A Ranking Algorithm for Online Social Network Search

Conference Paper · August 2013

DOI: 10.1145/2522548.2523134

A Ranking Algorithm for Online Social Network Search

Divya Sharma
IIM Calcutta
D.H. Road, Joka
Kolkata 700104, India
divyas12@iimcal.ac.in

AKZ Alam
IIM Calcutta
D.H. Road, Joka
Kolkata 700104, India
abula12@iimcal.ac.in

Parthasarathi Dasgupta
IIM Calcutta
D.H. Road, Joka
Kolkata 700104, India
partha@iimcal.ac.in

Debashis Saha
IIM Calcutta
D.H. Road, Joka
Kolkata 700104, India
ds@iimcal.ac.in

ABSTRACT

Online Social Networks have become the new arena for people to stay in touch, pursue their interests and collaborate. In October 2012, Facebook reported a whopping 1 billion users which is testimony of the fact how online social networks have proliferated and made inroads into the real world. Some of the obvious advantages of social networks are (1) 24X7 availability allowing users to stay in virtual touch at any time of the day, (2) get in touch with people who have similar interests and collaborate with them and (3) the ability to be able to search for users to add to one's friend circle. The third advantage is the focus of this paper. With the increasing number of users on online social networks, it is important that when a user searches for another user, appropriate results are returned. This paper identifies three criteria – proximity, similarity and interaction, which can be used to rank search results so that more appropriate results are presented to the searching user. Also, this algorithm allows the search ranking to be customized according to the nature of the online social network in question.

Categories and Subject Descriptors

H.3.5[**Information Storage and Retrieval]:** Online Information Services - *Web-based services*

General Terms

Algorithms, Management

Keywords

Algorithm, Online Social Network, Search Ranking.

1. INTRODUCTION

Social networks consist of sites that allow people to interact and share social experiences by exchange of multimedia objects (text, audio, and video) associated with the people themselves and their actions [7]. Every user on a social network possesses a user profile that contains all the information about the user ranging from basic information like name, date of birth, gender, location to more detailed information capturing educational and professional information and areas of interest. Online social networks have become abstractions of the real world where users interact, exchange and keep in touch. A major challenge, therefore, is the ability of a social network site to allow users to search for potential friends and in doing this, return relevant results. The need for such a search technique arises due to the inherent structure of social networks and the behavior of users on the network. Most searches on an online social network are queries containing names of users and a multitude of users may share the same name, which makes the trivial task of searching for friends very cumbersome. In order to overcome this challenge, it would make sense to rank the list of search results in order of decreasing relevance to the user searching for friends.

Here we discuss social network search ranking by means of an algorithm which takes into account three important factors that make search results relevant to a user – proximity, similarity and interaction. Based on these three factors, search ranks are allotted to search results when one user searches for another user by name. Also this algorithm takes into account the fact that all social networks are not of the same type. When a user searches for friends on a network of classmates, it is more relevant to rank results, which are closer to the searching user in the virtual space higher in the results list. On the other hand, on a social network for football fans, ranking must be done on the basis of sharing common interests like being fans of the same football club or the same football players. Still another example is a social network of acquaintances, where ranking on the basis of the level of interaction among users might be a more useful criteria.

Rest of the paper is organized as follows. Section 2 discusses some of the related recent works. Section 3 describes the proposed work including the definitions of the ranking metrics used, the proposed rank function, and the proposed method. Section 4 summarizes the empirical results and Section 5 highlights some of the real-life applications of the proposed method. Finally, Section 6 gives the concluding remarks and discusses scopes for future works.

2. RELATED WORK

Recently, there has been lots of interest in the field of online social network search and ranking. The work in [7] focuses on the problem of how to improve the search experience of the users. It suggests seed based ranking instead of text-based ranking by measuring shortest distances between the nodes in a friendship graph. This is the first work that makes use of the friendship graph in a big social network to improve the search experience. The paper in [1] presents a novel social search model for finding a friend with common interests in OSN (Online Social Network) with the introduction of trust value and popularity value. The trust value is calculated by the improved shortest path algorithm with a trust threshold, and the popularity value is obtained by the page rank algorithm iteratively. In order to ensure more accurate search results, [3] demonstrates an algorithm called E.L.I.T.E. which has five essential components - Engagement-U, Lifetime, Impression, Timeframe and Engagement-O. Engagement-U is the affinity between users which is measured by their relationships and other related interests between them, Lifetime is a trace of users' past based on their positive, neutral and even negative interactions and

Compute '13, August 22 - 24 2013, Vellore, Tamil Nadu, India
Copyright 2013 ACM 978-1-4503-2545-5/13/08…$15.00.
http://dx.doi.org/10.1145/2522548.2523134

actions with other users, Impression is the weight of each object determined by the number of positive responses from users, Timeframe is the timeline scoring technique in which an object naturally loses its value as time passes and Engagement-O is the attraction of users to objects which is measured between objects and associated interests of users.

Emphasizing on tree based search techniques, [4]compares the efficiency of reliable searching between Maximum Reliable Tree (MRT) algorithm and Optimum Branching Tree (OBT) algorithm and proposes the use of the MRT algorithm that is newly developed based on a graph-based method, as a generic technique which facilitates effective social network search and that can be the most reliable social network search method for the promptly appearing smart phone technologies. [5] explores the correlation between preferences of web search results and similarities among users by presenting an efficient search system called SMART finder. The work provides more information about SMART finder and publishes a quantitative evaluation of how SMART finder improves web searching compared to a baseline ranking algorithm. The concept of user tag feedback scores is employed in [2]. Based on this concept, a tag-based feedback web ranking algorithm is designed. The algorithm can efficiently use the user's feedback.

Searching for another person is one of the most fundamental uses to which social networks are put. However, most of the research work until now does not specifically look into this area. A small quantum of research work which does delve into this topic either provides just a conceptual framework or does not discuss the implementation of the framework in detail.

Considering that online social networks are an abstraction of real-world social networks, it is important to understand the factors that influence the associations between the users. From experience it can be said that one person's association with another person is a function of how closely linked the two users are, how many interests they share and how often they interact. We call these three factors proximity, similarity and interaction respectively. Though this is not an exhaustive list of factors that might influence association between two persons, it does give a fairly good idea about the same. This work tries to use this concept of association between users to rank search results when a particular user searches for another user on an online social network.

When any framework is to be implemented, it is required that the various factors which are being considered are quantified. The proposed algorithm in this work provides such a means to quantify the factors like proximity, similarity and interaction and then combine them to give a value for association between two persons. Along with providing a conceptual framework, this work also details out the implementation of the framework in practice.

3. THE PROPOSED WORK

3.1 Ranking Metrics

This paper defines three metrics for the purpose of ranking search results:

3.1.1 Proximity

On a social network, a user may be linked directly and indirectly to millions of users. A social network is a myriad web of interconnections and a node which is closer to the searching node in terms of number of hops is likely to be a better search result in comparison to a node which is several hops further away. *Proximity* measures the closeness of a node from the searching node. It is calculated by running a shortest distance algorithm and finding the minimum distance of the various nodes in the search result from the searching node. Let d_{ab} be the shortest distances between nodes *a* and *b*, then proximity p_{ab} is calculated as

$$p_{ab} = \frac{1}{1 + d_{ab}}$$

Proximity may also be calculated as (1 -d_{ab}/D), where D is the diameter of the social network, used to normalize d_{ab}. This would ensure the maximum value of proximity to be 1.

3.1.2 Similarity

Social networks provide users a platform for interacting with other users who share similar interests, listen to the same kind of music, read books from the same author, follow the same sport, share the same hobbies, etc. On a social network all these details are captured in the user profiles. When a user issues a search for another user, a user who is more similar to the searching user is likely to be a more relevant result in comparison to a user who is less similar. We define a user profile of user *a* as

$$P_a = \{k, l, m, n, \dots\}$$

where k, l, m, n … are the interests. Similarity S, between two nodes *a* and *b* in a list of search results with n nodes is defined as

$$S_{ab} = \frac{Cardinality(P_a \cap P_b)}{Cardinality(P_a \cup P_b \cup P_c \cup \dots \cup P_n)}$$

Alternatively, *similarity* may be defined in terms of the Jacquard similarity coefficient as

$$S_{ab} = \frac{Cardinality(P_a \cap P_b)}{Cardinality(P_a \cup P_b)}$$

In this case, we have considered the former definition of S_{ab} in order to define *similarity* in the context of all possible users' profiles. Though our definition of the S_{ab} might yield very small values of the metric, this would not affect the relative ranking of the users.

3.1.3 Interaction

Social networks provide users different means for interacting with one another. Most online social networks allow users to interact via textual *comments*, exchange links through *shares* and *like* the posts of other users. When two users interact on a social network, there are two factors that can be used to gauge the closeness of these two users – *frequency* of interaction and *recency* of interaction. Frequency captures volume of interaction (for each of comment, share, like) between two users within a given span of time. This span of time is defined by window size *w*, defined further in the discussion. Recency captures the time gap between the time of issuance of the search query and the most recent interaction (for each of comment, share, like) between the searching user and the user being searched. Frequency of an interaction of type T between user *a* and *b* is defined as

$$f_{ab}^{T_i} = 1 - \frac{1}{V_{ab}^{T_i}}$$

where T_i is the type of interaction (i = 1 for comment, i = 2 for share and i = 3 for like) and $V_{ab}{}^{Ti}$ is the volume of interactions between users *a* and *b* of type T_i. Recency of interaction between users *a* and *b* is defined as

$$r_{ab}^{T_i} = 1 - \frac{t_o - t_{ab}^{T_i}}{w^{T_i}}$$

where t_o is the time instance at which the search query was issued, t_{ab}^{Ti} is the time instance of the most recent interaction between user a and b of type T_i and window size w^{Ti} is defined as

$$w^{T_i} = \max\ (t_o - t_{ab}^{T_i}, t_o - t_{ac}^{T_i}, t_o - t_{ad}^{T_i}, \ldots, t_o - t_{an}^{T_i})$$

where users b, c, d, …, n are the search results of the query. The frequency and recency metrics are then used to define interaction i of type T_i between two users a and b as

$$i_{ab}^{T_i} = \alpha r_{ab}^{T_i} + (1 - \alpha) f_{ab}^{T_i}$$

where $0 \leq \alpha \leq 1$. α is defined as the relative importance of recency in comparison to frequency when quantifying a particular interaction.

The weighted interaction metric I between two users a and b is defined as the weighted sum of the three types of interactions (comment, share, like) as

$$I_{ab} = \beta i_{ab}^{T_1} + \gamma i_{ab}^{T_2} + \delta i_{ab}^{T_3}$$

where$\beta + \gamma + \delta = 1$. Here β, γ and δ define the percentage importance of the three types of interaction i.e. comment, share and like in the overall interaction metric I.

3.2 The Association Function

Once the three metrics *proximity, similarity and interaction* have been defined, the next step is to define the association between the searching user and the search results based on these three parameters. Association captures the effect of the three metrics in question and returns a composite value which describes how closely the user is associated with each of the search results. As discussed earlier, the importance of each of these metrics may vary according to the nature of the online social network. Therefore, weights are defined to give different weightage to each of these factors while calculating the rank of the search results.

The weighted association of a search result, b, being searched by user a is defines as

$$A_{ab} = \mu_1 p_{ab} + \mu_2 S_{ab} + \mu_3 I_{ab}$$

whereμ_1, $\mu_2$and μ_3 are the weights assigned to each of the metrics p_{ab}, S_{ab} and I_{ab}. It is important to note that

$$\mu_1 + \mu_2 + \mu_3 = 1$$

Therefore, μ_1, μ_2 and μ_3 can be defined as the percentage importance of proximity, similarity and interaction in calculating the association according to the nature of the social network. For example, in social network catering to football fans, where similarity is more important than proximity and interaction, μ_2 may have a higher value, while on a social network for classmates, proximity is more important, so, $\mu_1$may have a higher value.

3.3 The Rank Function

The ranks of the search results are subsequently obtained by sorting the results by the weighted association values. The search with the highest weighted association value is given rank 1, the search result with the second highest weighted association value is give rank 2, and so on, until the search result with the lowest weighted association value is given rank n.

3.4 Algorithm for Computing Search Ranks

The proposed algorithm is a ranking algorithm and, therefore, allows different search algorithms to be used to identify the unranked search results. Once the unranked search results are obtained, they are then ranked using the proposed algorithm. The association function is central to the calculation of the ranks of the search results.

We consider that n potential search results are returned by the searching algorithm for a given query. Once this list is obtained, the next steps are to calculate the association and then rank the list on the basis of the association values. The pseudo code for the algorithm is presented below. The variables and functions used in the pseudo code are explained in Table 1.

Table 1. Description of Variables/Functions

Variable/Function	Description
search_results	Array of unranked results of the search query
ranked_list	Array of ranked results of the search query
window_sizes	An array of window sizes for interactions of type comment (index 1), share (index 2) and like (index 3)
common_interests_ cardinality	Cardinality of the set of common interests of the searching user and the search results
association	Array of the association values of the search results
comment_recency_ window	Window size for comment recency
share_recency_window	Window size for share recency
like_recency_window	Window size for like recency
search()	Function to search potential results of a query
compute_ranks()	Function to rank the search results
get_window_sizes()	Function to compute the window sizes for interactions of type comment, share and like
get_common_interests_ cardinality()	Function to compute cardinality of the set of common interests
compute_association()	Function to compute association values of the search results
sort()	Function to sort the list of search results according to association values
get_proximity()	Function to calculate the value of proximity metric between two nodes
get_similarity()	Function to calculate the value of similarity metric between two nodes
get_interaction()	Function to calculate the value of interaction metric between two nodes
interaction_volume$_j$()	Function to return the number of interactions of type j between two nodes
recency_of_interaction()	Function to returns the recency of interaction between two nodes

Pseudo Code:

```
Initialize the network $N$, searching user $s$, search query $q$
rank_search_results($N$, $s$, $q$)
begin
    search_results[1 … $n$] ←search($q$, $N$)
    ranked_results[1 … $n$] ←compute_ranks($s$, search_results)
end
compute_ranks(s, search_results[1 … $n$])
begin
    window_sizes[1 … 3]←get_window_sizes($s$, search_results[1 … $n$])
    common_interests_cardinality←get_common_interests_cardinality(s,search_results[1 … $n$])
    association[1 … $n$] ←compute_association($s$, search_results, window_sizes, common_interests_cardinality)
    ranked_results[1 … $n$] ← sort(association[1 … $n$])
end
get_window_sizes(s, search_results[1 … $n$])
begin
    comment_recency_window←0
    share_recency_window←0
    like_recency_window←0
    for $i$ = 1 to $n$
        comment_recency_window←max(comment_recency_window, comment_recency$_{si}$)
        share_recency_window←max(share_recency_window, share_recency$_{si}$)
        like_recency_window←max(like_recency_window, like_recency$_{si}$)
    end
    window_sizes←[comment_recency_window, share_recency_window, like_recency_window]
end
get_common_interests_cardinality(s, search_results[1 … n])
begin
    common_interests←interests$_s$
    for $i$ = 1 to $n$
        common_interests←common_interests$\cup$interests$_i$
    end
    common_interests_cardinality←cardinality(common_interests)
end
compute_association(s, search_results[1 … $n$], window_sizes, common_interests_cardinality)
begin
for $i$= 1 to $n$
    proximity$_{si}$←get_proximity(s, search_results$_i$)
    similarity$_{si}$←get_similarity(s, search_results$_i$)/common_interests_cardinality
    interaction$_{si}$← get_interaction(s, search_results$_i$, window_sizes)
    association$_{si}$← $\mu_1$proximity$_{si}$+ $\mu_2$ similarity$_{si}$+ $\mu_3$interaction$_{si}$
end
end
get_proximity(s,i)
begin
    proximity ←1/(1+distance(s,i))
end
get_similarity(s, i)
begin
    similarity ←interests$_s$$\cup$interests$_i$
end
get_interaction(s, i, window_sizes)
begin
for j = 1 to 3
    frequency$_j$ ←1 – 1/interaction_volume$_j$(s,i)
    recency$_j$← 1 – recency_of_interaction$_j$(s,i)/window_size$_j$
    end
    frequency←$\beta$ frequency$_1$ + $\gamma$ frequency$_2$+ $\delta$ frequency$_3$
    recency←$\beta$ recency$_1$ + $\gamma$recency$_2$+ $\delta$ recency$_3$
    interaction ← $\alpha$ recency + (1 – $\alpha$) frequency
end
```

The *search* and *sort* functions in the above pseudo code are place holders for any searching and sorting algorithms that returns the list of search results based on the searching user's query and sort the search results on the basis of calculated association values respectively. Further details regarding the algorithm and its complexity have been described in [6].

4. EMPIRICAL RESULTS

4.1 The Framework

We propose to model a social network in the form of a graph, where the nodes correspond to the users and the edges correspond to the friendship between them. A part of one of the author's online social network was considered and anonymized for the purpose of this paper. Figure 1 shows this network, where a user *John* issues a search for another user *Maria*. The search query will return Maria *a*, Maria *b* and Maria *c*. The task now is to rank these results according to relevance to user John.

4.2 Simulation and Results

We now return to the framework described in Figure 1. The profiles of users we are interested in i.e. John, Maria$_a$, Maria$_b$ and Maria$_c$ are defined as follows:

$$P_{john} = \{k, m, n\}$$

$$P_{Maria_a} = \{k, m\}$$

$$P_{Maria_b} = \{m, n\}$$

$$P_{Maria_c} = \{k, l\}$$

The interactions between John and Maria$_a$, Maria$_b$ and Maria$_c$ are defined in Table 2.The ranking algorithm was then simulated by varying the weights of the different parameters α, β, γ, δ, μ_1, μ_2 and μ_3.

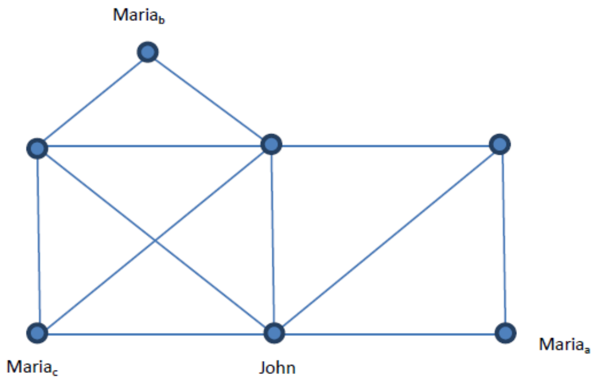

Figure 1. Model Social Network

4.3 Results and Comparisons

Some illustrative results of the proposed ranking algorithm for different weights are mentioned along with a comparison of search results from other popular social network sites and recent research work in this area in Table 3.

The small simulation network which was used for computing the results of the proposed algorithm exists on Facebook and Google+, in reality, for one of the authors. It was observed that the results obtained by issuing the same search on these networks can be replicated using the proposed algorithm by varying the various parameters, i.e., α, β, γ, δ, μ_1, μ_2 and μ_3. Table 3 is illustrative of this observation, as results at serial numbers 2, 5 and 6 give the same as those for Facebook and the result at serial number 1 is same as that of Google +. It is, therefore, critical to note that there may not be a unique set of parameters for ranking the search results in a particular fashion and the same search result ranking can be obtained by tuning the various parameters according to the requirements of the social network in which the algorithm is used. Though, this algorithm is not aimed at conjecturing the logic that might have been used to obtain search results by a social network website, it can, however, be used to obtain search results that concur with the results of the website in question to a certain extent.

The algorithm discussed in [2], based on trust and popularity was also implemented on the small simulation network. The result obtained was similar to that for Google+. Certain assumptions were made while implementing the algorithm in [2]. While calculating the contribution of trust in the rank, trust values for two adjacent nodes was taken as 0.5. For the calculation of the contribution of popularity in the rank, the damping factor *d* was taken to be 0.8 and the initial values of popularity were taken to be 0. The final values of popularity for each node with respect to the various keywords in the profile were obtained by running the circular algorithm until the values converged to a precision of 10%.

4.4 Scalability

The proposed algorithm was tested for scalability by varying the number of potential search results and measuring the time required for calculation of association function for the same. Table 4 shows selected results of the simulations. The time for execution with different number of potential search results was found to increase almost linearly. Figure 2 shows the plot of execution time versus the number of potential search results.

The important insight from this analysis is that calculation of the association function for as many as 1 million records is 3188 ms, which means that if we assume each user to have 1000 contacts in his/her network we can calculate the association of a user with other users up to two hops away in about 3 seconds. This can have important implications for the usefulness of the association function. Some likely usages to which the association function can be applied are discussed in a section below.

Table 2.Interaction between Nodes

User	**Comment**		**Share**		**Like**	
	Volume	**Most Recent**	**Volume**	**Most Recent**	**Volume**	**Most Recent**
Maria$_a$	3	05/08/2012	1	05/08/2012	12	18/09/2012
Maria$_b$	-	-	-	-	-	-
Maria$_c$	9	24/09/2012	10	18/07/2012	11	02/10/2012

Table 3. Selected Observations from Simulation

S.No	**Proposed Algorithm**							**Rank results**		
	α	β	γ	δ	μ_1	μ_2	μ_3	**Rank 1**	**Rank 2**	**Rank 3**
1.	0.5	0.5	0.3	0.2	0.34	0.33	0.33	Maria$_a$	Maria$_c$	Maria$_b$
2.	0.5	0.5	0.3	0.2	0.5	0	0.5	Maria$_c$	Maria$_a$	Maria$_b$
3.	0	0	0	0	0.5	0.5	0	Maria$_a$	Maria$_b$	Maria$_c$
4.	0	0	0	0	0	1	0	Maria$_a$	Maria$_b$	Maria$_c$
5.	0.5	1	0	0	0.5	0	0.5	Maria$_c$	Maria$_a$	Maria$_b$
6.	0	1	0	0	0.5	0	0.5	Maria$_c$	Maria$_a$	Maria$_b$
7.	**Facebook**							Maria$_c$	Maria$_a$	Maria$_b$
8.	**Google +**							Maria$_a$	Maria$_c$	Maria$_b$
9.	**Rank Algorithm Based on Trust and Popularity [2]**							Maria$_a$	Maria$_c$	Maria$_b$

Table 4. Selected Results of Simulation

No. of Potential Search Results	Execution Time (ms)
100	0.586108
500	2.867683
1000	6.002439
5000	33.4847
10000	61.12565
50000	174.8376
100000	323.0852
500000	1584.719
1000000	3188.202

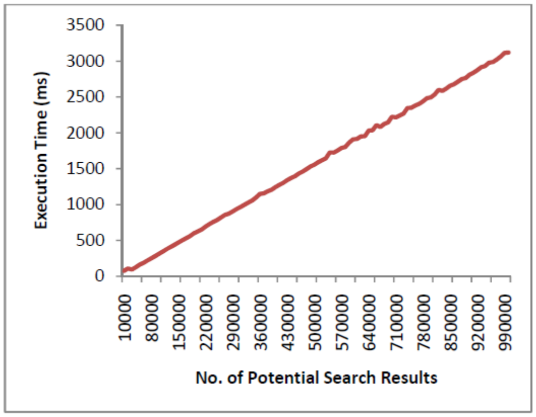

Figure 2. Execution Time vs. Potential Search Results

4.5 Observations and Discussion

The nature of the rank function defined by the proposed algorithm leads to some important observations:

1. $p \in (0,1]$

   Proximity by definition is the inversely related to the distance, *d*, between two nodes, where the distance means the number of hops to reach the destination node from the source node. Therefore,

   $$d \in N \Longrightarrow p \in (0,1]$$

2. $S \in [0,1]$

   Similarity is the ratio of the cardinality of the set of common interests between the searching user and a user in the search result and the cardinality of the set of the union of interests of the searching user and all the users in the search result. It is assumed that the profile of a user would contain at least one interest and a finite number of interests. Therefore,

   $$Cardinality(P_a \cap P_b) \in N$$

   $$Cardinality(P_a \cup P_b \cup P_c \cup \ldots \cup P_n) \in N$$

   $$\Longrightarrow S \in [0,1]$$

3. $I \in (0,1)$

   Interaction is defined as the weighted sum of the weighted sum of recency and frequency of a particular type of interaction. It is assumed that the window size, w, will always be a non-zero number. Therefore,

   $$recency \in [0,1)$$

   $$frequency \in [0,1)$$

   $$\beta,\gamma,\delta \in 0,1 \text{ subject to } \beta + \gamma + \delta = 1$$

   $$\Longrightarrow I \in (0,1)$$

4. $A \in (0,1)$

   Weighted association, A, is the weighted sum of proximity, similarity and interaction such that

   $$\mu_1,\mu_2,\mu_3 \in 0,1 \text{ and } \mu_1 + \mu_2 + \mu_3 = 1$$

   $$\Longrightarrow A \in (0,1)$$

The results from the proposed algorithm show that depending on the values set for the different parameters namely, α, β, γ, δ, μ_1, μ_2 and μ_3, suitable results can be obtained. The nature of a particular social network will dictate what values must be set for each of these parameters. The advantages of the proposed algorithm include:

a) *Intuitiveness:* The algorithm uses intuitive concepts like proximity, similarity and interaction to rank search results such that more relevant results may be ranked higher than less relevant ones.

b) *Adaptability:* The algorithm is not confined to providing suitable search ranks for any particular kind of network and can be adapted for use in any kind of social network by defining the various parameters according to the nature of the network

c) *Flexibility*: The algorithm can use the search results supplied by any search algorithm. Once the association values of the elements in the search list are obtained, subsequently, any algorithm can be used to sort the elements to obtain the ranked list. Therefore, the algorithm provides flexibility of implementation as it can be easily plugged between the search and sort algorithms in a social network.

5. APPLICATIONS OF THE PROPOSED WORK

Association, defined as a composition of proximity, similarity and interaction in this work, provides a means of quantifying the strength of linkage between two persons. This concept is not new, but, what is novel is how this can be used. Though this work is restricted to its use for search result ranking on social networks, it can be used in lots of other areas in the online space like marketing and advertising.

5.1 Use in Social Networks for Suggesting Friends

It is a common practice to suggest connections to a user on social network websites. The utility of this functionality in a social network is derived from its ability to suggest useful connections which might exist in a user's extended network and may have similar interests as those of the user. The association function can be used effectively in this regard and can identify potential connections for a user. Depending on the kind of social network, weights can be allotted to the factors of proximity and similarity and the corresponding association function values can be computed. Thereafter, the connections with association function values exceeding a pre-defined threshold can be suggested to a user. In this application, the factor of interaction is not considered as new connections are being suggested to a user with whom s/he does not have any previous communication history.

5.2 Use in Targeted Advertising and Marketing

Online advertisers can use this framework to identify prospects and then pursue them. An advertiser would be interested in finding people who might have similar affinities as their existing customers. The first step, for a given customer base would be to identify other people in the online social circles of these customers who have similar interests as the customers themselves. This can be done by calculating association values using the factor of similarity alone. Once prospects with high degree of similarity in interests as the existing customers have been found, the next step would be to identify people who may be easily influenced by the existing customers. For this, association values using both the factors of proximity and interaction can be computed. A prospect is likely to convert into a customer if s/he knows that the people who s/he is close to or often interacts with also use the product or service in question. The factors of proximity and interaction quantify this behavior of prospects and, therefore, association values on the basis of these two factors would serve to identify potential customers from the pool of prospects. In this way, targeted marketing and advertising can be carried out and the positive effects of word-of-mouth can made use of.

The association function can also be put to other innovative uses like identification of potential employees by job portals, recommendation for downloads of movies/songs, suggestions for online games, etc.

6. CONCLUSION

Online social networks have become an essential part of the lives of internet users. The importance of these networks will only grow over time and they are likely to become more and more complex and specialized as they evolve. The work presented in this paper tries to provide a generic solution for ranking of search results over social networks. Considering the large volume of searches being performed on social networks, the trivial function of providing relevant search results has become a differentiator for different social networks. This work takes into account the fact that all social networks are not similar and, hence, the same search result algorithm is not likely to be useful for all of them. As a result, an adaptive algorithm has been proposed which uses intuitive concepts like proximity, similarity and interaction to rank search results according to their relevance in a particular social network setting.

The proposed method has ample scope of improvement and future work. The ranking function may consider more aspects apart from those considered here. Considering the huge data volume associated with search through social networks, the proposed algorithm may have to be tuned appropriately for time-efficiency.

7. REFERENCES

[1] Huang, Chuan, Yinzi Chen, Wendong Wang, Yidong Cui, Hao Wang, and Nan Du. A novel social search model based on trust and popularity." In *3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT),* (2010), 1030-1034.

[2] Jiang, Zongli, and Jingsheng Li. A tag feedback based sorting algorithm for social search. In *International Conference on Systems and Informatics (ICSAI)*, (2012), 1482-1485.

[3] Lee, Khuan Yew, and Jer Lang Hong. ELITE—A novel ranking algorithm for social networking sites. In *9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),* (2012), 1009-1013.

[4] Lee, Wookey, JJ-H. Lee, JJ-S. Song, and CS-H. Eom. Maximum reliable tree for social network search. In *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC)*, (2011),1243-1249.

[5] Park, GunWoo, SooJin Lee, and SangHoon Lee. To Enhance Web Search Based on Topic Sensitive_Social Relationship Ranking Algorithm in Social Networks. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, 2009. WI-IAT'09, (2009), 3, 445-448.

[6] Sharma Divya, Dasgupta Parthasarathi, and Saha Debashis. A Ranking Algorithm for Online Social Network Search. In *Working Paper Series of Indian Institute of Management Calcutta*. (2013), WPS No. 724.

[7] Vieira, Monique V., Bruno M. Fonseca, Rodrigo Damazio, Paulo B. Golgher, Davi de Castro Reis, and Berthier Ribeiro-Neto. Efficient search ranking in social networks. In *Proceedings of the sixteenth ACM Conference on information and knowledge management*, (2007), 563-572.