# InfoSearch: A Social Search Engine

Prantik Bhattacharyya and Shyhtsun Felix Wu

**Abstract** The staggering growth of online social networking platforms has also propelled information sharing among users in the network. This has helped develop the user-to-content link structure in addition to the already present user-to-user link structure. These two data structures has provided us with a wealth of dataset that can be exploited to develop a social search engine and significantly improve our search for relevant information. Every user in a social networking platform has their own unique view of the network. Given this, the aim of a social search engine is to analyze the relationship shared between friends of an individual user and the information shared to compute the most *socially relevant* result set for a search query.

In this work, we present *InfoSearch*: a social search engine. We focus on how we can retrieve and rank information shared by the direct friend of a user in a social search engine. We ask the question, within the boundary of only one hop in a social network topology, how can we rank the results shared by friends. We develop *InfoSearch* over the Facebook platform to leverage information shared by users in Facebook. We provide a comprehensive study of factors that may have a potential impact on social search engine results. We identify six different ranking factors and invite users to carry out search queries through *InfoSearch*. The ranking factors are: ‘diversity’, ‘degree’, ‘betweenness centrality’, ‘closeness centrality’, ‘clustering coefficient’ and ‘time’. In addition to the *InfoSearch* interface, we also conduct user studies to analyze the impact of ranking factors on the social value of result sets.

## Keywords

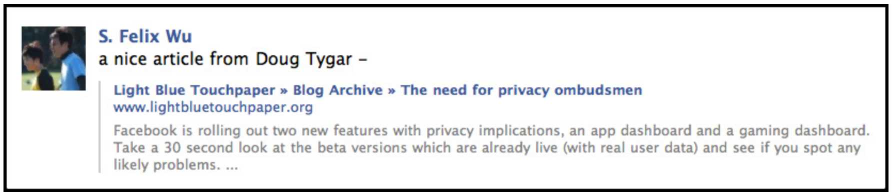
Online Social Network, Social Search.

Prantik Bhattacharyya
University of California, Davis, 1 Shields Ave, Davis, CA, e-mail: pbhattacharyya @ucdavis.edu

Shyhtsun Felix Wu
University of California, Davis, 1 Shields Ave, Davis, CA e-mail: sfwu@ucdavis.edu

## 1 Introduction

Users in online social networks have surpassed hundreds of millions in number. With this staggering growth in the network size, social network platforms like Facebook and Twitter have introduced various software tools to engage users. In addition to connecting and exchanging messages with friends on a regular basis, social network platforms also provide a great place to share useful information. Consequently, people have become very good at sharing the information that they value, support, endorse and think their friends might benefit from. Users share their favorite web-page(s) on current affairs, news, technology updates, programming, cooking, music and so on by sharing Internet URLs with their friends through the social network platform. Facebook has introduced 'Like', 'Share' and 'Recommend' buttons that content providers of any website can include on their website to help visitors share the URLs with their friends in a fast and easy way. Twitter has also introduced similar technologies to let users 'Tweet' the URL in addition to their personal comment about the URL.



S. Felix Wu
a nice article from Doug Tygar –

Light Blue Touchpaper » Blog Archive » The need for privacy ombudsmen
www.lightbluetouchpaper.org

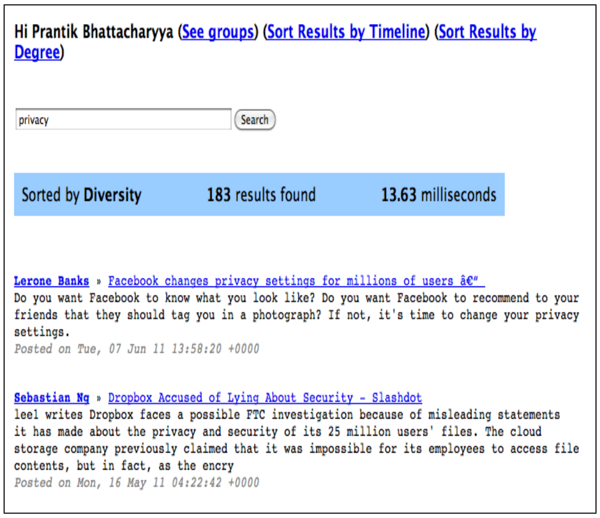
Facebook is rolling out two new features with privacy implications, an app dashboard and a gaming dashboard. Take a 30 second look at the beta versions which are already live (with real user data) and see if you spot any likely problems. ...

Fig. 1: Example of Information Sharing over Online Social Network (Facebook in this example).

The simplicity and ubiquitousness of this technology has propelled the integration of the web graph with the social graph. The additional information present in each individual's personal network can be utilized to develop search engines that include social context in information retrieval and ranking. In typical web search engines, users are restricted to search for information from the global web and retrieve results that are ranked relevant by a search engine's algorithm. For example, web search engines like Google, Yahoo! and Bing traditionally analyzes the information present in the form of hyper-link structures to rank results during a typical query. The intuitive justification for utilizing the hyperlink structure to rank web-pages is based on the idea that one web-page links to another web-page to indicate usefulness and relevance. During the process of crawling, indexing and ranking, each search engine formulates result set(s) for a set of keyword that are unique in nature and are identical to every user visiting the search engine. For example, when users search for queries related to 'programming' or 'cooking recipes', search results are similar in nature to every individual performing a query on the engine.

A search engine result set, however, can be significantly updated to incorporate social context as a factor during the ranking process. The social context in retrieving results will allow users to identify results based on the way their friends have shared and endorsed similar information. Each search query from a user will thus retrieve a unique set of result. The exclusive nature of each result set will thus be based on the large volume of information available in each individual user's personal network. The search process thus not only enables a user to access a set of information that has a distinct social component attached to it but also to gain from the collective knowledge of their respective social network. In other words, a search process is no longer limited to retrieving a random piece of information from the Internet with no trust value attached to it but extends to a retrieval process that includes a trusted source, that is, their friends' personal attachment or endorsement of that piece of information. Providing search results exclusively from the personal network of users creates a scope of unique challenges. How do we understand the relative importance of one user to another user in the network? How do we rank individual users? What are the primary factors that exemplify social relationship semantics?

The growth in the volume of shared information has also altered the way major search engine providers like Google and Microsoft rank web-search results. In 2011, the search engine companies introduced signals in their ranking algorithms to reflect patterns of information share across the social graph [24, 39]. The primary efforts are concentrated to introduce signals from social sharing to explore popularly shared URLs and boost their corresponding rankings in a result set that continues to be identical for all users with respect to a specific query.



Hi Prantik Bhattacharyya (See groups) (Sort Results by Timeline) (Sort Results by Degree)

Sorted by **Diversity** **183** results found **13.63** milliseconds

**Lerone Banks** » Facebook changes privacy settings for millions of users â€"
Do you want Facebook to know what you look like? Do you want Facebook to recommend to your friends that they should tag you in a photograph? If not, it's time to change your privacy settings.
*Posted on Tue, 07 Jun 11 13:58:20 +0000*

**Sebastian Ng** » Dropbox Accused of Lying About Security - Slashdot
leel writes Dropbox faces a possible FTC investigation because of misleading statements it has made about the privacy and security of its 25 million users' files. The cloud storage company previously claimed that it was impossible for its employees to access file contents, but in fact, as the encry
*Posted on Mon, 16 May 11 04:22:42 +0000*

Fig. 2: Screenshot of InfoSearch Application on Facebook: Results for the query 'privacy' appear for one of the authors.

In this work, we develop a search engine to demonstrate how user shared information can be exploited to deliver search results. Our work can be described in two parts. In the first part, we develop the social search engine system based on the Facebook platform that leverages the information shared by users in Facebook. The search engine is called *InfoSearch* and is available at https://apps.facebook.com/ infosearch. In the second part of our work, we discuss key issues that influence result ranking. We explore questions on how we can define the best result in a social context. In the absence of ground truth data about the relationship shared between two users (in real or online life), we investigate different ranking factors to analyze the social relationship between two users and rank search results. We provide a comprehensive study of factors that impact social search engine results. The ranking factors are based on an analysis of the structure of the social relationship between friends of a given user: social diversity, three different measures of centrality: degree, betweenness centrality and closeness centrality, a measure of clustering: clustering coefficient and finally a factor based on the time property of a shared information. We derive the social relationship between two users (friends) of a given user based on the social group structure shared between them in the user's individual social network. We present results based on the impact of the above ranking factors in retrieving information through user studies.

In section 2, we discuss related work. We formally describe the problem statement related to social search in section 3 and follow up with a discussion of social network relationship semantics in section 4. In section 5, we discuss the ranking factors and corresponding algorithms and section 6 describes the system development process. Section 7 presents statistics on usage. In section 8, we present our findings obtained through user studies and section 9 concludes with a discussion of future research directions.

## 2 Related Work

We discuss related work in this section. First, we discuss work in the area of search in social networks. Second, we discuss research related to the study of social relationship semantics. We primarily focus on research related to group and community formation in social networks.

Several projects have looked into the area of search in social networks. The research problems have broadly fallen into the following categories. First, the identity or profile search problem in which social network information is used to connect and subsequently search for users. Dodds et. al. [13] conducted a global social-search experiment to connect $60,000$ users to 18 target persons in 13 countries and validated the claims of small-world theory. Adamic et. al. [1] conducted a similar project on the email network inside an organization. More recently, Facebook has introduced 'Graph Search' [14] that aims to help user search for content linked by their friends. Facebook defines a content as any object on the open graph api. Examples of object in the open graph api include facebook-pages (e.g. a facebook account created by a local business, musician, artist) , facebook-apps (e.g. social games), facebook groups (e.g. university course groups, athletic group), photos shared by its users and geographic locations shared by the users.

In the second category, social networks have been leveraged to search for experts in specific domains and find answer to user questions. Lappas et. al. [22] addressed the problem of searching a set of users suitable to perform a job based on the information available about user abilities and compatibility with other users. The work in [10] attempted at automated FAQ generation based on message routing in a social network through users with knowledge in specific areas. Other works in similar directions have also been presented, e.g. [9, 32]. Query models [2] based on social network of users with different levels of expertize for the purpose of decentralized search have also been developed. Horowitz et. al. [20] presented *Aardvark*, a social network based system to route user questions into their extended network to users most likely knowledgeable in the context of the question.

In the third category, social networks are considered to improve search result relevancy. User connections are interpreted as a graph such that a user can be represented as a node and each friend connection can be treated as an edge between two nodes. Haynes et. al. [19] studied the impact of social distance between users to improve search result relevancy in a large social networking website, *LinkedIn*. The author defined the social distance between users based on the tie structure of the social graph and aims to provide improved relevance and order in profile identity entries. Link analysis algorithms, like PageRank [6, 8, 12], are also not suitable for application since during the search process of an individual user, results from members of their social circle should not be ranked based on a generalized analysis of the relative importance of those members in the larger network but rather on their local importance to the querying user [23, 37]. Mislove et. al. [25] considered the problem of information search through social network analysis. They compare the mechanisms for locating information through web and social networking platforms and discuss the possibility of integrating web search with social network through a HTTP proxy.

A primary way to understand social relationships is by analyzing social group formation in social networks. Work in group detection in graphs are primarily associated with community detection and graph partitioning problems. Past works [28] describe the motivation and technical differences between the two approaches. Detailed discussions can also be found in the recent survey [15]. Here, we discuss works related to community detection in social networks.

A common approach for finding sub-communities in networks uses a percolation method [11, 31, 30]. Here, $k$-clique percolation is used to detect communities in the graphs. Cliques in the graph are defined as complete and fully connected subgraphs of $k$ vertices. Individual vertices can belong to multiple cliques provided that the overlapping subgroups don't also share a $(k-1)$ clique. The work in [16] uses centrality indices to find community boundaries in networks. The proposed algorithm uses betweenness between all edges in the network to detect groups inside the graph. The worst-case runtime of the algorithm is $O(m^2n)$ for a graph of $m$ edges and $n$ vertices and is unsuitable for large networks. Improvements in the runtime have been suggested in later works [33, 35]. Impact of network centrality on egocentric and socio-centric measures have also been studied [23].

The betweenness approach places nodes in such a way that they exist only in a single community, restricting the possibility of overlapping communities and detecting disjoint groups in the network. To overcome this shortfall, algorithms in [17, 18] have proposed the duplication of nodes and local betweenness as a factor in detection of communities. Other approaches to identify overlapping communities have also been proposed [5, 4]. The above works describe the community structure based on relative comparison with the graph segment not included in the community [27] or based on comparisons with random graphs of similar number of nodes and vertices but different topological structures. For example, the definition of modularity [29] as an indicator of the community strength defines the measure as a fraction of the edges in the community minus the edges in a community created by the same algorithm on a random graph. Community definitions also include detection of groups within the network such that the interconnection between the different groups are sparse [17, 18]. In this work, we build the social search system on Facebook, utilizing the existing social graph as well as the information database being built by users. We discuss the details next.

## 3 Social Search - Problem Statement

In this section, we start with a discussion of the benefits of a social search engine and end by introducing the key information structures.

A user introduces an article to his/her friends by sharing the article URL on Facebook. It can be intuitively theorized that the user shared the article because he/she found the article to be relevant and beneficial in a particular context. Through the sharing process, the user extends the information database of his/her social network with the context of the shared article and consequently other friends in their network can benefit from this endorsement. In the example of Figure 1 the primary context of the article is ‘privacy’. Users in the network benefit from this shared knowledge when they try to find information related to ‘privacy’. Furthermore, the social context in this case i.e. the person who shared this information can help querying user(s) to disambiguate and choose from a large number of articles available on ‘privacy’ in general on the web.

It is important to understand that the subjectiveness of social relationships make it extremely difficult to correctly predict the value of each relationship. Furthermore, in the absence of ground truth data, it is also difficult to accurately postulate that one friend or user is more important compared to another user. In this direction, we focus on computing the most socially relevant “result set” rather than emphasizing on ranking individual results in a result set. Thus, in this work the relevance of a comprehensive result set is given a higher priority over the ranking of individual results during a search query and relevance values of each result sets are determined to select the best result set for a given user query. Next, we formally define the key information structures required to develop a social search engine and rank query results.

**Definition 1. Social Network:** A social network is a graph $G = (V,E)$, where $V$ is a set of nodes and $E$ is a set of edges among $V$. A node stands for a user in the social network, and an edge $e$ stands for a connection between two users $u$ and $v$. In our work, we consider undirected edges. The shortest geodesic distance between two nodes $n_1$ and $n_2$ in the network is defined as $d(n_1,n_2)$. Let $d(n_1,n_2) = \infty$ if no path exists between the nodes in the network.

**Definition 2. Ego Network:** For a user $u$, ego network is a graph $G(u) = (V(u),E(u))$, where $V(u)$ is a set of nodes that includes all friends of $u$, $F(u)$ and the node $u$ itself. $E(u)$ is a set of edges among $(V(u)-u)$ such that $\forall v \in (V(u)-u)$, $v$ and $u$ are friends and share an edge in $E$. Additionally, all edges between nodes in $(V(u)-u)$ that existed in $E$ are also included in $E(u)$.

**Definition 3. Mutual Friend Network:** A mutual friend network of an user $u$ is defined as a subset of the ego network, represented as $MF(u) = (F(u),E'(u))$. $F(u)$ is the set of all friends of user $u$ and $E'(u)$ is a subset of the edges from $E(u)$ with the edges between user $u$ and nodes in $F(u)$ absent.

**Definition 4. Shared Information:** A shared information in a social network can be identified as an URL or a document. An URL or document shared by a user $u$ is denoted by the tuple $(u,d)$. Each shared URL or document is tagged by a set of keywords $K(d) = (k_1^d, k_2^d, ..., k_m^d)$. Additionally, each information is also tagged by a time-stamp, $T(d)$, based on the time the information was shared by the user in the social network platform.

**Definition 5. Query:** A query $q$ by a user $u$ is defined as $Q(u,q)$. The query $q$ can be a single keyword or a set of keywords i.e. a key-phrase. We discuss details about how we distinguish keywords and key-phrases during the search process later in section 6.

**Definition 6. Factor:** The term 'factor' is used to define a ranking factor that orders and ranks results in the search process. The factors used in this work are defined in section 5.

**Definition 7. Result Candidates:** The result candidates, $RC(Q(u,q))$ for a query $Q(u,q)$ is defined as the set of shared document tuples $(v_i,d_j)$ such that $v_i \in F(u)$ and $\forall d_j, q \in K(d_j)$.

Let the number of results in $RC(Q(u,q))$ be represented as $\lambda$ such that $\lambda = |RC(Q(u,q))|$. Lets also denote the number of users in result candidates tuple list as $\lambda_v$ and the number of documents by $\lambda_d$. Also, lets assume the number of unique users in the above list as $\lambda_v'$.

**Definition 8. Result Set:** A result set, $RS(Q(u,q))$, for a query $Q(u,q)$ is defined as a set of $\rho$ document tuples $(v_i,d_j)$ such that $v_i \in F(u)$ and $\forall d_j, q \in K(d_j)$. Thus, for a query $Q(u,q)$ with result candidates, $RC(Q(u,q))$, the number of result sets possible is given by $\alpha = \lceil \frac{|RC(Q(u,q))|}{\rho} \rceil$.

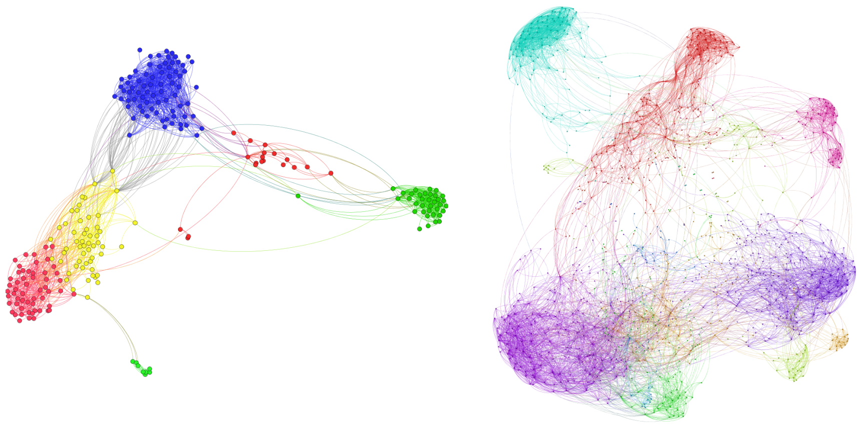
**Definition 9. Result Value:** The result value of a result set for a given $factor$ is defined as $RV(RS(Q(u,q)), \text{Factor})$. The method to compute the result value of a result set will vary according to the factor and will be described in section 5 along with each factor.

**Definition 10. Result Final:** The result final is a collection of result sets, ordered by decreasing result value. Thus, the result final for query $Q(u,q)$ can be defined as $RF(Q(u,q)) = \{RS_1(Q(u,q)), RS_2(Q(u,q)), .., RS_\alpha(Q(u,q))\}$ such that $RV(RS_1(Q(u,q)), \text{Factor}) \geq RV(RS_2(Q(u,q)), \text{Factor}) \geq ..RV(RS_\alpha(Q(u,q)), \text{Factor})$.

In the next section, we will discuss and define the semantics of social relationship to formalize contribution of each user as they impart social context to formulate the final result set.

## 4 Semantics of Social Relationships

There are multiple ways to understand the relationship between two users, $v$ and $w$, in a social network. The analysis can be based on the understanding of the social groups present in the graph or measures of centrality or the clustering properties of the social network graph. In the absence of ground truth data about the relationship shared between two users (in real or online life), in this work we explore multiple properties to analyze the social relationship between two users and provide a comprehensive study of factors that may have a potential impact on social search engine results.



(a) userA (b) userB

Fig. 3: Mutual friend network visualizations.

We base our analysis of the relationship between two users from the point of view of the user, $u$, performing a search query through the search engine. Thus, we analyze the relationship shared between users, $v$ and $w$, through the mutual friend network of the user $u$ i.e. $MF(u)$. For different users, $u_1$ and $u_2$ with respective mutual friend networks, $MF(u_1)$ and $MF(u_2)$ such that the graphs are distinct either in terms of topology or based on the number of users present in the network, the relationship shared between two users $v$ and $w$ where both $v, w \in MF(u_1)$ and $MF(u_2)$ may vary accordingly. We present example mutual friend network visualizations in Figure 3. The visualizations represents the mutual friend networks of 'userA' and 'userB' (details about the users and the network properties are mentioned in section 8.2), respectively, from their Facebook profile. The visualizations were created using the Gephi platform [3].

We empirically determine the social groups of a user's network by analyzing the mutual friend network of the user. In centrality based methods, we use the factors of degree, betweenness centrality and closeness centrality. We further explore clustering based methods, namely local clustering coefficient property, to determine the social relationship semantics between two users, $v$ and $w$. We present an example in Figure 4. Ego $e$ is connected to all the other nodes in the graph and shown using a broken line between the vertices and ego $e$. The mutual friend network of the ego $e$ is shown by the connected lines between the other vertices of the figure. We introduce the formal definition of each relationship characteristic and compare and contrast the merits of each property next.

A social group in the ego-network of user $u$ can be defined as a set of friends who are connected among themselves, share a common identity and represents a dimension in the social life of the user $u$. A social group can be defined in multiple ways. In this work, we base our definition on mutuality [37] and the formal definition is presented next.

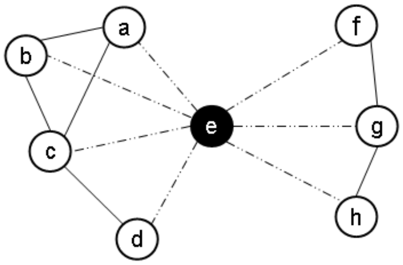


Fig. 4: Example ego-network of ego $e$

**Definition 11. Social Group:** A social group of a user $u$ is defined as $sg(u) = (V'')$ where $V''$ is a set of vertices such that $V'' \subseteq F(u)$ and for two users $v$ and $w$ in $V''$, $d(v, w) \leq k$ in the mutual friend graph, $MF(u)$. The set of all such social groups formed from the mutual friend graph of a user $u$ is represented as $SG(u)$.

### *4.1 Social Groups*

The above definition allows for duplication of users across different social groups since a user can belong to multiple social groups as it satisfies the geodesic requirement with other users of each group.

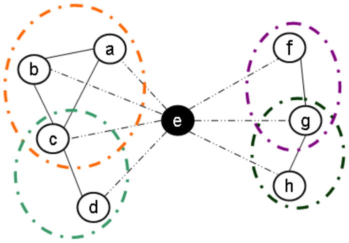


Fig. 5: Social Groups for an ego *e* at $k = 1$

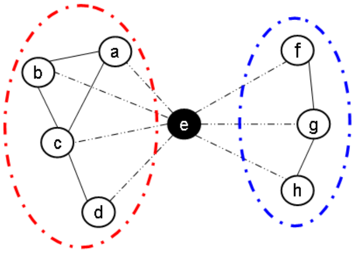


Fig. 6: Social Groups for an ego *e* at $k = 2$

Let user $u$'s social circle be divided into a set of groups represented as $SG_u = \{sg_u^i\}$, where $1 \leq i \leq ng_u$, $ng_u$ represents the number of social groups formed. Based on two different parameter values, examples of such groups are presented in Figure 5 and Figure 6. We observe that four social groups are discovered for $k = 1$. Nodes $c$ and $g$ overlap in both the two groups. Now, when we inspect the graph for $k = 2$, we discover only 2 social groups with no overlapping vertices. It is also important to note here that further increase in the value of $k$ has no effect in group generation. Thus, in a way the group formation gives a sense of separation or distance between the users based on the value of $k$ for the group formation process.

We use the set of all social groups formed from the mutual friend graph of an user $u$ to next define the social distance between two users present in the ego network of user $u$. Let user $v$ belong to the set of social groups $g_v$ such that $g_v \subset SG(u)$. Let $\eta_v$ represent the cardinality of $g_v$ and let each element of set of groups $g_v$ be represented as $g_v^i$ such that $1 \leq i \leq \eta_v$. We utilize the group member information to next define group distance and user distance.

**Definition 12. Social Group Distance:** The distance between two social groups is defined to be equal to the Jaccard distance between the groups. For two social groups, $sg(u)_i$ and $sg(u)_j$, from the set $SG(u)$ of user $u$, distance is defined as:

$$dist(sg(u)_i, sg(u)_j) = 1 - \left( \frac{|sg(u)_i \cap sg(u)_j|}{|sg(u)_i \cup sg(u)_j|} \right) \tag{1}$$

**Definition 13. User Distance in Ego Network:** User distance between two users, $v$ and $w$, in the ego network of user $u$ is defined as the mean distance between the two user's associated group(s). For users $v$ and $w$ associated with $\eta_v$ and $\eta_w$ number of social groups represented by $g_v^i$ and $g_w^j$ such that $1 \leq \eta_v$ and $1 \leq \eta_w$ respectively, user distance is defined as:

$$\omega(v,w) = \frac{\sum\limits_{\substack{1 \leq i \leq \eta_v \\ 1 \leq j \leq \eta_w}} dist(g_v^i, g_w^j)}{\eta_u \times \eta_w} \qquad (2)$$

The social group distance and user distance formula as proposed above paves the way for us to understand the social relationship between two users based on mutuality and creates scope for us to distinguish how distant (or close) users are to each other from the point of view of a single user. A high value in the user distance thus empirically suggests a separation (possibly to an extent of unfamiliarity) and furthermore existence of multiple facets to an individual's social life. For example, a typical individual has friends from their place of employment (which can be multiple and fairly distinct as individuals move through phases of professional career growth), place of education (with strong possibilities of multiple and distinct groups again as individuals go through high school, college, graduate school, etc.) and so on. The concepts related to social groups and multiple sections of a user's social network are analogous and the terms have been used interchangeably in rest of the paper. The 'diversity' factor as will be introduced in section 5.1 tries to capture the underlying hypothesis from the above discussion and helps build search engine results by exploiting the information present in a dormant format in social group information.

The semantics described next are more direct in this approach to capture social relationships and are used more explicitly to define respective factors and rank results in the social search engine.

## *4.2 Degree*

In this factor, we consider the degree of user $v$ in $MF(u)$ i.e. the factor that indicates the number of users in $F(u)$ connect to $v$. Let, this value be represented as $deg(v, MF(u))$, for all $v \in F(u)$. In the example of Figure 4, users $a, b$ and $g$ has a degree of 2, user $c$ has a degree of 3 and users $d, f$ and $h$ has a value of 1. The number indicates the strength of connectivity of a particular vertex in the mutual friend network. A high value can be interpreted as a signal of support for the friend and reflects their relative importance in $MF(u)$ and thus stands as an important signal to represent the social relationship shared between users.

While the value of degree (indegree and outdegree values in directional graphs) have been a signal of significant importance in graph based methodology developments, e.g. HITS, PageRank, in the context of social relationships and the mutual friend network of a user, the degree property can often formulate results to indicate biasness towards a few social relationships. For example, friends from a particular group (say place of work) can all know each other and can form complete graph, thus leading towards every user in the said group to have high and similar degree values and constraining the result set to include results from only one group. Other properties described next, e.g. betweenness, closeness centrality and clustering coefficient also tends to address these issues and thus, we believe 'diversity' offers a certain level of contrast to other social relationship characteristics and hence has the potential to offer interesting results in a social search engine result set.

### 4.3 Betweenness Centrality

The betweenness centrality characteristic of a node in a graph is used to quantify the extent to which a node lies between other nodes in the network [37]. The betweenness of a user $v$ is represented as $C_B(v, MF(u))$ for all $v \in F(u)$. The measure based on the connectivity of a node's neighbors, assigns a higher value for nodes that bridge clusters in the graph. The measure indicates the number of users that an individual user connects through to connect to other users in the graph and is another important signal to understand the semantics of social relationships.

The betweenness centrality of a node $v$ is computed as [7]: $C_B(v, MF(u)) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$ where $\sigma_{st}$ is total number of shortest paths from node $s$ to node $t$ and $\sigma_{st}(v)$ is the number of those paths that pass through $v$. In the example of Figure 4, users $a, b, d, f$ and $h$ has a betweenness centrality value of 0.0, user $c$ has a value of 0.13 and $g$ has a value of 0.067.

### 4.4 Closeness Centrality

The closeness centrality characteristic is a measure of how a node is central in a given network. The measure is defined as the sum of (geodesic) distance of a given node to all other nodes in the network [34, 7, 37]. Consequently, a user is termed as more central in the network if the total distance to all other users is lower relative to other user's respective value.

The closeness centrality for a user $v$ in the mutual friend network of user $u$ is defined as [7]: $C_C(v, MF(u)) = \sum_{t \in V \setminus v} 2^{-d_{MF(u)}(v,t)}$. In the example of Figure 4, users $a$ and $b$ has a closeness centrality value of 0.375, user $c$ has a value of 0.50, user $d$ has a value of 0.30, user $g$ has a value of 0.33 and users $f$ and $h$ has a value of 0.22.

### 4.5 Clustering Coefficient

The clustering factor captures the tendency of nodes to form a clique [38]. We particularly focus on the local clustering coefficient property of each user in the graph. For a user $v$ in the mutual friend network of user $u$, let the neighborhood of the user be defined as $N_{v,MF(u)} = w_i$ where $w_i$ is a user directly connected to user $v$ and $d(v,w_i) = 1$ in $MF(u)$. Lets define $k_v$ as the number of users, $|N_{v,MF(u)}|$, in the neighborhood, $N_{v,MF(u)}$ of user $v$ in $MF(u)$.

The local clustering coefficient of each user $v$ in the mutual friend network $MF(u)$ is defined as: $C_L(v,MF(u)) = \frac{2\times|e_{w_i,w_j}|}{k_v(k_v-1)}$ such that $w_i, w_j \in N_v$ and $e_{w_i,w_j} \in E'(u)$. In the example of Figure 4, user $a$ and $b$ has a local clustering coefficient value of 1.0, user $c$ has a value of 0.33 and users $d, f, g$ and $h$ has a value of 0.0.

The details of how the ranking algorithms satisfy the requirement of decreasing result value for each result set in result final are simple and intuitive and left to the reader. Based on the above factors to identify the social relationship semantics between two users, we next define the ranking factors and present the associated ranking algorithms to compute results in a social search engine.

## 5 Ranking Factors and Algorithms

In this section, we expand on our discussion of semantics of social relationships to introduce ranking factors. In the first subsection, we describe the ranking factors and also describe methods to evaluate the result value of any result set for a given ranking factor. In the final subsection, we talk about the ranking algorithm employed to rank results and determine the final result set(s) from the result candidates. We start by introducing the ‘diversity’ factor based on the definition of social groups as discussed in section 4.1.

### *5.1 Diversity*

The ‘diversity’ factor is based on the social group information of the querying user. The purpose of this factor is to maximize group representation in a result set such that the social diversity in a result set is maximized and a higher user distance between the users present in the result set can help user $u$ to inspect results that members from the various groups of the network share on the platform. The diversity value is based on the user-distance method defined in section 4.1 and is defined next.

**Definition 14. Diversity.** The diversity of a result set, $RS(Q(u,q))$, consisting of $\rho$ results is defined as the mean user distance(s) between each pair of users.

$$\triangle(RS(Q(u,q))) = \frac{\sum\limits_{v,w \in RS(Q(u,q))} \omega(v,w)}{|\rho|^2} \tag{3}$$

**Definition 15. Diversity Result Value.** The result value of a result set for the ‘diversity’ factor is defined as equal to the diversity value of the result set itself. Thus,

$$RV(RS(Q(u,q)), \text{`Diversity'}) = \triangle(u, RS(Q(u,q))) = \frac{\sum\limits_{v,w \in RS(Q(u,q))} \omega(v,w)}{|\rho|^2} \tag{4}$$

## 5.2 Degree

The ‘degree’ factor is based on the definition of ‘degree’ from section 4.2. The purpose of this factor is to select friends of the user performing a query who have the highest number of connections in the mutual friend network and define relevance in a social context as related to each contributing user’s popularity in the network.

**Definition 16. Degree Result Value.** The result value of a result set for the ‘degree’ factor is defined as the average of the degree value of all users present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{`Degeee'}) = \frac{\sum\limits_{v \in RS(Q(u,q))} deg(v, MF(u))}{\rho} \tag{5}$$

## 5.3 Betweenness Centrality

The ‘between centrality’ factor is based on the definition of betweenness centrality of individual users in the mutual friend network, section 4.3. The primary goal of this factor is to provide scope to build result sets such that users with highest values of betweenness centrality are ranked higher and provide relevancy to search engine results.

**Definition 17. Betweenness Centrality Result Value.** The result value of a result set for the ‘betweenness centrality’ factor is defined as the average of the betweenness centrality value of all users present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{`Betweenness Centrality'}) = \frac{\sum\limits_{v \in RS(Q(u,q))} C_B(v, MF(u))}{\rho} \tag{6}$$

## 5.4 Closeness Centrality

Similar to betweenness centrality, the ranking factor ‘closeness centrality’ is based on the definition of closeness centrality from section 4.4. The motivation here is to include results from users with higher values of closeness centrality in the top ranked result set.

**Definition 18. Closeness Centrality Result Value.** The result value of a result set for the 'closeness centrality' factor is defined as the average of the closeness centrality value of all users present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{'Closeness Centrality'}) = \frac{\sum\limits_{v \in RS(Q(u,q))} C_C(v, MF(u))}{\rho} \tag{7}$$

## 5.5 Clustering Coefficient

Clustering coefficient is introduced as a ranking factor based on the definition provided in section 4.5. The purpose here is include results from users with higher local clustering coefficients first and continue the process till all entries from result candidates are placed in result sets of decreasing value.

**Definition 19. Clustering Coefficient Result Value.** The result value of a result set for the 'clustering coefficient' factor is defined as the average of the clustering coefficient value of all users present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{'Clustering Coefficient'}) = \frac{\sum\limits_{v \in RS(Q(u,q))} C_L(v, MF(u))}{\rho} \tag{8}$$

## 5.6 Time

We introduce 'time' as the final factor to rank results. The time-stamp of each shared information, $T(d)$ is considered to rank the result candidates to compute the final result. In contrast to the previous factors that were based on the social relationship shared between users, the 'time' fator is established to reflect the most recent activity by users in the context of the query. For example, in the context of a query related to 'budget', the 'time' factor can successfully determine search results that link to the most recently shared information related to 'budget'.

**Definition 20. Time Result Value.** The result value of a result set for the 'time' factor is defined as the average time-stamp of the information set present in the result set. Thus,

$$RV(RS(Q(u,q)), \text{'Time'}) = \frac{\sum\limits_{d \in RS(Q(u,q))} T(d)}{\rho} \tag{9}$$

In addition to the above definition of a result value for the factor 'time', we also measure the standard deviation in time-stamp values of the information set present in the result set. The standard deviation value helps us understand the extent of 'freshness' or 'real-time' nature of the results. In the next section, we discuss the algorithms employed to compute final result set for each ranking factor.

## 5.7 Ranking Algorithms

The ranking algorithm generates the set of final results, $RF(Q(u,q)) = \{RS_1(Q(u ,q)), RS_2(Q(u,q)), .., RS_\alpha(Q(u,q))\}$ from the set of result candidates, $RC(Q(u,q))$. The steps associated with the ranking algorithms for each ranking factor are described next. We will start by recounting the terminologies associated with the set of result candidates, $RC(Q(u,q))$. The number of results in $RC(Q(u,q))$ is represented as $\lambda$, i.e. $\lambda = |RC(Q(u,q))|$. Also, the number of users in result candidates tuple list is denoted as $\lambda_v$ and the number of documents by $\lambda_d$. The number of unique users in the above list is assumed as $\lambda_v'$. A result set, $RS(Q(u,q))$, contains $\rho$ tuples of information.

Each information has a time-stamp data marked by $T(d)$. If a user has shared multiple pieces of information, the information set is sorted by the time-stamp, $T(d)$. The most recently shared information is ranked highest followed by information shared at later dates. The algorithm associated for 'diversity' factor is described next:

### 5.7.1 Diversity:

The result value for the 'diversity' factor is based on the relationship shared between two users (user distance property) present in the result set. The steps involved in the ranking algorithm for 'diversity' are described next.

1. If the number of result candidates is less than or equal to the size of a result set, i.e. if $\lambda \leq \rho$, then only one result set is possible and $RF(Q(u,q)) = RC(Q(u,q))$.
2. If the number of result candidates is greater than the result size set and the number of unique users is equal to the result set size, i.e. if $\lambda > \rho$ and $\lambda_v' = \rho$, then $RS(Q(u,q))$ is constructed using the most recently shared post (using information from $T(d)$) of $\lambda_v'$ users. This automatically ensures that maximum value of diversity is achieved in the result set. If the starting condition of result candidates processing is this step, then the result set becomes the first result set of the final result set, i.e. $RS_1(Q(u,q))$. Now, $RC_{\text{new}}(Q(u,q)) = RC(Q(u,q)) - RS_1(Q(u,q))\}$. The values related to $\lambda$ and $\lambda_v'$ are updated accordingly and in the next iterations to construct result set $RS_2(Q(u,q)), ..., RS_\alpha(Q(u,q))$, the applicable steps are followed.
3. If the number of result candidates is greater than the result size set and the number of unique users is less than the result set size, i.e. if $\lambda > \rho$ and $\lambda_v' < \rho$, $\binom{\lambda}{\rho}$ possible result sets are constructed and using the user information available in

each result set, result value for the 'diversity' factor is computed. The result set with the highest value of diversity is selected and $RC(Q(u,q))$ is updated to repeat the steps to compute next set of results. A user may contribute multiple times in the result set but the process ensures that the result set has the highest value of diversity. In the case of multiple result sets with equal value of 'diversity', knowledge about time-stamps of each shared information included in the result set is used to break the tie and the result set with the highest value of time-stamp (i.e. the result set with the most recently shared documents) is selected as the result.

4. If the number of result candidates is greater than the size of a result set and the number of unique users is also greater than the result set size, i.e. if $\lambda > \rho$ and $\lambda'_v > \rho$, we start by first constructing $\binom{\lambda'_v}{\rho}$ number of sets and compute the diversity value of each set. The set with the highest value of diversity is selected and documents associated with each user is selected to formulate the result set. The most recently shared document by users are used and in case of tie in diversity values, time-stamp values are used to break the tie and the set of most recently shared documents are declared as winner. The set of result candidates, $RC(Q(u,q))$, is updated and the steps are repeated till the set of result candidates has no more entries.

Based on the relationship shared between two users in a result set, the algorithm to rank results for the 'diversity' factor contrasts the corresponding ranking algorithm of other factors. Algorithm for other factors are presented next.

### 5.7.2 Degree, Betweenness Centrality, Closeness Centrality and Clustering Coefficient:

The algorithm to rank results for 'degree', 'betweenness centrality', 'closeness centrality' and 'clustering coefficient' factors is similar in nature and the steps are described next:

1. If the number of result candidates is less than or equal to the size of a result set, i.e. if $\lambda \leq \rho$, then only one result set is possible and $RF(Q(u,q)) = RC(Q(u,q))$.
2. If the number of result candidates is greater than the result size set i.e. if $\lambda > \rho$, the results are ordered by the respective value (degree, betweenness centrality, closeness centrality or clustering coefficient value) of each user present in $RC(Q(u,q))$ and the user with highest value is ranked first. Multiple entries by a user of higher value are placed in the final result set before entries from a user with lower degree value are considered.

### 5.7.3 Time:

The algorithm to rank results based on the 'time' factor is the simplest among all the factors. The set of information present in $RC(Q(u,q))$ is ordered according to their time-stamp value. The document shared most recently is ranked first followed by documents in decreasing value of time-stamp. The ordered set is finally used to construct $\alpha$ results sets and the final result set, $RF(Q(u,q))$.

This concludes our discussion on the ranking factors and the associated algorithms. In the next section, we discuss details about the implementation of the social search engine.

## 6 Social Search System Development

We built *InfoSearch* as a prototype social search engine over Facebook. *InfoSearch* is built as a Facebook application using the Facebook platform APIs and is available at http://apps.facebook.com/infosearch. Users are requested to authorize the application in order to use it. Once authorized, the three primary components of the application work together to deliver search results. The system architecture for the search engine is presented in Figure 7 and the components are described next.

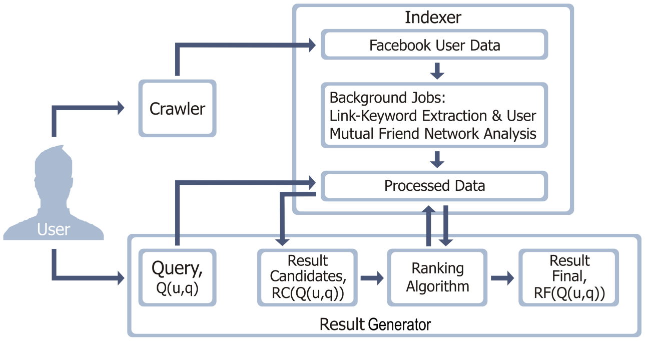


Fig. 7: Social Search Engine Architecture

### 6.1 Crawler

The purpose of the Crawler is to pull out information from the Facebook feed of each signed-in user using the Facebook API. The Facebook feed of a user consists of links, photos, and other updates from friends. In this work, the Crawler focuses on crawling the shared links to connect the web graph with the social graph. The Crawler is executed on a daily basis for each authorized user to retrieve the following data from their feed.

In our work, the Crawler employs the ‘links’ API provided by Facebook to crawl the various ‘links’ i. e. internet URLs shared by users on the Facebook platform. When called by the Crawler, the ‘links’ API returns a set of fields related to each link entry. Among the returned fields, we consider the following fields: a) ‘id’, b) ‘from’, c) ‘link’, d) ‘name’, e) ‘description’, f) ‘message’ and g) ‘created_time’ for the next component of our search engine. The Crawler also retrieves information about a user’s friend list to build the ego and mutual friend network of a user. The Crawler uses the ‘friends’ and ‘friends.getMutualFriends’ API to retrieve information about the nodes and edges, respectively to build the ego network of a user. The Crawler also provides scope to expand our architecture to include other social network platforms by mapping the field lists of each returned link with fields used by the next two components of the architecture.

## *6.2 Indexer*

The Indexer has two primary tasks. First, it analyzes the information retrieved by the Crawler to build an index of keywords for each shared URL. Second, the Indexer also performs the task of analyzing the mutual friend network of each user and build the corresponding user relationship data. Details of each task are described next.

Once the shared URLs are retrieved from the feed of each signed-in user, the next step is to build a keyword table for each URL with keywords extracted from the text retrieved from the URL. We use Yahoo!’s term extraction engine [26] for this purpose. The term extraction engine takes a string as input and outputs a result set of extracted terms. Additionally, we also use the Python-based topia.termextract library [21] to expand the keyword table. This library is based on text term extraction using the parts-of-speech tagging algorithm. We retrieve text from each URL and interpret the text using the aforementioned methods to finalize the set of keywords for each shared link. The second task of the Indexer is to analyze each signed-user’s mutual friend network and determine the user property information (i.e. values of degree, betweenness centrality, etc) of each friend in the network. We use the ‘R’ implementation of ‘kCliques’ to build the social group information set [36].

To understand the impact of $k$ in social group formation and accurate construction of social groups as users interact with *InfoSearch*, we built a Facebook application and surveyed users response for different values of $k$. We varied the value of $k$ between 1 and 5 and asked users for their thoughts on the accuracy of social groups formed at different values of $k$. Conclusions from user responses were then used to determine the appropriate value of $k$ for final result formulation in *InfoSearch*. In our current implementation, we use a value of $k$ equal to 3 to generate results for queries. We discuss details of this application and user feedback in Section 8.1.

The Indexer accomplishes the above two tasks by running data analytics background jobs on the raw data crawled from Facebook. The final processed data subsequently interacts with the third and final component, the result generator which is described next.

### 6.3 Result Generator

This is the final component in the system development. The purpose of this component is to a) process the user input query(ies), b) determine the result candidates, and c) formulate the final result set. In the first step, the user enters a query through the search engine web interface. At this step, users are also given the option to select their preferred way of ranking the possible results. In the next step, all documents related to the input query that originated from the friends of the user are retrieved. If no related documents are found and the query includes multiple keywords, the query is broken into multiple sub-queries and the search process is repeated to determine the related documents. If no documents are found at this stage, a 'no results found' message is sent to the user and the process stops. Otherwise, the set of related documents are promoted to potential result candidates and sent for processing by the ranking algorithms to determine the final result set. Based on the ranking factor selected by the user, the corresponding ranking algorithm is applied to the result candidates and the final result set is pushed forward to the application interface for display to the user.

In our current implementation, we set the number of results per result set, i.e. $\rho$ as equal to 8. We implement a pagination style such that every result set of $\rho$ results, i.e. $RS_1(Q(u,q)), RS_2(Q(u,q)), ..., RS_\alpha(Q(u,q))$ are placed on consecutive pages. Thus, the result sets are displayed to the user in the form of consecutive pages such that the first page displays the result set with highest value and decreases on later pages.

It is important to emphasize on the computational complexity involved in the final result construction at this stage. In traditional web search engines, final results for a variety of query keywords are pre-computed and result sets are cached for delivery to the user. In contrast, in a social search engine, as the number of result candidates and social context information present for each query varies, a result construction on the fly becomes a necessity and offers significant challenges to develop efficient and fast solutions. For example, during the process of determining final result sets using the 'diversity' factor, the number of sets possible for $\lambda'_v$ unique users in the result candidates is $\binom{\lambda'_v}{\rho}$. The number of potential result sets for a relatively small number of unique users, say $\lambda'_v = 16$, the number of sets possible is $\binom{16}{8} = 12,870$. This number increases exponentially for higher number of users in result candidates. Iteration through such a large number of possible result sets takes a considerable amount of time and renders the search experience slow and inefficient. In the current development phase of *InfoSearch*, we focus on highlighting the challenges of building social search engines and leave exploration of efficient algorithms for future works. However, as we will see during our user case studies in Section 8.2, Table 3, as the number of unique users in result candidates for a query can be substantially high, we resorted to using heuristic methods during the development process. In the final result set construction step, if the number of results, $\lambda$, and number of unique users, $\lambda'_v$, are both greater than the size of a result set, $\rho$, we consider the most recent 12 results sorted by 'time' and originating from 12 different users as constituent of the starting result candidates to construct the first final result set, introducing the next 8 results into result candidates list in addition to the remaining 4 results to generate the second final result set and so on. This step ensures we only have to construct $\binom{12}{8} = 495$ possible result sets before we decide the final result set at each iteration and users can enjoy the experience of receiving a quick result set for their query.

We also implement an additional feature to help users find information related to a specific friend or set of friends. This feature is implemented at the query step and the user has to specify the name of his/her friend(s) in conjunction with the query. In this particular situation, the retrieval process is limited to the set of information related to the specified user(s) only and the $time$ factor is used to rank the results at this step. In the following section, we discuss the deployment of *InfoSearch* and present a few statistics on its current usage and performance.

## 7 User Statistics

We invited colleagues from our lab to use the application. *InfoSearch* was made available in March 2011. We present the following statistics analyzing the usage between March and December 2011. *InfoSearch* gained 25 signed-in users and through the signed-in user's Facebook feed, it has access to regular updates of $5,250$ users. Each user has an average of 210 users in their ego network and their mutual friend graph has an average of 1414 edges.

During the time *InfoSearch* has been active, we have crawled links shared by $3,159$ users. This is a very significant number because it tells us that, among the users *InfoSearch* has access to, $60\%$ shared a web link with their friends in the social network. It is evident that the integration of web and social network graphs is taking place at a rapid pace and that the growth can have a significant impact on the way users search for information on the Internet.

The number of links shared by the users during this period is $31,075$. The number of keywords extracted using the Yahoo! term extraction engine and the Python topia.termextract library is $1,065,835$, which amounts to an average of 34 terms for each link. Additionally, we also consider the number of unique terms present in this pool to form a picture about the uniqueness in the shared content. We observe that the number of unique terms shared across all the links is $130,900$, which results in an average of 4 terms per link. We next discuss case studies to understand the performance of social search engine results under different ranking factors and algorithms. We start by discussing results from our user study to determine the best value of $k$ to formulate social groups.

# 8 User Studies

## *8.1 Social Group Analysis*

An interpretation of the number and qualitative properties of social groups proposed by any method is a matter of subjective analysis to a particular user. In our definition of social groups, we mention the permissible upper-bound geodesic distance of $k$ for two users to be a part a social group in the ego network of a user. A variation in the values of $k$ can thus determine different social groups and consequently can lead to different (favorable or unfavorable) appreciation of the quantitative and qualitative properties of the social groups. To understand the value of $k$ at which the users feel the social groups formed are best representative of their social network, we built a Facebook application [1] and sought out user feedback. We next describe the details.

A user must approve an application before the application can interact with the user. Once a user $u$ approves the application to read their respective social data, information about their friends are fetched. In the second step, the fetched friend information is used to construct the mutual friend graph, $MF(u)$. Next, we construct social groups, $SG(u)$, starting with value of $k$ equal to 1. We display the group formed to the user and sought out their feedback on two questions. In the first question, we asked users their opinion on the number of groups formed. The answer scores and their corresponding labels were a) 5, 'Too Many' b) 4, 'Many' c) 3, 'Perfect' d) 2, 'Less' and e) 1, 'Too Less'. In the second question, we asked participants of their feedback on the quality of the groups formed i.e. if the social groups formed were accurate representation of their real life groups. To obtain feedback for this question, we provide the participants the following scores along with the corresponding labels: a) 5, 'Yes, Perfectly' b) 4, 'To a good extent' c) 3, 'Average, could be better' d) 2, 'Too many related friends in separate groups' and e) 1, 'Too many unrelated friends in the same group'. We repeat the above step by incrementing the value of $k$ for an upper limit of $k = 5$.

| | Feedback Value | Standard Deviation |
|---|---|---|
| $k = 1$ | 3.84 | 0.84 |
| $k = 2$ | 3.41 | 0.85 |
| $k = 3$ | 3.03 | 0.67 |
| $k = 4$ | 3.12 | 0.92 |
| $k = 5$ | 2.25 | 1.14 |

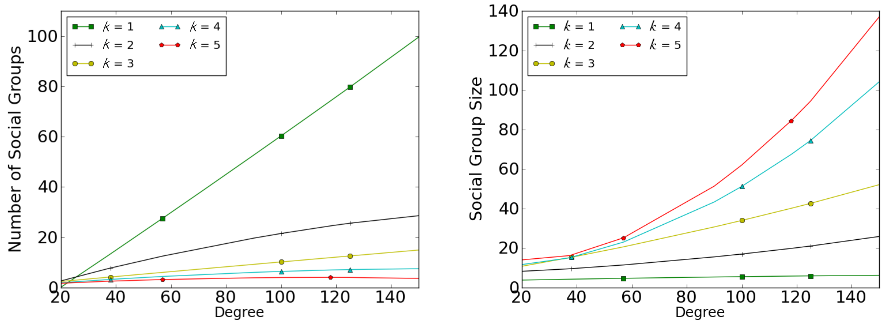
Table 1: User feedback scores on number of social groups detected

| | Feedback Value | Standard Deviation |
|---|---|---|
| $k = 1$ | 3.54 | 0.91 |
| $k = 2$ | 3.41 | 1.28 |
| $k = 3$ | 3.80 | 1.31 |
| $k = 4$ | 3.31 | 1.38 |
| $k = 5$ | 2.88 | 1.46 |

Table 2: User feedback scores on quality of social groups detected

Thirty users with varying size of friend lists signed into the application. Measurements from the logged-in user's egocentric networks are presented in Figure 8.

[1] The application is available at http://apps.facebook.com/group_friends.



(a) Number of Social Groups

(b) Average Size of Social Groups

Fig. 8: Logged-in user dataset analysis

We present results on the number of groups formed along with the average size of the groups for varying values of $k$ for different user degrees in each of the figures. At $k = 1$, the number of groups formed grows linearly with the degree of the user. At higher values of $k$, we observe that the number of groups formed significantly drops with larger average group sizes. For example, at $k = 1$, number of groups is equal to 60 and average group size is equal to 5 for users with degree equal to 100. However, at $k = 2$, for the same users, the average size of the groups have risen to 15 while the number of groups has dropped to only 20. This happens because as we increase the value of $k$ and correspondingly relax the requirements of member inclusion into a group, higher number of members are included into a single group including overlapping members. However, the more interesting observation comes when we compare the values obtained for $k = 4$ and $k = 5$. Since, we allow overlaps to exist across groups, if certain users exist over multiple groups for a given $k$, when we would allow a larger $k$, this overlapping user would cause the groups to collapse into a single group. Contrary to this assumption, we see only small changes in the values observed for $k = 4$ and $k = 5$ than for changes in values observed for $k = 3$ and $k = 4$, indicating that members in the mutual friend graph exist in small clusters that can be separated from each other at a certain cutoff level; $k = 4$ in this case.

Scores from the feedback analysis for the above two questions are presented in Table 1 and Table 2, respectively. We see the feedbacks on the number of social groups formed at $k = 3$ is approximately equal to 3, a score indicating a ‘Perfect’ division of the egocentric networks of the users into how they perceive their own social relationships to be divided in real life. It is also interesting to note in this section that the standard deviation at this instance is the least of all the feedbacks received.

User feedbacks on quality of the social groups formed are presented in Table 2. It is interesting to note that at values of $k$ equal to $1, 2$ and 3, feedbacks indicate a score between ‘Average, could be better’ and ‘To a good extend’ indicating that the social groups detected are indeed accurate representation of how users perceive their friends to be members of different sections in the real life. We thus conclude that a value of $k$ equal to 3 is a good choice to compute social groups and form the basis of providing diversity based results to users in *InfoSearch* during any query.

## *8.2 Search Result Analysis*

A social search engine generates unique results for every user. The subjective nature of results make it pointless to qualitatively compare with results from other search engines that generate identical results for all users. Thus, we cannot evaluate the results shown by *InfoSearch* for a query based on the results obtained from other web search engines. Instead, we focus on analyzing the impact of the ranking factors in the final result set for different users. We ask the following question: If a result set, $RS_i(Q(u,q))$, was generated using a particular ranking factor, what will be the result value of this result set for other ranking factors and how will the result value hold against similar values of result sets generated by other ranking factors? For example, in Figure 9, we compare diversity values of result sets generated by each ranking factor. We start by computing the final result for a given ranking factor and it's respective ranking algorithm. Once the final result has been computed and ranked result sets are available, we also evaluate the result value of each such result set for other ranking factors. Thus, for the example in Figure 9, we start by building the final result from the available result candidates for each ranking factor (i.e. 'diversity', 'degree', etc.) using the corresponding ranking algorithm. Once the result sets are ready, we compute the 'diversity' result value of the result set to compare and contrast the values in Figure 9. We perform similar actions to evaluate and discuss the result values for other ranking factors between Figures 10 and 14.

We perform user studies based on the information shared in the ego network of two authors of this work. The first author is labeled as 'userA' and the second author is labeled as 'userB'. userA has 246 members in his ego network. The number of edges shared between the members are 2235, that is, an average of 9.08 edges per member. userB has 1129 friends and the number of edges between the members are 7071, that results in an average of 6.26 members. Furthermore, the average clustering coefficient of each of the networks is 0.606 and 0.431 for 'userA' and 'userB' respectively. Aided with the visualizations presented in Figure 3, it is evident from these statistics that the respective ego networks are very different in topological characteristics and our next step is to understand how the ranking factors impact the final result set formation. We compare the results based on how the result value of each ranking factor holds up against the other ranking factors. For each ranking factor, we start by computing the final result set, $RF(Q(u,q))$. In the following discussions, we discuss the result value for the result set ranked highest i.e. we discuss the attributes of $RS_1(Q(u,q))$. We consider two queries for the user study: 'budget' and 'privacy' because of their relevancy among a large number of users in the social network. We present statistics related to each query for both users in Table 3.

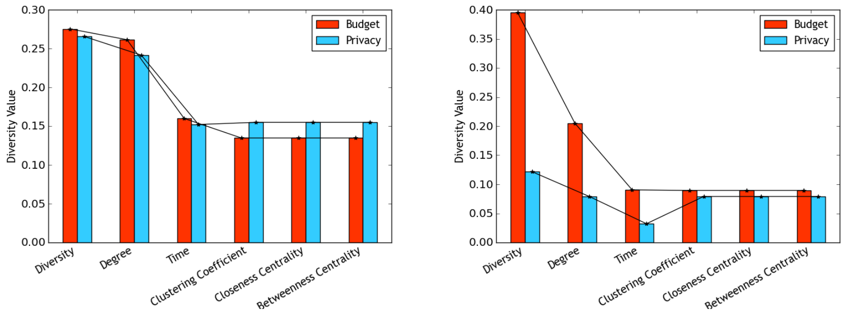
| | Query | Total number of results | Unique number of users sharing results |
|---|---|---|---|
| *userA* | 'Privacy' | 199 | 60 |
| | 'Budget' | 30 | 13 |
| *userB* | 'Privacy' | 1008 | 246 |
| | 'Budget' | 121 | 49 |

Table 3: Statistics on results candidates for query

The statistics in the table also illustrates the computation challenges to construct result set(s) in a social search engine setup as discussed in Section 6.3. In the above examples, the worst case scenario is to construct a result set of $\rho$ results from a possible result candidate of 1008 results originating from 246 unique users where we can construct $\binom{246}{8} = 2.96 \times 10^{14}$ sets to select the best result set. Clearly, this is a situation we want to avoid when we compute result for users on the fly. A consideration of this issue motivated us to exploit methods that will help us scale the computation and thus, finally in our result generation process, we consider only the 12 most recent result in the result candidate set to construct each result set. Next, we discuss the result values. We start by evaluating result values for the diversity factor.

Diversity result value of a result set is given by $RV(RS(Q(u,q)), \text{`Diversity'})$ and the values are plotted in Figure 9. The diversity values in the plot have been computed for $k = 3$. It is expected that the result sets produced using the diversity factor and it's corresponding ranking algorithms that aims to select the result set with the maximum value of diversity, has the highest values of diversity compared to the values of result sets generated by other factors. The plots confirm this hypothesis, however, it is interesting to note the difference in values of result sets computed using other factors. The consistency in decreasing values is best exemplified in the case of userB and query 'Budget'. userB's relatively large network (1129 friends) helps in retrieving results from a vast section of the network with high values of distance and corresponding diversity between the users. In contrast, diversity values for result sets formulated using the clustering and centrality measures are lowest in nature and shows signs of partiality in result formulation by contributions from only a few segments in the network.

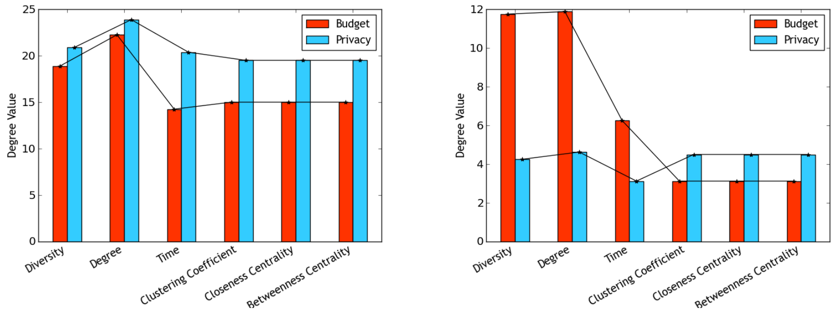
We also observe the lowest diversity value related to any result set in the case for the result set computed by 'time' factor. In the context of a large number of possible result candidates for query 'privacy' for userB, diversity value is only 0.03 compared to the diversity value of 0.12 for the result set determined by the diversity factor itself. Similar patterns can also be observed for query 'Budget', values of 0.09 and 0.39 for results ranked by time and diversity respectively. We infer from this observation that information once shared by a member in a social group, has a tendency to flow between the members of the particular social group before it is shared by members of other social groups. This leads us to believe that result sets formed based on time of sharing can lead to information sources that originate within particular social groups and will have the lowest social diversity value. While



(a) userA (b) userB

Fig. 9: Analysis for ranking factor ‘Diversity’.

the diversity based algorithm tries to maximize the value of social diversity in results, time factor, among the other factors mostly retrieve results that have the least value of social context present. We next discuss the degree values of result sets.

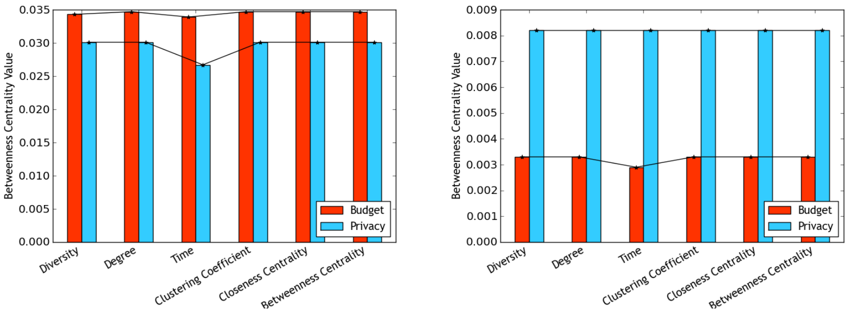


(a) userA (b) userB

Fig. 10: Analysis for ranking factor ‘Degree’.

The degree value of a result set is given by $RV(RS(Q(u,q)),\text{‘Degree’})$ and the values are plotted in Figure 10. Similar to results ranked by ‘diversity’ factor which were expected to generate result sets with the highest values of diversity among any of the factors, the ‘degree’ value is also expected to be the highest among all the result sets for the result set generated by the ‘degree’ factor and it’s corresponding ranking algorithm. The plots confirm the expectation. The values for queries ‘Budget’ and ‘Privacy’ for userA are 22.25 and 23.87 respectively compared to the second highest values generated by ‘diversity’ factor at 18.87 and 20.87, respectively.

Similar trends are also observed for userB in Figure 10b. However, it is surprising to notice the difference between values when compared to the values generated by the 'degree' factor. The values for query 'Budget' for factors 'time', 'clustering coefficient', 'closeness centrality' and 'betweenness centrality', 14.25, 15, 15, 15 for userA and 6.25, 3.25, 3.12, 3.12 for userB, respectively, are significantly lower while the 'diversity' factor, 18.87 for userA and 11.75 for userB, is able to relatively match up with the values of the 'degree' factor, 22.25 for userA and 11.875 for userB. The relative matching in the results is significant because although developed for a different reason, the 'diversity' factor is successful in capturing the essence of the 'degree' factor and provide comparable values for the 'degree' metric, thus showcasing itself as a strong candidate to power social search engine ranking algorithms.

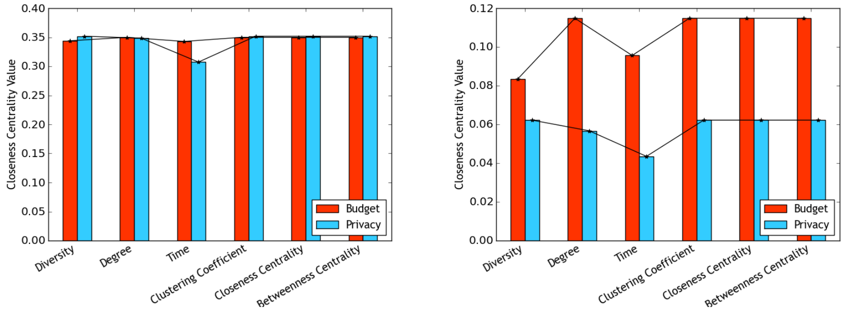


(a) userA (b) userB

Fig. 11: Analysis for ranking factor 'Betweenness Centrality'.

We next analyze the result values for the ranking factors based on centrality measures, i.e. 'betweenness centrality' and 'closeness centrality'. Analogous to the 'diversity' and 'degree' factors, result sets are also expected to have the maximum value of betweenness centrality and closeness centrality when the result sets were computed based on the respective factor and associated algorithm. We notice the phenomenon in the plots in Figures 11 and 12. Furthermore, we observe that the measures also generate similar result values for other factors. The highest value of betweenness centrality is observed to be 0.0345 for userA and 0.0033 for userB during analysis for query 'Budget' and 0.0301 for userA and 0.0082 for userB for query 'Privacy' the betweenness centrality factor (among other factors with equal values).

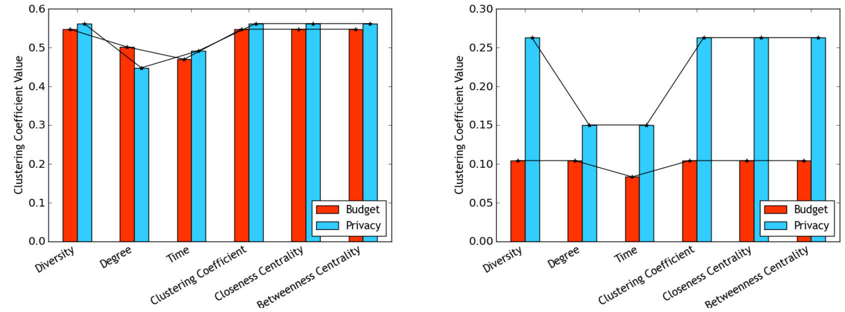
We see relatively low fluctuation in result values except for in the values generated by the 'time' factor based result set. The respective value for 'time' factor is 0.0339, 0.0029, 0.0267 and 0.0082, a percentage difference of 1.74%, 12.12%, 11.30% and 0%, respectively. This strengthens our previous argument that information has a tendency to flow between social groups before it spreads into a broader



(a) userA (b) userB

Fig. 12: Analysis for ranking factor 'Closeness Centrality'.

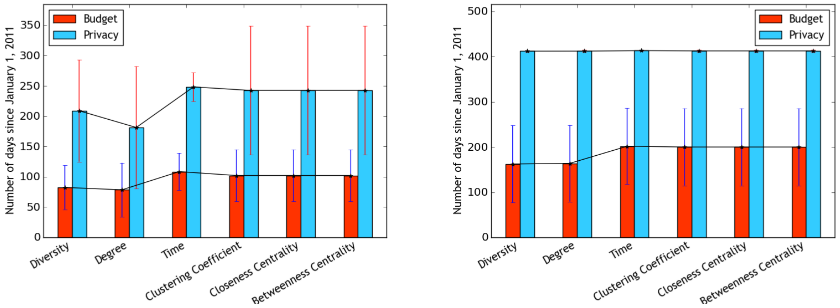
section of the ego network and a social search engine based solely on the 'time' factor thus fails to offer any advantage in terms of exploiting the prevalent social information. Next, we look at the 'clustering coefficient' result values. Unsurprisingly, we find a repeat of the same behavior here too with the 'time' factor offering the least value among all factors and failing to capture the social relationship based information into the result set. Finally, we investigate the 'time' characteristic of result sets.



(a) userA (b) userB

Fig. 13: Analysis for ranking factor 'Clustering Coefficient'.

We analyze time value of result sets from a reference date such that we can understand the relative 'freshness' of the data shared in the network. For example, if we observe two result set(s), we observe the average time-stamp value of shared information is 10 and 100 days in the future from the reference date, we term the result with the average time-stamp value of 100 days since the reference date to be more relevant and fresh to the user. Moreover, we also look at the standard deviation in the time-stamp values of the shared information and we term a result set with minimum values of deviation as the relevant result. Result values for the 'time' factor is presented in Figure 14.



(a) userA (b) userB

Fig. 14: Analysis for ranking factor 'Time'.

The reference point for 'time' value analysis is placed on January $1^{st}$, 2011 and the plots showcase number of days since the reference point. Thus, expectedly we observe the results generated based on 'time' factor has the maximum value compared to the respective value of result sets built using other factors. In the example of userA for query 'Budget', the value of result ranked using 'time' factor is 108 days whereas in contrast the lowest value is offered by the result set ranked by the 'degree' factor at 78 days. Furthermore, the corresponding deviation in the time-values are 30 days and 45 days respectively. Similar trends can also be observed in other cases. This happens because when results are ranked according to social relationship based factors, results that were shared a significantly long time ago are ranked higher in order to enrich the social value of the result set. Although not unexpected, a time based ranking of results thus, fails to accommodate social relationship semantics and provides a result set that is mostly partial to only a sub-section of the user's ego network. In the next section, we conclude our work with a discussion about future work.

## 9 Concluding Remarks

In this chapter, we described our efforts to build *InfoSearch* over the Facebook platform as a prototype social search engine and provide scope to users to search through the posts shared by their friends. In the process, we identified six important factors related to ranking search results for social search systems. Users can employ either one of the factors to rank results as they search through *InfoSearch*. Based on data collected through the Facebook feeds of two authors, we also performed user studies to understand the impact of ranking factors in the formation of result sets. We observed that 'time' based ranking of results, while providing the latest posts, fails to include sufficient social information in the result based on the value generated for both 'degree' and 'diversity' factors.

Among the factors based on semantics of social relationships between a user performing a query and a user sharing a piece of information, 'diversity' based factor provides sufficient social context into the result set as well as performs well in comparison to 'degree' factor to include time characteristics in the result set. We believe the area of social search engines has an immense potential in the area of information search and retrieval and we want to expand this work into multiple directions. First, we want to grow the usage of *InfoSearch* by inviting more users to use our system on a regular basis and provide us feedback on their opinion about the quality of results formulated. Second, we want to extend the system architecture to include the scope of distributed databases and develop the application into a distributed system capable of handling thousands of queries at any given time. Third, we want to extend the factors involved in the ranking process to include other online social network platform focused factors like 'interaction intensity between users'. Finally, we aim to develop methodologies and standards to objectively evaluate social search engine results.

## References

1. Adamic, L., Adar, E.: How to search a social network. Social Networks **27**(3), 187–203 (2005). DOI 10.1016/j.socnet.2005.01.007
2. Banerjee, A., Basu, S.: A social query model for decentralized search. In: Proceedings of the 2nd Workshop on Social Network Mining and Analysiss. ACM, New York, vol. 124 (2008)
3. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An open source software for exploring and manipulating networks (2009)
4. Baumes, J., Goldberg, M., Krishnamoorthy, M., Magdon-Ismail, M., Preston, N.: Finding communities by clustering a graph into overlapping subgraphs. In: International Conference on Applied Computing (2005)
5. Baumes, J., Goldberg, M., Magdon-Ismail, M.: Efficient identification of overlapping communities. In: IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 27–36. Springer (2005)
6. Borodin, A., Roberts, G.O., Rosenthal, J.S., Tsaparas, P.: Link analysis ranking: algorithms, theory, and experiments. ACM Transactions on Internet Technology **5**(1), 231–297 (2005). DOI 10.1145/1052934.1052942
7. Brandes, U.: A faster algorithm for betweenness centrality. Journal of Mathematical Sociology **25**, 163–177 (2001)
8. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems **30**(1-7), 107–117 (1998)
9. Cross, R., Parker, A., Borgatti, S.: A bird's-eye view: Using social network analysis to improve knowledge creation and sharing. IBM Institute for Business Value (2002)

10. Davitz, J., Yu, J., Basu, S., Gutelius, D., Harris, A.: iLink: search and routing in social networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 931–940. ACM (2007)
11. Derényi, I., Palla, G., Vicsek, T.: Clique percolation in random networks. Physical review letters **94**(16), 160,202 (2005)
12. Dhyani, D., Ng, W.K., Bhowmick, S.S.: A survey of Web metrics. ACM Computing Surveys **34**(4), 469–503 (2002). DOI 10.1145/592642.592645
13. Dodds, P.S., Muhamad, R., Watts, D.J.: An Experimental Study of Search in Global Social Networks. Science **301**(August), 827–829 (2003)
14. Facebook: Introducing facebook graph search. Available at https://www.facebook.com/about/graphsearch (2013)
15. Fortunato, S.: Community detection in graphs. arXiv **906** (2009)
16. Girvan, M., Newman, M.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences **99**(12), 7821 (2002)
17. Gregory, S.: A fast algorithm to find overlapping communities in networks. In: ECML/PKDD. Springer (2008)
18. Gregory, S.: Finding Overlapping Communities Using Disjoint Community Detection Algorithms. In: Complex Networks, pp. 47–61. Springer (2009)
19. Haynes, J., Perisic, I.: Mapping search relevance to social networks. Proceedings of the 3rd Workshop on Social Network Mining and Analysis - SNA-KDD '09 **09**, 1–7 (2009). DOI 10.1145/1731011.1731013
20. Horowitz, D., Kamvar, S.D.: The anatomy of a large-scale social search engine. Proceedings of the 19th international conference on World wide web - WWW '10 p. 431 (2010). DOI 10.1145/1772690.1772735
21. Index, P.P.: Content term extraction using pos tagging. Available at http://pypi.python.org/pypi/topia.termextract/ (2011)
22. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09 p. 467 (2009). DOI 10.1145/1557019.1557074
23. Marsden, P.: Egocentric and sociocentric measures of network centrality. Social Networks **24**(4), 407–422 (2002)
24. Mike Cassidy, M.K.: An update to google social search. Available at http://googleblog.blogspot.com/2011/02/update-to-google-social-search.html (February 17, 2011)
25. Mislove, A., Gummadi, K., Druschel, P.: Exploiting social networks for internet search. In: 5th Workshop on Hot Topics in Networks (HotNets06). Citeseer, p. 79 (2006)
26. Network, Y.D.: Term extraction documentation for yahoo! search. Available at http://developer.yahoo.com/search/content/V1/termExtraction.html (2011)
27. Newman, M.: Detecting community structure in networks. The European Physical Journal B-Condensed Matter and Complex Systems **38**(2), 321–330 (2004)
28. Newman, M.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences **103**(23), 8577 (2006)
29. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026,113 (2004). DOI 10.1103/PhysRevE.69.026113
30. Palla, G., Barabási, A., Vicsek, T.: Quantifying social group evolution. Nature-London-**446**(7136), 664 (2007)
31. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814 (2005)
32. Plangprasopchok, A., Lerman, K.: Exploiting social annotation for automatic resource discovery. In: AAAI workshop on Information Integration from the Web (2007)
33. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Sciences **101**(9), 2658 (2004)
34. Sabidussi, G.: The centrality index of a graph. Psychometrika **31**(4), 581–603 (1966). URL http://www.springerlink.com/index/10.1007/BF02289527

35. Tyler, J., Wilkinson, D., Huberman, B.: Email as spectroscopy: Automated discovery of community structure within organizations. In: First International Conference on Communities and Technologies (2003)
36. Vince Carey Li Long, R.G.: Package rbgl. http://cran.r-project.org/web/packages/RBGL/RBGL.pdf (2011)
37. Wasserman, S., Faust, K.: Social network analysis: Methods and applications. Cambridge university press (1994)
38. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. Nature **393**(6684), 440–442 (1998). URL http://dx.doi.org/10.1038/30918
39. Wingfield, N.: Facebook, microsoft deepen search ties. Available at http://online.wsj.com/article/SB10001424052748703421204576327600877796140.html (May 16, 2011)