)	(-7,-7)	(-10,-5)
$G3$	(1,1)	(1,2)	(-1,-3)	(-2,-4)	(1,1)	(-2,-4)	(-5,-5)

(b) Normal form of the game, w=5

Figure 3: The bi-clustering (party planning) game

don't. Moreover, we assume that P_1 and P_2 do not cooperate but are privy to the guest list of the other at any given point. Thus, the game is for P_1 and P_2 to create guest lists for each party such that they maximize their compensation. It is randomly selected who begins and players alternate selecting individual clients to add or remove from their invitation lists. The game ends when both party planners have completed creating their lists or after a finite time interval. In the subsequent sections we will refer to the game described above as the party planners game or bi-clustering game interchangeably.

Clearly, all subsets $A_1 \subseteq G_1$ and $A_2 \subseteq G_2$ may constitute the terminal nodes of the bi-clustering game. Let the context $\mathbb{K}_{ij} = (G_i, G_j, I_{ij})$ represent the preferences of clients, then, any subspace (A_1, A_2) represents a party and can also be thought of as a strategy profile (p_1, p_2). In order to maximize compensation, each party planner certainly must attempt to invite clients with similar preferences in relation to the clients of the opposing party planner. The bi-clustering game is a finite game in which players have perfect information, therefore, Nash equilibrium points do exist and constitute the primary solution to the game.

DEFINITION 4. *Given $\mathbb{K}_{12} = (G_1, G_2, I_{12})$, then the bi-clustering game is defined as the normal form game*

$$\mathcal{G}_{bicluster} = \langle \{1,2\}, \{\mathcal{P}(G_1), \mathcal{P}(G_2)\}, \{r_1, r_2\} \rangle \qquad (1)$$

where $\mathcal{P}$ denotes the power-set, and r_i is a user selected reward function measuring the subsequent pay-off to the party planner for a given party.

Definition 4 represents the bi-clustering game in its most general form. Additional rules may explicitly be appended to the game (such as minimum bounds on the size of parties) or implicitly enforced through the appropriate selection of reward functions.

DEFINITION 5. *Given $\mathbb{K}_{12} = (G_1, G_2, I_{12})$, then the bi-clusters of $\mathbb{K}$ are defined as the Nash equilibrium pairs (A_1, A_2) of $\mathcal{G}_{bicluster}$.*

Below, it is illustrated that with the use of an intuitive and simple reward function the definition of a bi-cluster in a context is encapsulated by definitions 4 and 5.

3.1 Concepts as Nash Equilibrium Points

As specified by the party-planning game, the reward of P_1 and P_2 is a function of the satisfaction of party-goers. Intuitively, the more friends a party-goer encounters, the greater their satisfaction. On the other hand, as party-goers encounter people they do not care for their satisfaction is negatively impacted. Hence, for a single party-goer $g_1 \in G_1$ (dually $g_2 \in G_2$) attending party (A_1, A_2), define the satisfaction of g_1 as

$$sat_1(g_1, A_2) = \frac{|\psi^2(g_1) \cap A_2| - w * |A_2 \setminus \psi^2(g_1)|}{|A_2|} \qquad (2)$$

where w represents the weight of a non interaction. Once again we assume P_1 and P_2 are good at their jobs and the value of w is known to them. The reward of each party planner is then the average satisfaction of all clients magnified by the number of clients:

$$r_i^{sat}(A_i, A_j) = \sum_{g_i \in A_i} sat_i(g_i, A_j) \qquad (3)$$

Letting $w = 0$, then r_i^{sat} is exactly the sum of node degrees of all g_i in the subspace A_j normalized by $|A_j|$; hence, node degree is a special case of this reward function. Figure 3 illustrates a sample instance of the bi-clustering game utilizing the r_i^{sat} reward function. Specifically, figure 3(a) represents the context as a bipartite graph, while 3(b) exhibits the normal-form of the game played with $w = 5$ along with Nash equilibrium points. Notice that in this case the Nash equilibrium points correspond exactly to the maximal bi-cliques or bi-clusters of the network. In general, as w grows the fewer negative interactions party planners will tolerate when attempting to maximize r_i^{sat}.

THEOREM 1. *For any instance of $\mathcal{G}_{bicluster}$ in which r_i^{sat} is the selected reward function there exists w^* such that: $\forall w \geq w^*$ if (A_1^*, A_2^*) is a concept of $\mathbb{K} = (G_1, G_2, I_{12})$ then (A_1^*, A_2^*) is a Nash equilibrium point of $\mathcal{G}_{bicluster}$.*

PROOF. *Let (A_1^*, A_2^*) be a concept and*

$$w = \max\{\max_{g_1 \in G_1} \{|\psi^2(g_1)|\}, \max_{g_2 \in G_2} \{|\psi^1(g_2)|\}\}$$

Assume that (A_1^*, A_2^*) is not a Nash equilibrium; then w.l.o.g there must exist (A_1, A_2) such that $r_1^{sat}(A_1, A_2^*) > r_1^{sat}(A_1^*, A_2^*)$. Two cases arise:

1. *(A_1, A_2^*) is a semi-concept. Then*

$$\begin{aligned} \sum_{g_1 \in A_1} sat_1(g_1, A_2^*) &> \sum_{g_1^* \in A_1^*} sat_1(g_1^*, A_2^*) \\ \frac{|A_1| * |A_2^*|}{|A_2^*|} &> \frac{|A_1^*| * |A_2^*|}{|A_2^*|} \\ |A_1| &> |A_1^*| \end{aligned}$$

By the maximality of concepts, this is a contradiction.

2. *(A_1, A_2^*) is not a semi-concept, then two sub-cases arise:*

   (a) *$A_1 \supseteq A_1^*$. In this case $|A_1| = |A_1^* \cap A_1| + |A_1 \setminus A_1^*|$. Let $d_1 = |A_1 \setminus A_1^*|$ and $d_2 = |A_1^* \cap A_1|$, then by definition of a concept*

$$\begin{aligned} |\psi^2(a_1) \cap A_2^*| &\leq |A_2^*| - 1 \\ |A_2^* \setminus \psi^2(a_1)| &\geq 1 \end{aligned}$$

for any $a_1 \in A_1 \setminus A_1^*$. Furthermore,

$$w \geq |A_2^*|$$

Hence

$$\begin{aligned} r_1^{sat}(A_1, A_2^*) &\geq r_1^{sat}(A_1^*, A_2^*) \\ \frac{d_1(|A_2^*|-1) + d_2|A_2^*| - w * d_1}{|A_2^*|} &\geq \frac{|A_1^*| * |A_2^*|}{|A_2^*|} \\ d_1(|A_2^*|-1) + d_2|A_2^*| - w * d_1 &> |A_1^*| * |A_2^*| \\ d_1(|A_2^*|-1) + d_2|A_2^*| - |A_2^*|d_1 &> |A_1^*| * |A_2^*| \\ d_2|A_2^*| - d_1 &> |A_1^*| * |A_2^*| \end{aligned}$$

which is a contracdiction by the property of set difference.

(b) $A_1 \not\supseteq A_1^*$. *Let* $d_1' = |A_1 \setminus A_1^*|$ *and* $d_2' = |A_1^* \cap A_1|$. *In this case, the argument is the same as above, with the distinction that* $d_2 > d_2'$ *and* $d_1 \leq d_1'$.

□

Theorem 1 establishes that definition 5 paired with an intuitive reward function encompasses several previous approaches to bi-clustering in binary relations. Simply varying the value of w leads to FCA [14], fault-tolerant FCA [22], and quasi-biclique [25] approaches.

4. GAME THEORETIC FRAMEWORK

The HIN-clustering problem can be viewed as a multi-player version of $\mathcal{G}_{bicluster}$.

DEFINITION 6. *Given HIN* $\mathbb{G}_n$, *then the HIN-clustering game is defined as the normal form game*

$$\mathcal{G}_{hin} = \langle \{1, \ldots, n\}, \{\mathcal{P}(G_1), \ldots, \mathcal{P}(G_n)\}, \{r_1, \ldots, r_n\} \rangle \quad (4)$$

In the HIN-clustering game, n party planners attempt to plan a party to maximize their individual rewards. Party-goers from each set of clients only interact with clients of other sets specified by $\mathbf{E} \in \mathbb{G}_n$. The multi-player game proceeds exactly as the two-player game and each planner is privy to the guest lists of all other party planners. The terminal nodes of the multi-player party planner game constitutes all subspaces $(A_1, \ldots, A_n)$ of $\mathbb{G}_n$.

DEFINITION 7. *Given a HIN,* $\mathbb{G}_n$, *then the HIN-clusters of* $\mathbb{G}_n$ *are defined as the Nash equilibrium points of* $\mathcal{G}_{hin}$.

Adopting definition 7, the problem of enumerating HIN clusters is equivalent to that of finding Nash equilibrium points in a normal form game. This problem is notorious and has been described as "the most fundamental problem" at the interface of computer science and game theory; however, recent efficient algorithms have been proposed [23]. In particular, the algorithm presented in [23] makes use of a simple search strategy and simple heuristics but has been shown to be quite effective. Thus, in general, these algorithms may be used in conjunction with definition 7 to enumerate HIN clusters with any given reward function. In this paper, however, we propose a more specialized framework tailored to the party planner game due to the fact that we have a deeper understanding of the structure of the problem.

Consider a two player normal form game, then a simple labeling technique for locating all Nash equilibria is [20]:

1. Mark all second components that are maximal among all second components in each row.
2. Mark all first components that are maximal among all first components in each column.
3. Any cell in the matrix that has both components marked is then in a state of Nash equilibrium.

Clearly, the drawback of the labeling technique is the assumption that the game matrix is pre-computed. However, it is infeasible to compute the entire game matrix for the party planners game; in the worst case this would involve enumerating at least $2^{\max(|G_1|,|G_2|)}$ cells in the game matrix. As a result, GHIN is engineered utilizing the labeling technique augmented with heuristics.

4.1 n-Clusters as Candidates

In order to condense the search space for locating a Nash equilibrium, we propose a heuristic to identify initial candidate subspaces. The premise of the argument is that any reward function, $r_i(A_1, \ldots, A_n)$, must be a function of the count and distribution of the edges or interactions in the subspace. Using the notation that $\{-i\}$ denotes the integers $1, \ldots, i-1, i+1, \ldots, n$, each party planner i wishes to maximize the edge count of domain i given the current selection of all other players $\{-i\}$. Hence, we conclude that it is advantageous to all players to start with a party in which all party-goers know each other and is maximal.

HEURISTIC 1. *Given* $\mathbb{G}_n$, *all Nash equilibrium points of* $\mathcal{G}_{hin}$ *are supersets of the* ***n-clusters*** *of* $\mathbb{G}_n$. *An* n*-cluster is a subspace* $(A_1, \ldots, A_n)$ *such that the following conditions hold:*

1. $\forall i, j$ *s.t.* $(G_i, G_j) \in \mathbf{E}$, $(A_i \subseteq G_i, A_j \subseteq G_j)$ *is a semi-concept.*
2. $\forall i$ *there does not exist* g_i *s.t. for* $(A_{-i}, A_i \cup g_i)$ *condition 1 still holds.*

The game theoretic framework for mining HIN-clusters, GHIN, is presented as algorithm 1. The framework sacrifices completeness in order to overcome the computational cost of enumerating Nash equilibrium points. On the other hand, GHIN attempts to maximize coverage of all objects by striving to form the initial n-cluster candidates from each object in the HIN. The set R is utilized to keep track of these 'seed' objects to form the initial candidates. Lines 4-5 entail generating an initial candidate n-cluster C; such subspaces may be computed utilizing the algorithm presented in [1]. Notice on line 5 that only n-clusters that contain objects in R will be formed as candidates; this step attempts to maximize the diversity of the clusters enumerated while overlapping clusters may still be formed through the refinement phase on lines 7-16. Following heuristic 1 and the labeling technique described earlier, GHIN attempts to locate a Nash-equilibrium point by holding all object-sets A_{-i} as fixed points while maximizing the reward for player i (lines 8-11). After each iteration of the *for* loop, C is updated with all the added objects for domain i. Clearly, the entire *repeat* loop is guaranteed to terminate since objects are only being added. The next refinement phase (lines 12-15) is analogous to the previous one except that GHIN attempts to remove objects whose removal increases the reward for player i. The entire refinement process iterates until no change is possible or a user-defined maximum number of iterations, max, is reached. By definition, C constitutes a Nash equilibrium if the refinement phase loop terminates prior to reaching max. Finally, in the case that C is a cluster, then R is updated by removing all objects found in C. The algorithm terminates when no more objects are left in R to form candidates with.

4.2 Computational Complexity

The running time complexity of GHIN depends on the complexity of computing r_i for a given subspace. In general, the worst

Input: $\mathbb{G}_n$, Reward functions: $(r_1, \ldots, r_n)$
Data: Maximum number of iterations to find Nash equilibrium max

begin
  $R \leftarrow \bigcup_{i=1}^{n} G_i \in \mathbf{V}$;
  repeat
    $i \leftarrow$ Randomly select i from $1 \ldots n$;
    $C = (A_1, \ldots, A_n) \leftarrow$ form n-cluster starting from G_i that only encompasses objects in R;
    $R \leftarrow R \setminus \bigcup_{i=1}^{n} A_i \in C$;
    repeat
      repeat
        for $i \leftarrow 1$ **to** n *in random order* **do**
          Add all $g_i \in G_i$ to A_i s.t. $r_i(A_i \cup g_i) > r_i(A_i)$;
      until *No change in* C ;
      repeat
        for $i \leftarrow 1$ **to** n, *in random order* **do**
          Remove all $g_i \in A_i$ from A_i s.t. $r_i(A_i \setminus g_i) > r_i(A_i)$;
      until *No change in* C ;
    until *No change in* C *or* max *reached* ;
    if *Nash equilibrium found* **then**
      Print C as a HIN-cluster ;
      $R \leftarrow R \setminus \bigcup_{i=1}^{n} A_i \in C$;
  until $|R| < 1$;
end

Algorithm 1: Game Theoretic Framework (GHIN) for HIN-clustering

case occurs when all objects in $\mathbb{G}_n$ are utilized to form candidates and the refinement phase executes max number of times. Forming a single candidate with the algorithm proposed in [1] runs in $O(|G_i|)$ time; therefore, the worst case complexity of GHIN is $O(|G_i| * max * |G_i| * n * O(r_i))$. $O(r_i)$ is the computational complexity of computing r_i for a subspace and G_i is the largest domain in $\mathbb{G}_n$. Considering max and n as small constants and assuming $O(r_i)$ is linear with respect to G_i (as is the case with the two reward functions presented next), a more realistic running-time complexity is $O(|G_i|^3)$. On the other hand, our experimental results indicate that even this running time is rarely encountered as the set R tends to die down fairly quickly (see experimental section).

5. REWARD FUNCTIONS

The advantage of GHIN is the fact that any reward function that is primarily based on edge-counts and their distribution may be utilized without any change to the algorithm. Here we present two reward functions with which we implemented GHIN.

5.1 Satisfaction

Extending the satisfaction reward function presented in section 3 to a HIN yields

$$sat_i(g_i, A_{-i}) = \sum_{A_j \subseteq G_j, (G_i, G_j) \in \mathbf{E}} sat_i(g_i, A_j)$$

$$r_i^{sat}(A_i, A_{-i}) = \sum_{g \in A_i} sat_i(g_i, A_{-i})$$

This reward function is a direct generalization of the one presented in section 3. This intuitive reward function is simple to implement and experimental results indicate that it works well in information networks that have approximately equivalent density levels in all contexts. On the other hand, if the contexts of $\mathbb{G}_n$ have significantly different density levels then r_i^{sat} is biased towards objects from those domains that have higher density due to the high node degree of such objects.

5.2 Expected Satisfaction

To address the bias of r_i^{sat} we develop r_i^{esat} in which the satisfaction of a client is based on the expected number of positive interactions and not on the pure count. Given $\mathbb{G}_n$, assume each $g_i \in G_i$ is independent. For a given subspace $(A_1, \ldots, A_n)$ the expected number of positive interactions for $g_i \in A_i$ is the number of success in a sequence of $|A_j|$ draws from a finite population of $|G_j|$ objects without replacement. Hence, the expected number of success is the expected value of a hypergeometrically distributed random variable. For object g_i and object-set A_j in subspace $(A_1, \ldots, A_j, \ldots, A_n)$, let $exp_{ij}(g_i, A_j)$ denote the expected number of positive interactions between g_i and objects in A_j and $var_{ij}(g_i, A_j)$ denote the variance:

$$exp_{ij}(g_i, A_j) = \frac{|A_j| * |\psi^j(g_i)|}{|G_j|}$$

$$var_{ij}(g_i, A_j) = \frac{|A_j| * |\psi^j(g_i)| * (|G_j| - |A_j|) * (|G_j| - |\psi^j(g_i)|)}{|G_j|^2 * (|G_j| - 1)}$$

Define satisfaction of a client g_i at a party as the difference between the actual number of positive interactions and the expected number of positive interactions; this can be captured as the z-score:

$$esat(g_i, A_j) = \frac{|\psi^j(g_i) \cap A_j| - exp_{ij}(g_i, A_j)}{\sqrt{var_{ij}(g_i, A_j)}} - w$$

$$esat(g_i, A_{-i}) = \sum_{A_j \subseteq G_j, (G_i, G_j) \in \mathbf{E}} esat(g_i, G_j)$$

$$r_i^{esat}(A_i, A_{-i}) = \sum_{g_i \in A_i} esat(g_i, A_{-i})$$

In the above formulation w is a user-defined parameter which determines how large the standardized score must be to make a positive contribution to the overall reward. The reward for party planner i is simply the sum of the expected satisfaction of each client.

5.3 Tiring Party-Goers

A common problem with overlapping clustering techniques is that the number of clusters enumerated may be quite large [2]. This problem may be overcome in the GHIN framework by incorporating an additional 'tiring' factor into any reward function r_i. The more times party-goer g_i appears in a cluster the less likely g_i is to appear in future clusters. Let $c(g_i)$ denote the number of clusters g_i has appeared in upto the current time-step, then let

$$t = f(c(g_i))$$

where

$$f : \mathbb{N} \rightarrow (0, 1]$$

and f is anti-monotonic. For example:

$$f(x) = \frac{1}{x^2}$$

$$f(x) = \frac{1}{e^x}$$

HIN name	Description	Num domains	Num classes	Total num objects
MER	Newsgroup, Middle East politics and Religion	3	2	24,783
REC	Newsgroup, recreation	3	2	26,225
SCI	Newsgroup, science	3	4	37,413
PC	Newsgroup, pc and software	3	5	35,186
PCR	Newsgroup, politics and Christianity	3	2	24,485
FOUR_AREAS	DBLP subset of database, data mining, AI, and IR papers	4	4	70,517

Figure 4: Real-world HINs

HIN	Algorithm	F_1	$F_{0.5}$	F_2
MER	GHIN expsat	**0.627051**	**0.736396**	**0.622735**
	GHIN sat	0.553790	0.649559	0.569664
	NetClus	0.3759	0.4512	0.322
	MDC	0.3661	0.4533	0.3070
REC	GHIN expsat	**0.544189**	**0.633362**	**0.508778**
	GHIN sat	0.434367	0.485025	0.451840
	NetClus	0.2784	0.2870	0.2704
	MDC	0.2845	0.2953	0.2746
SCI	GHIN expsat	**0.484068**	**0.589704**	**0.530239**
	GHIN sat	0.402306	0.481798	0.462886
	NetClus	0.2609	0.2583	0.2635
	MDC	0.2532	0.2529	0.2535
PC	GHIN expsat	**0.334827**	**0.520472**	0.302943
	GHIN sat	0.306503	0.432229	**0.345382**
	NetClus	0.2254	0.2068	0.2477
	MDC	0.2282	0.2116	0.2476
PCR	GHIN expsat	**0.640894**	**0.793399**	0.508778
	GHIN sat	0.541986	0.574588	**0.530971**
	NetClus	0.3642	0.4396	0.3109
	MDC	0.3440	0.4268	0.2810
FOUR_AREAS	GHIN expsat	**0.623117**	**0.598877**	**0.650079**
	GHIN sat	0.5315	0.506687	0.5588
	NetClus	0.3612	0.36655	0.3560
	MDC	0.5085	0.5162	0.5010

(a) B^3 F-scores

Algorithm	Class	C1	C2	C3	C4
GHIN expsat	DB	0.0601266	**0.93633**	0.0133188	0.0512748
	DM	0.028481	0.0363608	0.0106007	**0.850142**
	IR	**0.882911**	0.0204432	0.133188	0.0339943
	AI	0.028481	0.00686642	**0.842892**	0.0645892
NetClus	DB	0.0553833	**0.450802**	**0.500074**	0.0955971
	DM	0.163934	0.15815	0.128535	0.304584
	IR	0.179553	0.0512035	0.242707	0.112786
	AI	**0.60113**	0.339844	0.128684	**0.487033**
MDC	DB	0.186681	0.232455	**0.803727**	0.000000
	DM	0.261844	0.000000	0.128592	0.161790
	IR	0.003183	0.278748	0.000000	**0.75888**
	AI	**0.548292**	**0.488797**	0.067680	0.079323

(b) Cluster purity of FOUR_AREAS

Figure 5: Extrinsic verification of GHIN compared with NetClus and MDC

Incorporating the tiring factor into the two previously described reward functions :

$$sat(g_i, A_j) = \frac{t * |\psi^i(g_i) \cap A_j| - w * |A_j \setminus \psi^j(g_i)|}{|A_j|}$$

$$esat(g_i, A_j) = t * \frac{|\psi^j(g_i) \cap A_j| - exp_{ij}(g_i, A_j)}{\sqrt{var_{ij}(g_i, A_j)}} - w$$

6. EXPERIMENTAL RESULTS

Six real-world information networks summarized in figure 4 were utilized in the experimental study. Five of the datasets were derived from the 20NewsGroups dataset [4] containing the domains of documents, subject-lines, and all-terms. The FOUR_AREAS network was previously used in [27] and is a subset of the DBLP dataset containing the domains of papers, authors, abstracts, and conferences. All six networks contain class labels: Newsgroup datasets use the usenet group of a document as a class label, while the FOUR_AREAS dataset labels papers according to their research area as data mining, machine learning, database, or information retrieval. All code, data, and experimental setup are publicly available at `http://homepages.uc.edu/~alqadaf`.

6.1 Extrinsic Verification

The clusters produced by `GHIN` using both r_i^{sat} and r_i^{expsat} were compared with the clusters mined by `NetClus` [27] and `MDC` [6]. The `NetClus` algorithm was shown to outperform both `RankClus` [26] and `PLSA` [33], while `MDC` is a generalization and improvement of the seminal information-theoretic co-clustering algorithm introduced by Dhillon [13].

Terms	Authors	Conferences
data	Surajit Chaudhuri	VLDB
database	Divesh Srivastava	SIGMOD
queries	H. V. Jagadish	ICDE
databases	Jeffrey F. Naughton	PODS
querys	Michael J. Carey	EDBT
xml	Raghu Ramakrishnan	
mining	Jiawei Han	KDD
learning	Christos Faloutsos	PAKDD
data	Wei Wang	ICDM
frequent	Heikki Mannila	SDM
association	Srinivasan Parthasarathy	PKDD
patterns	Ke Wang	ICML

(a) Sample clusters from FOUR_AREAS

Abstract terms	All terms
israel	israelis
israelis	arabs
apartheid	palestinian
terrorism	occupied
jerusalem	zionist

(b) Sample clusters from MER

Figure 6: Sample clusters mined with GHIN

Cluster validity of each clustering was determined via the F_1, $F_{0.5}$, and F_2 measures of the B^3-Precision and B^3-Recall extrinsic cluster validity metrics [3]. These measures were selected to ensure a fair comparison of overlapping and non-overlapping clustering [3]; in addition, it was proven that the B^3 measures retain the desirable properties of cluster homogeneity, cluster completeness, rag bag, and cluster size vs quantity, while many popular extrinsic clustering measures such as precision, recall, mutual information and entropy do not. Let $C(g)$ and $L(g)$ denote the cluster and class label that object g belongs to respectively. Then the precision and recall of any pair of objects g and g' is given as

$$Prec(g, g') = \frac{\min(|C(g) \cap C(g')|, |L(e) \cap L(g')|)}{|C(g) \cap C(g')|} \quad (5)$$

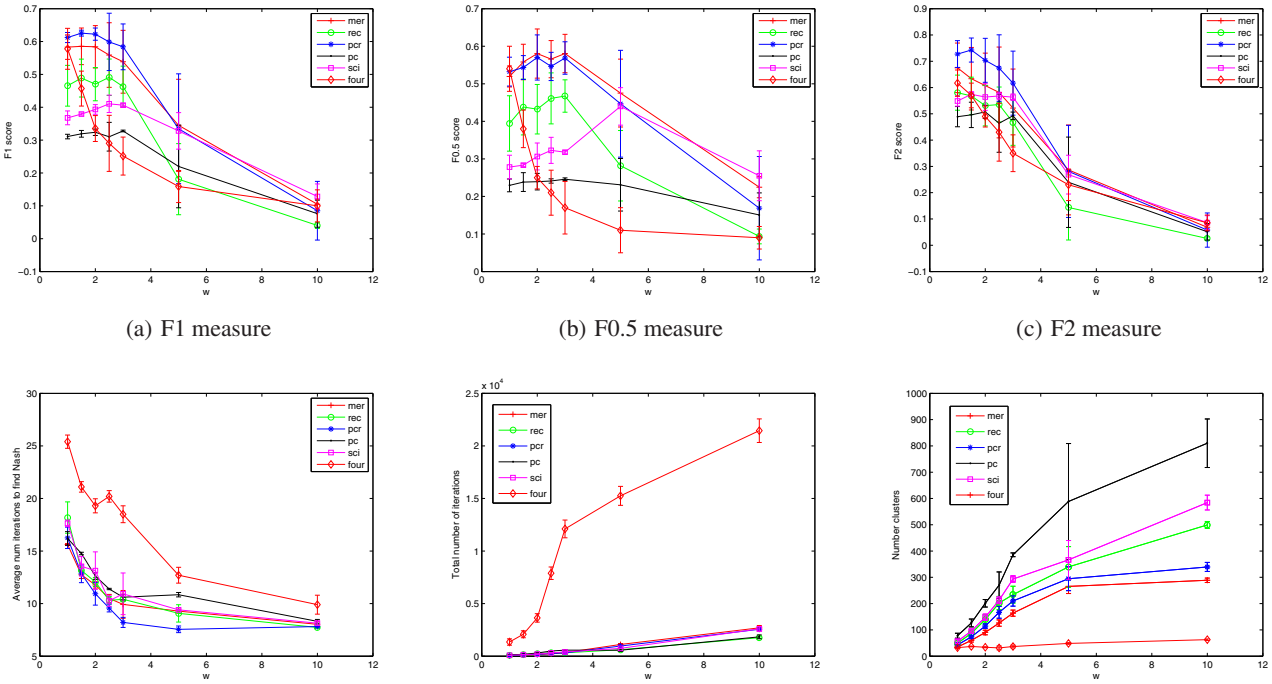
$$Rcl(g, g') = \frac{\min(|C(g) \cap C(g')|, |L(g) \cap L(g')|)}{|L(g) \cap L(g')|} \quad (6)$$

Note that equation 5 is only defined when g and g' share a cluster and equation 6 is only defined when g and g' share a class label. Intuitively, $Prec$ grows if there is a matching category for each clusters where two items co-occur; Rcl grows when we add a shared cluster for each class shared by two items. Thus if we have fewer shared clusters than needed, we loose recall; if we have fewer classes than clusters we lose precision. From these measures B^3-precision and recall are derived:

$$B^3 Prec = Avg_g \left[Avg_{g', C(g) \cap C(g') \neq \emptyset} \left[Prec(g, g') \right] \right] \quad (7)$$

$$B^3 Rcl = Avg_g \left[Avg_{g', L(g) \cap L(g') \neq \emptyset} \left[Rcl(g, g') \right] \right] \quad (8)$$

The $F_{0.5}$ measure weighs precision twice as high as recall while



(a) F1 measure (b) F0.5 measure (c) F2 measure

(d) Average number of iterations to find Nash equilibrium (e) Total number of iterations (f) Number of clusters

Figure 7: Parameter study on cluster quality with r_i^{esat}

F_2 weighs recall twice as much as precision and F_1 weighs them equally.

Both NetClus and MDC require the k parameter to indicate the number of desired clusters; this parameter was varied as 2, 4, 8, 16, 32, 64. The max value in GHIN was set to 50 along with tiring factor $f(x) = \frac{1}{x^{3/2}}$ for all experiments; w was varied as illustrated in the parameter study section. For NetClus, the cluster label is obtained according to the largest posterior probability. Each algorithm was run on every HIN with each parameter selection 20 times; the best F-measures are displayed in figure 5(a). As can be seen, GHIN consistently produces higher quality clusters in every information network. Comparing the average and median scores of all runs yielded very similar results. MDC and NetClus consistently produced the best results when $k = 4$ closely matching the number of "true" classes for all F measures; the quality of the clustering decreased significantly as k was increased. GHIN produced between 50 and 200 clusters depending on the setting of w (see parameter section). While the number of clusters is significantly higher than the "true" number of clusters, the clusters produced by GHIN represent very precise, fine-grain and nuanced clusters as opposed to the very broad and global approach NetClus and MDC employ. For example, while the number of categories in MER is only 2, clearly, finer delineations and sub-categories of news posts exist within the categories of Religion and Middle East politics. As seen in figure 6(b), GHIN produced a cluster with terms specifically related to the Israeli-Palestinian conflict; the cluster contained 71 posts all dealing with topic. The trend of smaller more focused clusters was consistently repeated for all datasets. Investigating the purity of clusters further emphasizes this point; figure 5(b) displays the distributions of the objects in the top four clusters (ranked by accuracy of the objects) in the FOUR_AREAS network. Once again, the topical nature of the clusters is evident. On the other hand, MDC and NetClus are hindered by the k parameter; small values of k obstruct precision while large k detracts from recall due to non-overlapping clusters.

6.2 Sample Clusters

Figure 6 displays clusters mined by GHIN from the various information networks. In all cases not the entire cluster is displayed but only the top 5-6 objects ranked by their individual satisfaction score. The clusters from FOUR_AREAS clearly correspond to the documents in the field of databases and data mining; as an added bonus we see that the satisfaction score ($esat$ in this case) does a good job of ranking the objects. Furthermore, the terms in the MER cluster are distinctly linked to the most commonly discussed issue in Middle Eastern politics: the Arab-Israeli conflict.

6.3 Parameter Study

Experiments were performed to study the effects of the w parameter with r_i^{esat}. The overall goal was to observe how cluster quality was effected by parameter selection in addition to examining the effect of w on the algorithmic characteristics of GHIN. We expect larger values of w to yield more precise and accurate clusters at the cost of recall. As can be observed in figures 7(b)-7(c) this trend is clearly visible. The $F_{0.5}$ measure tends to increase with larger values of w until a breaking point (generally where $w = 5$) at which the low recall values dominate the increasing precision. This trend is reiterated by the F_2 measure which continually decreases as w grows larger. While the number of clusters grows (figure 7(f)) with w, fewer objects are clustered overall. This trend is explained by the fact that as stricter criterion on cluster formation is imposed, fewer iterations of the refinement phase are executed and the original candidates form the clusters (figure 7(d)). In these cases the fact that GHIN is an incomplete algorithm leads to less stable results as evidenced by the larger standard deviations in the number of clusters formed (figure 7(f)). As a result, we recommend utilizing values of w between 1.5 and 3. Finally, figure 7(e) highlights

the fact that the set R tends to die down pretty quickly resulting in far fewer iterations than objects in the HIN. Hence, for the recommended range of w values, GHIN tends to run in $O(|G_i|^2)$ time. Actual running times on an AMD Atholon 64 X2 Dual Core with 6 GiB of ram did not exceed 200 seconds for all Newsgroup HINs. The execution time maxed out at 500 seconds on FOUR_AREAS utilizing the recommended range of w and exceeded 2 hours at values of $w \geq 5$.

7. CONCLUSION

In this paper a novel game-theoretic framework for heterogeneous information network clustering (GHIN) was presented. Modeling the clustering problem as a game in which players attempt to maximize their reward, clusters are defined as the Nash equilibrium solution concepts. This framework presents a unifying definition of a HIN-cluster. Furthermore, specific domain knowledge may be incorporated via specifying the rules of the game and developing differing reward functions. Utilizing an intuitive reward function we illustrated that the framework encompasses previous well-established formulations of bi-clustering. Additionally, two reward functions were developed in concert with GHIN. Experimental results on several real world information networks demonstrated that GHIN was especially advantageous in mining precise, fine-grain and nuanced clusters.

In depth algorithmic issues related to GHIN were not addressed and further studies are needed in this direction. Specifically, a generalized and efficient systematic approach for enumerating Nash equilibrium points within the confines of the clustering game should be developed as the randomization in the current scheme is a potentially destabilizing factor. Moreover, future work should focus on establishing more effective and efficient reward functions along with suitable heuristics.

8. REFERENCES

[1] F. Alqadah and R. Bhatnagar. An effective algorithm for mining 3-clusters in vertically partitioned data. In *CIKM '08*, 2008.

[2] F. Alqadah and R. Bhatnagar. Discovering substantial distinctions among incremental bi-clusters. In *SDM'09*, 2009.

[3] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12 (4):461–486, 2009.

[4] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[5] A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In *SDM'07*, 2007.

[6] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In *ICML '05*, pages 41–48, 2005.

[7] I. Bhattacharya and L. Getoor. Relational clustering for multi-type entity resolution. In *MRDM '05*, 2005.

[8] S. R. Bulo' and M. Pelillo. A game-theoretic approach to hypegraph clustering. In *NIPS2009*, 1571-1579, 2009.

[9] S. Busygin, O. Prokopyev, and P. M. Pardalos. Biclustering in data mining. *Comput. Oper. Res.*, 35(9):2964–2987, 2008.

[10] Y. Cheng and G. Church. Biclustering of expression data. In *ISMB '00*, 2000.

[11] H. Cho, I. Dhillon, Y. Guan, and S. Sra. Minimum sum squared residue co-clustering of gene expression data. In *SDM '04*, 2004.

[12] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD '01*, 2001.

[13] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD '03*, pages 89–98, 2003.

[14] B. Gamter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, Berlin, 1999.

[15] J. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67, No. 337:123–129, 1972.

[16] 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.*, 19(12):1625–1637, 2007.

[17] B. Long, X. Wu, Z. M. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs. In *KDD '06*, pages 317–326, 2006.

[18] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu. Spectral clustering for multi-type relational data. In *ICML '06*, pages 585–592, 2006.

[19] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1 (1):24–45, 2004.

[20] E. Mendelson. *Introducing Game Theory and Its Applications*. Chapman & Hall / CRC, 2004.

[21] G. Owen. *Game Theory*. Academic Press, 1995.

[22] R. Pensa and J. Boulicaut. Towards fault-tolerant formal concept analysis. *Congress of the Italian Association for Artificial Intelligence AI* IA*, 3673:21–23, 2005.

[23] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a nash equilibrium. In *Games and Economic Behavior*, pages 664–669, 2004.

[24] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. Murali. A monte carlo algorithm for fast projective clustering. In *SIGMOD '02*, pages 418–427, 2002.

[25] K. Sim, J. Li, V. Gopalkrishnan, and G. Liu. Mining maximal quasi-bicliques to co-cluster stocks and financial ratios for value investment. In *ICDM '06*, pages 1059–1063, 2006.

[26] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. Rankclus: Integrating clustering with ranking for heterogeneous information network analysis. In *EDBT'09*, 2009.

[27] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD'09*, 2009.

[28] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. In *ISMB 2002*, 2002.

[29] X. Yin, J. Han, and P. S. Yu. Cross-relational clustering with user's guidance. In *KDD '05*, 2005.

[30] X. Yin, J. Han, and P. S. Yu. Linkclus: efficient clustering via heterogeneous semantic links. In *VLDB '06*, 2006.

[31] M. J. Zaki and M. Ogihara. Theoretical foundations of association rules. In *SIGMOD-DMKD '98*, 1998.

[32] M. J. Zaki, M. Peters, I. Assent, and T. Seidl. Clicks: An effective algorithm for mining subspace clusters in categorical datasets. *Data and Knowledge Engineering special issue on Intelligent Data Mining*, 60 (2):51–70, 2007.

[33] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *KDD '04*, 2004.