# Towards Community Detection in Locally Heterogeneous Networks

Charu C. Aggarwal* Yan Xie† Philip S. Yu‡

**Abstract**

In recent years, the size of many social networks such as *Facebook*, *MySpace*, and *LinkedIn* has exploded at a rapid pace, because of its convenience in using the internet in order to connect geographically disparate users. This has lead to considerable interest in many graph-theoretical aspects of social networks such as the underlying communities, the graph diameter, and other structural information which can be used in order to mine useful information from the social network. The graph structure of social networks is influenced by the underlying social behavior, which can vary considerably over different groups of individuals. One of the disadvantages of existing schemes is that they attempt to determine *global communities*, which (implicitly) assume uniform behavior over the network. This is not very well suited to the differences in the underlying density in different regions of the social network. As a result, a global analysis over social community structure can result in either very small communities (in sparse regions), or communities which are too large and incoherent (in dense regions). In order to handle the challenge of local heterogeneity, we will explore a simple property of social networks, which we refer to as the *local succinctness property*. We will use this property in order to extract compressed descriptions of the underlying community representation of the social network with the use of a min-hash approach. We will show that this approach creates balanced communities across a heterogeneous network in an effective way. We apply the approach to a variety of data sets, and illustrate its effectiveness over competing techniques.

**Keywords:** Social networks, clustering

## 1 Introduction

Social networking has become an increasingly important application in recent years, because of its unique ability to enable social contact over the internet for geographically dispersed users. A social network can be represented as a graph, in which nodes represent users, and links represent the connections between users. Increased interest in the field of social networking has also resulted in a revival of graph mining algorithms. Therefore, a number of techniques have recently been designed for a wide variety of graph mining and management problems such as clustering, classification, frequent-pattern mining and indexing [3, 4].

Since social networks are typically very large, it is often challenging to develop effective community detection algorithms which work effectively for all parts of the network. On the other hand, social networks have a number of natural structural properties, which can be leveraged for designing more effective algorithms. One such property of social networks is that they are *locally heterogeneous*. This means that the link densities in different regions of the social network may be quite different. The heterogeneous nature of the social network creates a number of challenges for community detection. The use of *global analysis* can either construct very small communities in sparse local regions, or report large and incoherent communities in dense regions. Therefore, it is important to use *local structural analysis* over the social network in order to examine the relevance of a community relative to the local structure of the network.

It is also well known that social networks are often quite sparse [14] in terms of the underlying network structure. This means that the number of links emanating from a particular node is relatively small compared to the total number of nodes in the network. The neighbors for a given node are also typically correlated with one another by linkage behavior. Therefore, even in cases in which a node may have a large number of neighbors, these neighbors can be disjointed into a small number of correlated groups or communities. This effectively means that the number of communities that a given node may belong to is usually quite small. We refer to this sparse and correlated property of social networks as the *local succinctness property*. This property can be leveraged in order to characterize the global structural behavior of the social network as a compact decomposition of the local behavior. We will use this property in order to design an effective approach which first determines the local communities specific to different portions of the network and then merges these local mosaic of communities into a higher level global view. This approach allows us the flexibility to treat different portions of the network differently depending upon their locality behavior. We will show that this

*IBM T.J. Watson Research Center, charu@us.ibm.com
†University of Illinois at Chicago, yxie8@uic.edu
‡University of Illinois at Chicago, psyu@cs.uic.edu

two phase approach is far more effective in the determination of the underlying community structure than a global approach which determines all the global communities from scratch.

We will use a min-hash approach in order to determine a compact data structure representation which can be leveraged for finding a small number of local communities specific to each individual. This min-hash approach will exploit the local view of the social network for each node, and construct a *local projection* of that community for each individual. Then, we will merge the *local community projections* into a concise set of global communities. We note that since each global community maps onto multiple local community projections, we can achieve a high degree of compression of the community representation with the use of this approach. Furthermore, this provides a natural decomposition and interpretation of the global communities in the context of local characteristics of the data. We will study our results in the context of a number of real and synthetic data sets. We will examine the communities which are determined by using this approach, and show that such communities are far more accurate than those determined by well known global techniques such as those discussed in [8].

This paper is organized as follows. In the next section, we will introduce the quantification framework used to define local communities. In section 3, we will introduce a probabilistic min-hash approach which uses the observations in section 2 in order to determine the communities in the social network. In section 4, we will study the application of these techniques on a number of real and synthetic data sets. We will compare our method to a popular baseline approach. Section 5 contains the conclusions and summary.

**1.1 Related work** The problem of community detection related to that of finding dense regions in the underlying graph [1, 11, 17, 22]. The problem of community detection is generally defined in the form of a clustering of the underling network [5, 7, 15, 8, 14, 13, 18, 19, 20]. A survey of a number of important algorithms for community detection is provided in [20]. Discussion of important statistical properties of web communities is discussed in [14]. Evolutionary characteristics of dynamic communities are studied in [2, 6, 7, 12]. The problem of community detection has also been studied in the context of combining node content in order to improve its effectiveness [21, 23].

One of the key shortcomings of social networks is that they tend to treat the entire network in a uniform way. This leads to imbalanced communities with extremely large (but meaningless) communities in dense regions, and extremely small (but incomplete) communities in sparse regions. As this paper suggests, the structure of social networks exhibits certain natural locality properties which allow for the effective decomposition of the community detection problem into a number of smaller subproblems. This paper leverages these locality properties of social networks in order to design effective algorithms for community detection.

| Symbol | Description |
|---|---|
| $A$, $B$ | Node Sets. |
| $i$, $j$ | Node indices. |
| $\mathcal{N}(i)$ | Neighbor set of node $i$. |
| $\mathcal{N}(A)$ | Neighbor set of node set $A$ |
| $J(A, B)$ | Pairwise Jaccard coefficient between $A$ and $B$ |
| $J(A_1 \ldots A_n)$ | Multi-way Jaccard similarity for $A_1 \ldots A_n$ |
| $JN(A, B)$ | Pairwise Jaccard coefficient of neighbor sets, which is the same as $J(\mathcal{N}(A), \mathcal{N}(B))$ |
| $JN(A_1 \ldots A_n)$ | Multi-way Jaccard similarity for neighbor sets of $A_1 \ldots A_n$ |

Table 1: Description of notation.

## 2 Intuition: Local Heterogeneity and Succinctness

In this section we present a number of simple intuitions that motivate our community detection approach. Two key properties of most social networks are as follows: **(1)** The social network is usually sparse, and the degree of each node is small compared to the number of nodes in the network. **(2)** The social network exhibits considerable variations in the structure and local density over different regions of the data. This is reflected in the well known power-law behavior of node degree distributions [10].

This variation in local density is a challenge to the effective development of techniques for community detection, especially when global statistics are used in order to model the communities. This is because the natural density of different portions of the network is very different. Therefore, the use of any uniform approach for determining communities in different regions will either result in the loss of the ability to detect communities which are located in relatively sparse regions of the network, or in the determination of irrelevant communities in the network. Therefore, we will explore some of the locality characteristics of social networks and design an algorithm for community detection which is based upon this local behavior.

In real life, *communities are typically formed as a result of the interaction between particular entities during specific periods of time.* Since different periods of

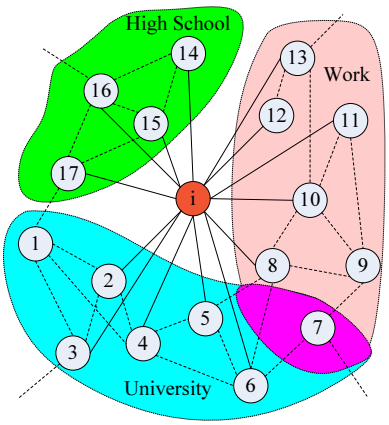


Figure 1: Illustration of local community structure.

time often lead to interactions with different geographical, interest, professional or student groups, this often leads to communities which may have some overlap but are *largely* disjoint from one another. It has often been observed that contacts belong in certain typical categories, such as friends from high-school, university or work, as illustrated in Figure 1. Furthermore, friends within each of these categories tend have stronger ties with members of the same category compared to others, although cross-category ties are still possible. In dense regions, some highly-connected bridging members may exist, which may connect unrelated communities, and such regions create a challenge for community detection algorithms. This is because such situations may result in the creation of unbalanced super-communities containing several unrelated members. A global approach of changing parametric descriptions in order to forcibly disconnect such communities may result in very small communities in sparse regions of the social network.

The discussion on local succinctness also suggests that each member belongs to a relatively small number of communities in the network. This suggests that it is possible to effectively decompose the analysis of community structure in a node-specific way without losing efficiency. In order to perform such a local analysis, we need to construct a set-based similarity measure, which characterizes the *local cohesiveness* of a set of nodes in the social network.

A widely popular similarity measure which is used for sets is the Jaccard coefficient.

Definition 1. (Jaccard similarity) *Given two sets $\mathcal{A}$ and $\mathcal{B}$, the Jaccard similarity is defined as follows:*

$$J(\mathcal{A}, \mathcal{B}) := |\mathcal{A} \cap \mathcal{B}|/|\mathcal{A} \cup \mathcal{B}|$$

Jaccard similarity can also be extended beyond pairs of sets to the multi-set case:

Definition 2. (Multi-way jaccard similarity) *Given a collection of sets $\mathcal{A}_1 \ldots \mathcal{A}_n$, the multi-way Jaccard similarity is defined as follows:*

$$J(\mathcal{A}_1 \ldots \mathcal{A}_n) := |\cap_i \mathcal{A}_i|/|\cup_i \mathcal{A}_i|$$

This measure has often been used for frequent pattern mining [9]. The case of social networks is particularly challenging because of the local heterogeneity in the underlying communities. Therefore, we will propose a local-community definition for social networks.

**2.1 Local Edge-based Communities** The local edge-based community is defined in terms of the Jaccard coefficient of the neighbor set of that node. In order to understand this measure, we first need to define the *edge group affinity* in terms of the Jaccard coefficient. Note that we distinguish this from the Jaccard coefficient on the node sets themselves, since the *neighbor sets* of the nodes are used for definition purposes. The neighbor set of a given node is defined as the set of nodes which are "linked to" by it, along with the node itself. We denote the *neighbor set* of node $i$ by $\mathcal{N}(i)$.

Definition 3. (Edge group affinity) *Given a set of nodes $\mathcal{I} \equiv \{i_1, i_2, \ldots, i_K\}$ the group affinity is defined as the multi-way Jaccard similarity of* **their neighbor sets**. *This similarity is defined as follows:*

$$JN(\mathcal{I}) := |\cap_{i \in \mathcal{I}} \mathcal{N}(i)|/|\cup_{i \in \mathcal{I}} \mathcal{N}(i)|.$$

We distinguish the *group affinity* $JN()$ from the direct set-based Jaccard $J()$, since the former is based on the *neighbor sets*, whereas the latter is based on the *sets themselves*. The group affinity satisfies the anti-monotonicity property.

Corollary 2.1. (Anti-monotonicity) *If $\mathcal{J} \subset \mathcal{I}$ then $JN(\mathcal{I}) \leq JN(\mathcal{J})$.*

The above condition is easy to verify, since the numerator of $JN(\mathcal{I})$ is the cardinality of the set intersection, whereas the denominator is the cardinality of the set union. The former reduces with increasing number of sets in the intersection, whereas the latter increases with increasing number of sets in the intersection. Thus results in the anti-monotonicity property.

The group affinity of a set of nodes is clearly dependent upon the behavior of its neighborhood. In particular, the measure depends on the number of *hub* nodes in its *neighborhood*. Therefore, we will formalize the concepts of set-neighborhood and hub-nodes.

Definition 4. (Neighborhood) *A node $j$ is in the neighborhood of $\mathcal{I}$, if it is connected to at least one node of $\mathcal{I}$. In other words, there exists $i \in \mathcal{I}$ such that $j$ lies in $\mathcal{N}(i)$.*

DEFINITION 5. (HUB NODE) *A node is a hub-node with respect to $\mathcal{I}$, if it is connected to all nodes of $\mathcal{I}$.*

The definitions of hub-node and neighborhood provide us with a natural and intuitive way of understanding the edge-based group affinity.

OBSERVATION 1. *The value of $JN(\mathcal{I})$ is the fraction of nodes from the neighborhood of $\mathcal{I}$, which are also hub nodes with respect to $\mathcal{I}$.*

A fully isolated clique with $K$ nodes has a Jaccard affinity of one. Any $K'$-node subset of an isolated clique has affinity of $K'/K < 1$. A non-isolated clique also has an affinity value less than one. Finally, if $\mathcal{I}$ contains no hub node, then the affinity value is zero.

In practice, as the set size of a community increases, the presence of a hub node becomes significantly less likely. For example, let us compute the edge-based group affinity for the set containing all the nodes in Figure 1. The denominator of the corresponding group affinity expression is equal to the eighteen depicted nodes, plus the four nodes adjacent to the fringe outgoing edges. This yields a total of twenty. However, the numerator is zero, because the group has no hub node, which is connected to *all* of these twenty nodes. Therefore, the affinity of this group is zero. This is reasonable to expect, since the set of nodes in Figure 1 contain a multitude of different communities which are quite unrelated to one another. On the other hand, many of the smaller communities are much more closely related to one another. Such smaller community values will have non-zero affinity values.

It is often the case in many real social networks, that smaller communities tend to be more tightly knit and have larger group affinities than larger communities. Therefore, it is important to construct the *locally relevant communities* for each node. These locally relevant communities are leveraged for the purpose of community detection. An important point is that we need some way of *quantifying* the behavior within the locality of a particular node. We do this by using a tail-thresholding technique. The tail-thresholding technique derives an appropriate Jaccard threshold within the locality of a node by examining the distribution of the *pairwise group affinities*.

DEFINITION 6. (TAIL THRESHOLDING) *For a given node $i$, let $p_1(i) \ldots p_n(i)$ represent the pairwise group affinities for the $n$ different 2-element sets containing $i$ and each of the $n$ different nodes. In other words, we have $p_r(i) = JN(\{i, r\})$, Let $\mu(i) = \sum_{r=1}^{n} p_r(i)/n$. Then, the tail threshold $T(i)$ for node $i$ is defined by $\mu(i)$.*

The aim of defining tail thresholds separately for each node is to make it sensitive to the behavior of that locality. We can now use this threshold in order to define local edge-based communities.

DEFINITION 7. (LOCAL EDGE-BASED COMMUNITIES) *A local edge-based community for node $i$ is a maximal set of nodes $\mathcal{G}$ which satisfy the following conditions:*

- *$\mathcal{G}$ contains node $i$*
- *The edge-based group affinity is above the* tail threshold *$T(i)$ for node $i$. In other words, we have $JN(\mathcal{G}) > T(i)$, and there is no superset $\mathcal{G}' \supset \mathcal{G}$ such that $JN(\mathcal{G}') > T(i)$.*

The local communities are eventually converted into a more consolidated set of communities. We will describe the motivation of this process below.

**2.2 Consolidating the Local Mosaic of Communities** A more compact and consolidated set of communities is more useful for data mining purposes. Furthermore, the local community for a particular node may not reflect all the members which are relevant to the underlying base community. For example, in the case of Figure 1, the local school-related community for any particular node may be pieced together with the other school-related nodes, in order to create a more coherent and complete community.

Once the local communities have been determined, we work with this set directly, and do not need to use edge-linkage behavior within this set. Rather, we use the set relationships between the different local communities in order to consolidate them. Specifically, we determine patterns from the local communities which share a large number of nodes and perform the consolidation process. In order to achieve this goal, we use the Jaccard coefficient between the different node sets. Therefore, the measure is defined with the direct use of the node set based Jaccard $J()$ rather than the linkage based Jaccard $JN()$. This is because we are interested in the edge-linkage behavior only at the local level in order to ensure that the differing density in different parts of the graph is properly accounted for. At the end of the first phase, the local heterogeneity has already been accounted for in the min-hash sampling process. Thus, we can work with the patterns obtained at the end of the first phase in a global way, without fear of loss of important communities. We further note that the local succinctness property also implies that the social network can be expressed in terms of a small number of local communities. This implies that the second step of consolidating the local communities is also tractable in practice.

## 3 The Local Min-Hash Scheme for Community Detection

In this section, we will discuss a local min-hash scheme for community detection in social networks. We use this approach as a proxy for effective pattern sampling in a way which accounts for the heterogeneities in the underlying network structure. The min-hash technique [9] is very useful in determining the normalized correlations between sets of nodes. It was first used [9] in the context of determining normalized associations between sets of items. Subsequently, it was used for characterizing the underlying structure of large dense graphs [11] by examining link correlations. This min-hash scheme works by first summarizing the entire social network in a single data structure of compressed size. This data structure can typically be held in main memory. Then, we create a small-size description of the underlying communities in the data. We will see that such an approach is flexible enough to accommodate both a local and global view. This provides a better understanding of how the communities relate both at the local and the global level. We will also see that this leads to a balanced clustering in which the heterogeneous nature of the community across the network is properly accounted for.

In order to represent the social network, we use the *conceptual representation of a* node-node adjacency matrix. For a network containing $n$ nodes, we create a $n \times n$ matrix, in which the entry $(i, j)$ is 1 if the node $i$ is linked to node $j$. Otherwise the entry $(i, j)$ is set to 0. All diagonal entries are always set to 1. Since we assume that "friendship linkages" are bi-directional, this matrix is symmetric in nature. In practice, this representation cannot be used efficiently, because the matrix is very sparse, and the number of nodes may be too large to maintain the entire $n \times n$ matrix explicitly.

In order to explain the approach further, we will first introduce the min-hash scheme for community detection. We sort the rows of this adjacency matrix, and determine the index of the first row for each column for which *any of these entries* are 1. It can be shown that the probability that these indices are the same for a pair of columns $i$ and $j$ is equal to the Jaccard coefficient used to measure the affinity between two nodes. This is because the denominator of the Jaccard Coefficient corresponds to a union event on set membership, whereas the numerator corresponds to the intersection event. The intersection event occurs if and only if all the min-hash indices for that set of columns are the same. Thus, by repeating this approach $k$ times, it is possible to estimate the Jaccard coefficient by computing the fraction of times (in the sample) that the Jaccard coefficient is the same between the two columns. In practice, the actual matrix is not used in order to perform the estimation, because we can work with only the edges incident to a node, as discussed below.

We construct $k$ different random sort-orders of the nodes. For each node $i$ and the $p$th sort-order ($p \in \{1 \ldots k\}$), we examine its links, and determine the node index $Q(p, i)$ for the first node *linked* to $i$ in this sort order. Thus, for each node $i$, we determine $k$ different minimum indices, which are denoted by $Q(1, i)$, $Q(2, i)$ $\ldots$ $Q(k, i)$. This creates a matrix $\mathcal{M}$ of size $k \times n$. For modest values of the sample factor $k$, this matrix can be held in main memory. This is because we make the assumption that $k$ is much smaller than $n$. All subsequent computations will be performed with the use of this summary structure which is held in main memory. We make the observation, that the Jaccard coefficient for a set of nodes $S$ can be *estimated* by determining the *fraction* of the $k$ rows in $\mathcal{M}$ for which all columns corresponding to $S$ take on the same value. We summarize as follows:

OBSERVATION 2. *For a given set $S = \{i_1 \ldots i_r\}$, the Jaccard-coefficient $\mathcal{AJ}(S)$ for the set is given by the fraction of rows from $\mathcal{M}$, such that each such row $j$ satisfies the following relationship:*

$$Q(j, i_1) = Q(j, i_2) = \ldots = Q(j, i_r) \tag{3.1}$$

The min-hash index is used in order to construct a *transactional representation* of the underlying data. *For each row*, we partition the set $Q(j, 1) \ldots Q(j, n)$ into groups for which the min-hash index values are the same. For each such partition, we create a categorical transaction containing the indices of the corresponding columns. For example, if a partition contains $\{Q(j, 3)$, $Q(j, 104)$, $Q(j, 232)$, $Q(j, 723)\}$, then we create the transaction $T_j = \{3, 104, 232, 723\}$. We can construct transactions $T_1 \ldots T_h$ corresponding to the different equi-index partitions from a single row. This process is repeated for each value of $j \in \{1 \ldots k\}$, and the transactions from each set are added to $\mathcal{T}$. We make the following observation:

OBSERVATION 3. *Let $\mathcal{T}$ be the transactions constructed from the min-hash index set. Then, the group affinity of a set of nodes $S$ is equal to the absolute support of $S$ in $\mathcal{T}$ divided by $k$.*

The absolute support of set $S$ is defined as the raw number of transactions which contain the set $S$. We note that the above transformation converts the first stage of the problem of local community pattern mining to a version of the frequent pattern mining problem. However, the standard frequent pattern mining problem uses a single global support in order to mine the patterns. This is not helpful for the case of our problem,

because we need to determine communities which are *locally relevant*. However, in this case, we need node-specific supports in order to determine the relevant patterns. In the next section, we will study this problem and a possible solution.

**3.1 Determining Local Communities** As discussed earlier, the local communities are defined based on a local threshold $T(i)$ which is used in order to define the importance of the community specific to node $i$. This threshold $T(i)$ translates into an *item-specific* (or more accurately *node-specific*) support for the frequent pattern mining problem. A straight forward solution is to determine the frequent patterns for node $i$ independently with the support value of $T(i)$, by considering only the portion of the database containing node $i$. This is however not an efficient solution to the problem, since it requires us to solve the frequent pattern mining problem as many times as the number of nodes in the data. For a large social network, this may lead to unacceptable running time. Therefore, we would like to create a pattern mining algorithm which can efficiently determine the underlying frequent patterns in the data. Therefore, we formalize this problem as follows:

PROBLEM 1. (LOCAL FREQUENT PATTERNS)
*Determine any locally frequent pattern $P$ from transaction set $\mathcal{T}$ with respect to local supports $T(1), \ldots, T(n)$, such that the support of $P$ in $\mathcal{T}$ is at least $min_{i \in P} T(i)$.*

In practice, there are tremendous numbers of overlaps in the local frequent patterns for different nodes, especially if the values of $T(i)$ for different nodes are close together. In the extreme case, when all values of $T(i)$ are the same, the problem translates to one application of the standard frequent pattern mining problem. A better solution is to *consolidate* the determination of frequent patterns. An important property of this formulation is that it continues to satisfy a *weak version* of the *downward closed property*. In [16], it has been shown how the problem of *finding frequent patterns with multiple minimum supports* can be solved effectively. We use this approach in order to mine the local frequent patterns from the underlying data.

Since the min-hash scheme is a randomized approach for determination of the communities, a natural question which arises is about the accuracy of the determination of the local communities. If the local communities are determined accurately, then the final set of communities will also be extremely accurate. In general, we would like to determine the probability of a false positive and false negative with the use of this approach. Therefore, we define the concept of $\delta$-false positives and $\delta$-false negatives.

DEFINITION 8. *A set of nodes $P$ is a $\delta$-false positive, if the Jaccard affinity in the original data is less than $min_{i \in P} T(i)$, but it is reported as a valid local community by the min-hash approximation with an estimated affinity of at least $min_{i \in P} T(i) \cdot (1 + \delta)$.*

Similarly, we can define the concept of a $\delta$-false negative.

DEFINITION 9. *A set of nodes $P$ is a $\delta$-false negative, if the Jaccard affinity in the original data is larger than $min_{i \in P} T(i)$, but it is not reported as a valid local community by the min-hash approximation scheme, since the estimated affinity is less than $min_{i \in P} T(i) \cdot (1 - \delta)$.*

We note that the concept of $\delta$-false negatives and $\delta$-false positives are a generalization of the concept of false negatives and false negatives. If the probability of $\delta$-false positives and $\delta$-false negatives is extremely low for small values of $\delta$, then we can practically obtain high quality approximations of the corresponding patterns, since we are assured that only false positives which are close to the threshold support have any chance of being erroneous, and the missed patterns can also be recovered by resetting thresholds appropriately. Furthermore, we can reset the thresholds by $(1 + \delta)$ and $(1 - \delta)$ to determine approximate subsets and supersets of the true patterns, and then use one more pass over the data in order to determine the exact set of patterns with high probability. We can show the following results:

THEOREM 3.1. *The probability that a given set of nodes $P$ is a $\delta$-false positive for a min-hash sample of size $k$ is given by at most $e^{-\delta^2 \cdot k \cdot min_{i \in P} T(i)/4}$.*

*Proof.* Let $Z_i$ be a 0-1 indicator variable which determines whether or not the $i$th sample of the min-hash contains the pattern $P$. Then, the random variable which denotes the value reported by the min-hash scheme is given by $Y = \sum_{i=1}^{k} Z_i$. Let $v$ be the expected value of the random variable $Y$. Since $v$ is at most $k \cdot \min_{i \in P} T(i)$, we can pick some value of $\phi > 0$, so that $v = \min_{i \in P} T(i)/(1 + \phi)$. Therefore, the probability that a pattern is a $\delta$-false positive is given by $P(Y > k \cdot (1 + \delta) \cdot (1 + \phi) \cdot v)$. Therefore, by using the Chernoff bound, we have:

$$\begin{aligned} P(Y > k \cdot (1+\delta) \cdot (1+\phi) \cdot v) &\leq e^{-v \cdot (\delta + \phi + \delta \cdot \phi)^2/4} \\ &\leq e^{-v \cdot (\delta + \delta \cdot \phi)^2/4} \leq e^{-k \cdot \delta^2 \cdot (1+\phi)^2 \cdot v/4} \\ &= e^{-k \cdot \delta^2 \cdot (1+\phi) \cdot \min_{i \in P} T(i)/4} \leq e^{-k \cdot \delta^2 \cdot \min_{i \in P} T(i)/4} \end{aligned}$$

The result follows.

**Algorithm** *ConsolidateCommunities*(Local Communities: $\mathcal{L}$; Number of Communities: $n_g$);
**begin**
  $\mathcal{R}$ = Set of $n_g$ randomly sampled patterns from $L$;
  **Phase I**
  **repeat**
    Assign each pattern in $P \in \mathcal{L}$ to the seed $S \in \mathcal{R}$ for which $J(P, S)$ is as large as possible;
    Let $Q_i \subseteq \mathcal{L}$ be the set of patterns assigned to the $i$th seed (where $i \in \{1 \ldots n_g\}$)
    Determine the nodes which occur in at least $f_{min} \times |Q_i|$ patterns of $Q_i$ and let $P_i$ be the set of such nodes;
    Reset $R$ by replacing the $i$th seed by $P_i$;
    Eliminate any centroid in $\mathcal{R}$ to which fewer than $n_{thresh}$ data points were assigned and replace by randomly sampled patterns from $\mathcal{L}$;
  **until**(termination_criterion1);
  **Phase II**
  **repeat**
    Assign each node in the graph to the pattern $P_i$ from $\mathcal{R}$ so that the node has the maximum number of incident links to $P_i$;
    Redefine the set $P_i$ as the set of nodes assigned to the corresponding cluster;
    { The sets $P_1 \ldots P_{n_g}$ define disjoint communities at this point }
    Redefine $\mathcal{R}$ with the new values of $P_1 \ldots P_{n_g}$;
  **until**(termination_criterion2);
  report the sets $P_1 \ldots P_{n_g}$ as the set of communities in the data;
**end**

Figure 2: Consolidating Local Communities

We can prove a similar result for the case of $\delta$-false negatives. In this case we use the lower tail Chernoff bound.

THEOREM 3.2. *The probability that a given set of nodes $P$ is a $\delta$-false negative for a min-hash sample of size $k$ is given by at most $e^{-\delta^2 \cdot k \cdot min_{i \in P} T(i)/2}$.*

**3.2 Consolidating Local Communities** The local communities determined in the previous section need to be consolidated into a final set of compact communities. This is because the min-hash technique of the previous section will create a large number of overlapping communities which need to be consolidated into a coherent set of communities. For this purpose, we use the parameter $n_g$ which determines the number of communities into which we wish to summarize the patterns. The choice of a smaller value of $n_g$ leads to a higher level of summarization, whereas larger values of $n_g$ lead to better granularity and detail. We use a two phase approach in which the first phase pieces together local communities in order to create the cores of the locally relevant communities, whereas the second phase then re-constructs these cores in a more comprehensive way with an iterative approach. For the first phase, we use a partitional methodology in which we start off with $n_g$ different seeds denoted by $\mathcal{R}$, which are sampled from the local community set. These seeds are then used to successively re-group the local patterns by using iterative assignment of the local patterns to the different centroids. Each pattern is assigned to the centroid with which it has the largest Jaccard similarity. In the second phase, nodes are assigned to communities to which they have the largest *link-based* similarity.

Let $\mathcal{L}$ be the set of local patterns determined with the use of local frequent-pattern mining algorithms. We sample $n_g$ patterns from the set $\mathcal{L}$ in order to create the seed set $\mathcal{R}$. This seed set $\mathcal{R}$ is successively refined in two phases. In the first phase, we compute the Jaccard similarity between each pattern in $\mathcal{L}$, and each of the seed sets in $\mathcal{R}$. For each pattern $P \in \mathcal{L}$ and seed $S \in \mathcal{R}$, we compute the Jaccard coefficient $J(P, S)$, and assign pattern $P$ to the seed for which it has the largest such similarity value. The set of patterns from $\mathcal{L}$ which are assigned to the $i$th seed are denoted by $Q_i$. Once the assignment is performed, we recompute the set $\mathcal{R}$, by using a *frequency-truncated centroid* of the data points assigned to a given pattern in $\mathcal{R}$. The frequency-truncated centroid of the set of patterns $Q_i$ is a set of nodes $P_i$. The set $P_i$ is computed by determining the set of nodes which occur in at least a fraction $f_{min}$ of the patterns in $Q_i$.

Some seeds from $\mathcal{R}$ may not be assigned many patterns from $\mathcal{L}$. Such seeds are outlier patterns, and can be discarded. Specifically, we discard all those centroids (or community cores) from $\mathcal{R}$ for which fewer than $n_{thresh}$ members[1] are assigned. These cores are replaced by randomly chosen patterns from the set $\mathcal{L}$. The repeated use of the above approach results in successively refined community cores. This approach is repeated iteratively. The termination criterion is that the average Jaccard coefficient between assigned patterns and their cluster seeds does not change by more than 1% from the last iteration. This is the end of the first phase, which provides the "core" of the different communities. However such cores may have overlap with one another, and may also miss some of the nodes entirely. In the second phase, we use these cores in order to re-define a crisper set of communities.

While the first phase is based on the use of the Jaccard Coefficient between *patterns* and *seeds*, the second phase is based on comparing the link behavior of nodes to individual seeds. The aim of the second phase

[1]For the purpose of implementation, we use $n_{thresh} = 3$.

is to re-organize the cores in a more comprehensive way with the use of link behavior. The cores of the second phase is inherited from the cores at the end of the first phase. As in the case of the first phase, the approach is iterative, in which the cores are successively refined, and the overlap between the different seeds are eliminated. Each node is assigned to the core which shares the maximum number of nodes in its linked neighborhood. Since the initial cores may be missing many nodes, it is possible that some nodes may not be assigned to any pattern at all. However, more and more nodes will be included in the cores in successive iterations, because of the use of linkage based assignment. As in the previous case, we keep track of the average objective function which is implicit in the assignment of nodes to the different communities. In this case, the objective function is defined as the average fraction of links of a node which point to the assigned community. The second phase is terminated, when this objective function does not improve by more than 1% in any particular iteration. The overall approach for consolidating local communities is illustrated in Figure 2.

## 4 Experimental evaluation

We tested the effectiveness of our approach on a number of real and synthetic data sets. As a baseline, we used the well known Newman algorithm [8].[2] We will first demonstrate the effectiveness of our approach with the use of a case-study on the DBLP data set. Then, we will use concrete quantitative measures in order to illustrate the superiority of our scheme. We will show that the local community-detection approach is able to adjust well to the varying density in different parts of the network and generate more balanced clusters which are qualitatively superior.

**4.1 Data Sets** The algorithm was tested on three data sets: two real data sets and one synthetic data set. In particular, the two real data sets we used were the well-known *DBLP data set* [3] and a *Condensed Matter Collaboration Network data set* [4]. The DBLP data set models co-author relationship among researchers, in which nodes correspond to authors, and edges correspond to co-author relationships between nodes. In the experiment, only those researchers who have no less than 5 papers were used for experimental purposes. The other real data set is the *Condensed Matter Collaboration Network data set* which models scientific collaborations between authors with papers in the topic of Condensed Matter Physics in arXiv. It has 21,363 nodes and 182,628 edges. Besides the two real data sets, one synthetic data set was generated by the R-Mat data generator [5]. In order to generate this data set, we used input parameters to be $a = 0.25$, $b = 0.25$, $c = 0.25$, $S = 16$, and $E = 200000$ (using the CMU NetMine notations). For community detection purposes, all the data sets were pre-processed in a way that only the largest single connected component was kept. This is also a basic input requirement of Newman algorithm. In the next sections, we will examine the relative effectiveness of the local community detection algorithm with the Newman algorithm with the use of both case studies and some more concrete effectiveness measures which are discussed below.

**4.2 Effectiveness Measures** In order to measure the effectiveness of the approach we used an *interest-ratio based link purity measure*. The idea was to remove some of the nodes and their incident edges from the data, and perform the clustering on the remaining data set. In practice, about 10% of nodes were removed for testing purposes. We test how well their links relate to the different clusters which were created without the use of these nodes. The nodes which were removed from the data during the clustering phase are referred to as *test nodes*. Ideally, we would like the links of a given (test) node to belong to a single community as far as possible. Therefore, for a given test node $i$, we determine the *dominantly linked community* as the community to which the node $i$ links the most. The *dominant purity* $p_i$ of node $i$ is defined as the fraction of the links of node $i$ which are incident on the dominant community. While this may seem like a good measure, it has the flaw that it is not very sensitive to the distribution of points among communities. For example, a trivial solution in which most nodes belong to a single community would result in a dominant purity value of 1 for most nodes. In order to deal with the issue of group cardinality distributions, we define the *dominant interest ratio* $I_i$ of a node $i$ as the ratio of the dominant purity of node $i$ to the fraction of the total number of network nodes which are contained in the dominantly linked community of node $i$. Let $N$ be the total nodes in the entire network, and $C_i$ be the total number of nodes in the dominant community of node $i$. Then the dominant interest ratio $I_i$ can be defined as follows:

$$I_i = \frac{p_i}{C_i/N} \tag{4.2}$$

The idea here is that the denominator of the expression (fraction of nodes in the dominantly linked cluster

[2] http://www.cs.unm.edu/ aaron/research/fastmodularity.htm
[3] http://dblp.uni-trier.de/xml/
[4] http://snap.stanford.edu/data/ca-CondMat.html
[5] http://www.cs.cmu.edu/~deepay/

of node $i$) would define the expected cluster purity in a completely random partitioning of the nodes. The overall effectiveness measure is the *average* of all the dominant interest ratios of the different test nodes. Thus, we average the value of $I_i$ for all test nodes $i$.

**4.3 Case Studies** In order to provide an intuitive idea of why the local community detection scheme performed well, we studied the communities created by the scheme on the DBLP data set. In both cases, we generated 400 communities. It turned out that the sizes of the communities were much more skewed in the case of the Newman algorithm, as compared to the local community detection algorithm. Many of the larger communities in the case of the Newman algorithm were putting together dense regions of the DBLP collaboration graph in order to create communities which correspond to completely unrelated topic areas. For example, the Newman algorithm created two very large communities each of which contained about 20% of the DBLP authors, when we ran the algorithm with an input parameter of 400 communities. On the other hand, for the same input parameter, the largest community in the case of the local community detection algorithm contained less than 1% of the total authors. One of these large communities generated by the Newman algorithm contained the following set of authors:
**Jiawei Han, Mani Srivastava, Rajeev Alur, Donald Towsley, Barbara Liskov** ...

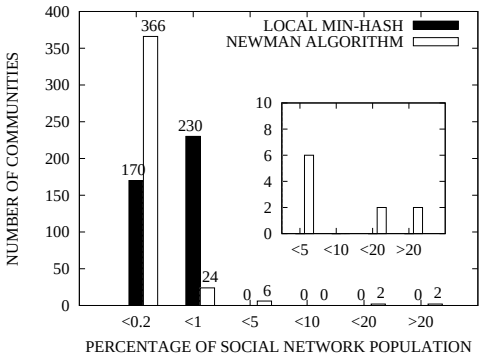


Figure 3: Distribution of Social Network Population in Communities

These authors work in very diverse areas. Jiawei Han is a well known expert in data mining, Mani Srivastava in communications networking, mobile and sensor networks, Donald Towsley in performance analysis (with some emphasis on networking), Rejeev Alur in model checking and verification systems, and Barbara Liskov in programming methodology. Such *chained* communities were created because of a number of bridge authors which created dense communities in certain regions of the DBLP graph. This created a chain of prolific authors from diverse areas. A global approach is unable to distinguish these aggregate trends from true community behavior. On the other hand, each of these authors was placed in a different community by the local community detection algorithm, each of which was itself much more coherent. For example, the community for Jiawei Han contained less than 1% of the total authors, and contained the following individuals:
**Jiawei Han, HongJiang Zhang, Lei Zhang, ChengXiang Zhai,** ...

All of these authors are well recognized in various areas of data, text or web mining. In addition, the community also contained a number of students of these faculty members. Most communities which were determined by the local approach were typically of *balanced size*, and contained a tightly knit community of core members. In order to understand this better, we will examine the distribution of the data points in different communities by the two schemes, when we used an input parameter of 400 communities. The results are illustrated in Figure 3. The histogram illustrates the number of communities, each of which contains a particular (range of) percentage of the entire social network population. All communities determined by the local algorithm were modestly sized, and most contained between 0.2% to 1% of the base population. There was no community with more than 1% of the population. On the other hand the Newman algorithm constructs two communities with more than 20% of the data points, and another pair with between 10% and 20% of the data points. These four communities contained more than 70% of the overall social network population on the aggregate. As discussed in the examples above the members in this 70% of the population are often quite diverse, and do not provide an interesting overview of the community behavior in the social network. At the same time, the Newman algorithm constructed a very large number of extremely small communities with less than 0.2% of the social network population. These two extremes correspond to the dense and sparse regions of the social network, which are treated in a very different way by the Newman algorithm, because of its uniform approach to inherently heterogeneous data. Neither of these extremes is helpful in determining interesting communities. On the other hand, as we will show with the help of quantitative measures in the next section, the local algorithm was able to determine interesting and coherent communities of modest size.

**4.4 Quantitative Effectiveness Results** In addition to the case study discussed above, we also tested

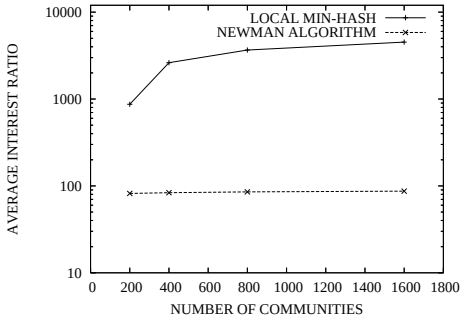


Figure 4: Quality Variations with Increasing Number of Communities (DBLP)

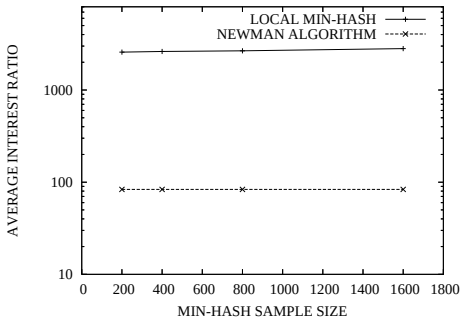


Figure 5: Quality Variations with Increasing Min-Hash Sample Size (DBLP)

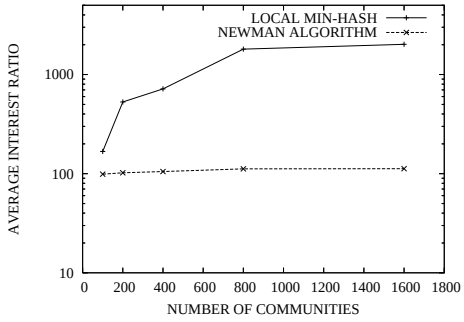


Figure 6: Quality Variations with Increasing Number of Communities (Condensed Matter Collaboration Network)

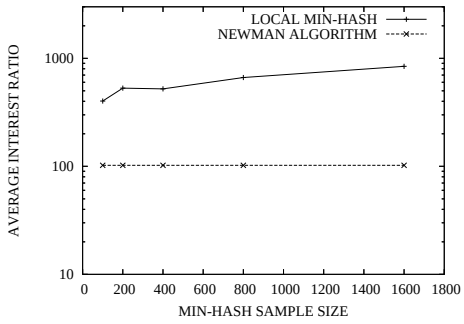


Figure 7: Quality Variations with Increasing Min-Hash Sample Size (Condensed Matter Collaboration Network)

our approach with the use of more concrete quantitative measures. Specifically, we used the statistical interest-based quantitative measures which are described in an earlier subsection. In each case, we will show that our approach is able to significantly outperform Newman's well known technique in terms of the interest measures. Thus, this provides a more concrete validation of what we have already demonstrated with the use of the case-studies discussed above.

In Figure 4, we have illustrated the variation in quality with an increasing number of communities. The number of communities is illustrated on the $X$-axis, whereas the interest ratio is illustrated on the $Y$-axis. The number of communities on the $X$-axis varied between 200 and 1600. The size of the min-hash sample was fixed at 400. It is clear that our local community detection scheme is significantly superior to the Newman algorithm in terms of the statistical interest ratio. It is important to note that the Y-axis is on a logarithmic scale, and the *local min-hash technique outperforms the Newman method by between one and two orders of magnitude*. For example, when we set the number of communities at 400, the average interest ratio for the local min-hash scheme was 2621.38, whereas that for the Newman method was 83.5063. A second broad trend which we observed was that the interest ratios increased with the number of communities. This is because the use of a larger number of communities is able to separate out the distinct communities (and sub-communities) much better. It is also evident from Figure 4, that this trend is more pronounced for the local min-hash technique, as compared to the Newman algorithm. For example, when we tested the scheme with an input community cardinality parameter which was set at 200 communities, the average interest ratios for the min-hash and Newman schemes were 871.771 and 81.9612 respectively. When we used an input parameter of 1600 communities, the average interest ratios for the min-hash scheme increased to 4525.42, whereas that for the Newman scheme increased to 87.1342. Thus, the effectiveness of the local min-hash technique increased by a factor of 5.19, whereas that of the Newman algorithm increased by a factor of only 1.06. The difference in trends with increasing number of communities for the two schemes is because the min-hash technique is able to discriminate among the

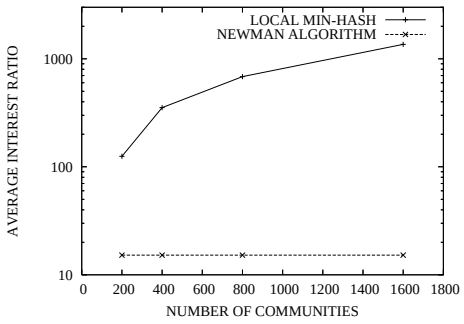


Figure 8: Quality Variations with Increasing Number of Communities (RMAT Generated Set)

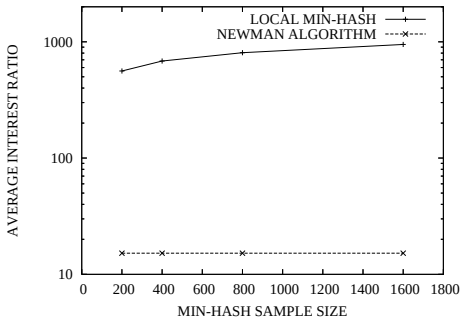


Figure 9: Quality Variations with Increasing Min-Hash Sample Size (RMAT Generated Set)

different communities better over different parts of the network as the number of communities increases. The effect of locality becomes even more crucial for the case of fine-grained community construction. This is not quite as true for the Newman algorithm in which some of the coarser communities in some local portions of the network do not get discriminated better with increasing number of communities.

In Figure 5, we have illustrated the variation in the effectiveness of the schemes with increasing min-hash size. In this case, we fixed the number of communities at 400. We note that the Newman technique does not use a min-hash approach, and therefore its effectiveness is simply a horizontal line on the chart as a baseline. The min-hash size is illustrated on the $X$-axis, whereas the effectiveness in terms of the average interest ratio is illustrated on the $Y$-axis. The results show that while it is possible to increase the effectiveness of the scheme by increasing the min-hash sample size, the results of the local community detection algorithm are quite robust to the use of different sample sizes. Furthermore, the local community detection technique outperforms the Newman algorithm over all ranges of the min-hash sample sizes. Even, at the lower-end of the min-hash sample size of 200, the local community detection technique outperforms the Newman algorithm by a factor of about 30.87. The robustness of the min-hash sample size suggests that it may be possible to work effectively at the lower end of the sample sizes in order to achieve very good results.

The trends with increasing number of communities for the *Condensed Matter Collaboration Network Data Set* are illustrated in Figure 6. The min-hash sample-size was fixed at 200. In this case, the trends with the increasing number of communities are even more pronounced. For example, at the lower end, when only 100 communities are used, the local community detection outperforms the local min-hash technique by a factor of about 1.69. On the other hand, at the higher end, when we use 1600 communities, the local min-hash scheme outperforms the Newman algorithm by a much greater factor of 18.00. With an increase in the number of communities, the importance of heterogeneity in locality increases, and therefore the local scheme performs much more effectively than the global scheme. The variations in effectiveness with increasing min-hash sample size for this data set are illustrated in Figure 7. The number of communities was fixed at 200. In this case, there was more variation in effectiveness with the sample size for the local community detection, as compared to the DBLP data set. However, even in this case, the local community detection was very robust and significantly outperformed the Newman algorithm over the entire range of sample sizes.

The results for the R-MAT synthetic data set with increasing number of communities are illustrated in Figure 8. The min-hash sample size was fixed at 400. In this case, the Newman algorithm was almost completely invariant on quality with increasing number of communities. On the other hand, the local community detection approach improved significantly with increasing number of communities. The variation with increasing sample-size is illustrated in Figure 9. The number of communities was fixed at 800. As in previous cases, the min-hash scheme was extremely robust to the use of different sample-sizes, and was significantly superior to the Newman algorithm over the entire range of tested parameters. Thus, the results suggest that the use of the local approach offers significant advantages for community detection in heterogeneous social networks, and this advantage increases when more fine grained communities are determined. In many cases, our approach provides orders of magnitude advantage in the quality of the underlying communities, because it focuses on determining interesting local variations with a carefully designed min-hash technique.

## 5 Conclusions and Summary

In this paper, we examined the problem of community detection in social networks from the perspective of the heterogeneity of the link density in the social networks. Such heterogeneous densities can result in an inability of global algorithms to behave in a balanced way across dense and sparse regions of the network. For example, this could result in a global algorithm either chaining together irrelevant members in a single community, or it could result in very small communities in sparse regions of the network. As an example, we studied a well known algorithm by Newman, and showed (both by case studies and aggregate quantitative results), that the local approach proposed by our method is significantly superior to this algorithm. This is because our method uses carefully designed local methods in order to extract interesting patterns from all parts of the network. This can adapt well to local variations in density and provide coherent and balanced clusters over the entire social network.


## Acknowledgements

We would like to thank Spiros Papadimitriou for his comments and help in drawing figures.

The first author's work was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. The work of the second and third authors is supported in part by NSF through grants IIS-0905215, DBI-0960443, OISE-0968341 and OIA-0963278.


## References


[1] J. Abello, M. G. Resende, and S. Sudarsky, *Massive quasi-clique detection*, LATIN, (2002), pp. 598–612.

[2] L. Backstrom, D. P. Huttenlocher, J. M. Kleinberg, and X. Lan, *Group formation in large social networks: membership, growth, and evolution*, KDD Conf., (2006), pp. 44–54.

[3] C. Aggarwal and H. Wang, *Managing and Mining Graph Data*, Springer, (2010).

[4] C. Aggarwal, *Social Network Data Analytics*, Springer, (2011).

[5] D. Bortner and J. Han, *Progressive clustering of networks using structure-connected order of traversal*, ICDE Conf., (2010), pp. 653–656.

[6] D. Chakrabarti, R. Kumar, and A. Tomkins, *Evolutionary clustering*, KDD Conf., (2006), pp. 554–560.

[7] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, *Evolutionary spectral clustering by incorporating temporal smoothness*, KDD Conf., (2007), pp. 153–162.

[8] A. Clauset, M. E. J. Newman, and C. Moore, *Finding community structure in very large networks*, Phys. Rev. E 70, 066111, (2004).

[9] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, and J. Ullman, and C. Yang, *Finding interesting associations without support pruning*, IEEE TKDE, 13(1), (2001), pp. 64–78.

[10] M. Faloutsos, P. Faloutsos, and C. Faloutsos, *On power law relationships of the internet topology*, SIGCOMM, (1999), pp. 251–262.

[11] D. Gibson, R. Kumar, and A. Tomkins, *Discovering large dense subgraphs in massive graphs*, VLDB Conf., (2005), pp. 721–732.

[12] M.-S. Kim and J. Han, *A particle-and-density based evolutionary clustering method for dynamic networks*, PVLDB, 2(1), (2009), pp. 622–633.

[13] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, *Trawling the web for emerging cyber-communities*, Computer Networks 31(11-16) (1999), pp. 1481–1493.

[14] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, *Statistical properties of community structure in large social and information networks*, WWW Conf., (2008), pp. 695–704.

[15] Y.-R. Lin, J. Sun, P. Castro, R. B. Konuru, H. Sundaram, and A. Kelliher, *Extracting community structure through relational hypergraphs*, WWW Conf., (2009), pp. 1213–1214.

[16] B. Liu, W. Hsu, and Y. Ma, *Mining association rules with multiple minimum supports*, KDD Conf., (1999), pp. 337–341.

[17] J. Pei, D. Jiang, and A. Zhang, *On mining cross-graph quasi-cliques*, KDD Conf., (2005), pp. 228–238.

[18] M. Rattigan, M. Maier, and D. Jensen, *Graph Clustering with Network Structure Indices*, ICML Conf., (2007), pp. 783–790.

[19] V. Satuluri, and S. Parthasarathy, *Scalable graph clustering using stochastic flows: applications to community discovery*, KDD Conf. (2009), pp. 737–746.

[20] L. Tang and H. Liu, *Graph mining applications to social network analysis*, Managing and Mining Graph Data, Ed. Charu Aggarwal, Haixun Wang, (2010).

[21] T. Yang, R. Jin, Y. Chi, and S. Zhu, *Combining link and content for community detection: a discriminative approach*, KDD Conf., (2009), pp. 927–936.

[22] Z. Zeng, J. Wang, L. Zhou, and G. Karypis, *Out-of-core coherent closed quasi-clique mining from large dense graph databases*, ACM Transactions on Database Systems, Vol 31(2), (2007).

[23] Y. Zhou, H. Cheng, and J. X. Yu, *Graph clustering based on structural/attribute similarities*, Proc. VLDB Endow., 2(1), (2009), pp. 718–729.