How to Find Important Users in a Web Community? Mining Similarity Graphs

Clemens Schefels
Institute of Computer Science, Goethe-University Frankfurt am Main
Robert-Mayer-Straße 10, 60325 Frankfurt am Main
Email: schefels@dbis.cs.uni-frankfurt.de

***Abstract*—In this paper we provide a useful tool to the web site owner for enhancing her/his marketing strategies and rise as consequence the click rates on her/his web site. Our approach addresses the following research questions: which users are important for the web community? Which users have similar interests? How similar are the interests of the users of the web community? How is this specific community structured? We present a framework for building and analyzing weighted similarity graphs, e.g., for a social web community. For that, we provide measurements for user equality and user similarity. Furthermore, we introduce different graph types for analyzing profiles of web community users. We present two new algorithms for finding important users of a community.**

***Keywords*-Computer aided analysis; World Wide Web; Data analysis; Graph theory;**

I. INTRODUCTION

Nowadays, web-based user communities enjoy great popularity. Facebook[1] has more than 900 million members and even the relatively new Google+[2] about 170 million. In this highly competitive environment, it is crucial for web site owners to understand and satisfy their web community.

Previous research discovered community structures in these networks, but focused only on the pure friendship structure of these communities [1]. In this paper, we present a tool for building and mining similarity graphs. These similarity graphs are built from the interest profiles of the users of a web community. We use the Gugubarra framework [2], [3], developed by DBIS at the Goethe-University Frankfurt, to build interest profiles of web users.

In Gugubarra each user profile is stored as a vector that presents the supposed interests of a user u_m related to a topic T_i at time t_n. Each vector row contains the calculated interest value of the user for a given topic. The values of the interest are between 0 and 1, while 1 indicates high interest and 0 indicates no interest for a topic (see Figure 1). Gugubarra generates for each registered user several profiles:

A *Feedback Profile*, (FP), which stores the data explicitly given by a user. For that, we ask the users from time to time about their interests in respect to a set of predefined topics.

[1]https://www.facebook.com/
[2]https://plus.google.com/

A *Non-Obvious Profile*, (NOP), which stores behavioral data not explicitly given by the user, but automatically created by analyzing the user behavior on the web site. The behavioral data stored in the NOP indicates, for example, which pages a user has visited, and which actions she/he has performed on that web page. Most of this information is extracted out of the web server log, but Gugubarra has refined the common click-stream analysis [4], [5], by extending it with new concepts, namely: *zones*, *topics*, *actions*, and *weights* [3], [6].

In [7], we introduced the *Relevance Profile* (RP). An RP is calculated by integrating the two available profiles for the user, the NOP and the FP. The benefit of the RP is that it integrates both, calculated data as well as explicit feedback of the user, in a flexible way into one single user profile. Figure 1 shows an example of an RP, where we calculated the data of a user u_m at time t_n, based on her/his behavior and explicit feedback, showing a supposed low interest in topic T_1 (0.3), high interest in topic T_2 (1.0), and no interest in topic T_3 (0.0).

$$RP_{u_m,t_n} = \begin{pmatrix} 0.3 \\ 1.0 \\ 0.0 \end{pmatrix} \begin{matrix} \leftarrow T_1 \\ \leftarrow T_2 \\ \leftarrow T_3 \end{matrix}$$

Figure 1. Relevance Profile of user u_m for three topic T_1, T_2, T_3.

In what follows, we assume that users are aware and have granted permission that implicit data is collected and kept in their profile for them.

To measure the similarity of the users we are using different techniques from graph theory. First, we will introduce the similarity threshold that helps the web site owner in building the graphs of her/his community. Second, we will provide several algorithms to find important users in the similarity graph. There exists not only one valid definition for importance of users because it depends–as always–on the point of view. For this reason, we provide nine algorithms to discover the importance of users. Two of these algorithms are new designed in respect to the needs of similarity graphs.

The rest of the paper is structured as follows: Section II recalls the basic concepts that will be used in the rest of

 ISBN: 978-1-61208-242-4

the paper. In Section III, we define the similarity of users. Section IV presents the main contribution of this paper, our analysis tool for building and mining similarity graphs. In Section V we use our analysis tool with a real usage dataset. Section VI presents the conclusions of this work.

II. Literature Review

In this section we introduce basic concepts from literature that are used in our framework.

A. Similarity measurement

Due to the fact that the RP contains all information about the interests of the users, we want to use it to compute the similarity between the interests of *all* users. First we have to definite the equality of users:

Two users u_i and u_j are equal in respect to a topic T_r of a web site at time t_n if the interest values of T_r of their RPs are equal:

$$RP_{u_i,t_n}(T_r) = RP_{u_j,t_n}(T_r) \; where \; i \neq j. \tag{1}$$

To compare users we need a measurement for similarity. Similarity measurements are very common in the research field of data mining. For example, documents are often represented as feature vectors [8], which contain the most significant characteristics like the frequency of important keywords or topics. To compute the similarity of documents, the feature vectors are compared with the help of distance measurements: the smaller the distance the more similar the documents are.

Gugubarra interest profiles, i.e., the RP, can be considered as feature vectors of the users, too. They contain the most significant characteristics of our users, e.g., the interests in different topics of a web site. Therefore we can use the similarity measurements of data mining theory to compute similarity between the members of our community.

An important requirement on the similarity measurement algorithm is its performance, because a web community can cover lots of users. Consequently we have to choose a similarity measurement with a high performance so that the analysis program will scale with the high number of users. Aggarwal et al. proved in [9] that the *Manhattan Distance*, also known as *City Block Distance* or *Taxicab Geometry*, is very well suited for high dimensional data. We shared in [6] that web sites may have up to 100 topics. Thus, we have to deal with very high dimensional feature vectors, i.e., one dimension per topic.

The Manhattan Distance (L_1-norm) [10] is defined as follows:

$$d_{\text{Manhattan}}(a,b) = \sum_i |a_i - b_i| \tag{2}$$

with $a = RP_{u_m,t_n}$, $b = RP_{u_r,t_n}$ and $m \neq r$.

B. Graph Theory

Leonhard Euler founded the graph theory with the Seven Bridges of Königsberg problem [11]. In this section, we present the basic definitions of graph theory that are necessary for our tasks.

A *graph* G [12] is a tuple $(V(G), E(G))$. $V(G)$ is a set of *vertices* of the graph and $E(G)$ is the set of *edges* which connects the vertices[3].

A graph G can be represented [14] by an *adjacency matrix* $A = A(G) = (a_{ij})$. This $n \times n$ matrix, n is the sum of the vertices of G, is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } \{v,w\} \in E(G) \\ 0 & \text{otherwise.} \end{cases} \quad with \; v, \; w \in V(G) \tag{3}$$

In a *simple graph* an edge connects always *two* vertices [15]. This means that $E(G)$ consists of unordered pairs $\{v,w\}$ with $v,w \in V(G)$ and $v \neq w$ [12]. In a social network vertices could represent the members of this network and the edges could stand for the friendship relation between these vertices–so friends are connected together.

Every pair of distinct vertices of a *complete graph* [12] are connected together.

The connections between edges can be *directed* or *undirected*. In a directed graph the edges are an ordered pair of vertices v, w and can only be traversed in the direction of its connection. This means that a *simple graph* is undirected. This feature is very useful, e.g., to model the news feed subscriptions of a user in a social network, a one-way friendship.

A *loop* is a connection from a vertex to itself [14]. A loop is not an edge.

Labeled vertices make graphs more comprehensible. Vertices can be labeled with identifiers, e.g., in the social network graph with the names of the users.

In the same way edges can be labeled to denote the kind of connection. In the social network graph example, the label could represent the kind of relation between users, e.g., friend or relative.

With *weighted graphs*, the strength of the connection between the single vertices can be modeled. Every edge has an assigned weight. In a social network the weight could be used to display the degree or importance of the relationship of the users. A weighted graph can also be represented by an adjacency matrix (see Definition 3 above) where a_{ij} is the weight of the connection of $\{v,w\}$. See Example 4 for an adjacency matrix of a similarity graph of five users:

[3]Sometimes it is postulated [12] that $V(G)$ and $E(G)$ has to be finite but there exists also definitions about infinite graphs [13]. However, the number of web site users should be finite.

 ISBN: 978-1-61208-242-4

$$A = \begin{pmatrix} 0.00 & 1.28 & 1.19 & 2.79 & 1.18 \\ 1.28 & 0.00 & 1.63 & 2.83 & 1.90 \\ 1.19 & 1.63 & 0.00 & 2.50 & 1.35 \\ 2.79 & 2.83 & 2.50 & 0.00 & 2.85 \\ 1.18 & 1.90 & 1.35 & 2.85 & 0.00 \end{pmatrix} \quad (4)$$

Every number represents the weight of the edges between two vertices, e.g., $a_{2,4} = 2.83$ represents the edge weight of the two vertices with the numbers 2 and 4. The diagonal of this matrix is 0.00 because the graph has no loops. In an undirected graph the adjacency matrix is symmetric.

A vertex w is a *neighbor* of vertex v if both are connected via the same edge. The neighborhood of v consists of *all* neighbors of v. In a social network a direct friend is a neighbor and all direct friends are the neighborhood.

A *path* [16] through a graph G is a sequence of edges $\in E(G)$ from a starting vertex $v \in V(G)$ to an end vertex $w \in V(G)$. If there exists a path form vertex v to w both vertices are connected. The number of edges on this path is called *length* of the path and the *distance* between v and w is the length of the shortest path between these two vertices. A path with the same start and end point is called *cycle*. Two vertices v and w are *reachable* from each other if there exists a path with the start point v and the end point w. If all vertices are reachable from every vertex the graph is called *connected*.

G' is a *subgraph* [14] of G if $V(G') \subset V(G)$ and $E(G') \subset E(G)$. G is than the *supergraph* of G' with $G' \subset G$.

A *community* in a graph is a *cluster* of vertices. The vertices of a community are dense connected.

C. Importance

There exist many algorithms to measure the importance of a vertex in graph. We introduce seven of the most common algorithms:

Sergin Brin and Lawrence Page [17] used their *PageRank* algorithm to rank web pages with the link graph of their search engine Google[4] by importance. This algorithm is scalable on big data sets (i.e., search engine indices). Usually the PageRank algorithm is for unweighted graphs. But there exists also implementations for weighted graphs [18]. Pujol et al. [19] developed an algorithm to calculate the reputation of users in a social network. The results of the comparison of their algorithm with the PageRank show that the PageRank is also well suited for reputation calculation, i.e., importance calculation.

The *Jaccard similarity coefficient* [20] of two vertices is the number of common neighbors divided by the number of vertices that are neighbors of at least one of the two vertices being considered [21]. Here the pairwise similarity of all vertices is calculated.

The *Dice similarity coefficient* [21] of two vertices is twice the number of common neighbors divided by the sum of the degrees of the vertices. Here the pairwise similarity of all vertices is calculated.

Nearest neighbors degree calculates the nearest neighbor degree for all vertices. In [22] Barrat et al. define a nearest neighbor degree algorithm for weighted graphs.

Closeness centrality [23] measures how many steps are required to access every other vertex from a given vertex.

Hub score [24] is defined [21] as the eigenvector of AA^T where A is the adjacencies matrix and A^T the transposed adjacencies matrix of the graph.

Eigenvector centrality [25] [21] correspond to the values of the first eigenvector of the adjacency matrix. Vertices with high eigenvector centralities are those which are connected to many other vertices which are, in turn, connected to many others.

III. User Similarity

In Gugubarra, the RP provides the most significant information about a user which is calculated from all implicit and explicit feedback profiles. To calculate user similarity we take the RP interest value of every topic of each user and calculate the Manhattan Distance between all users of the web community as illustrated in the following example:

Lets assume we have a web site with three topics T_1, T_2, and T_3. This web site has two registered users u_1 and u_2. The RPs of the two users were calculated at time t_1:

$$RP_{u_1,t_1} = \begin{pmatrix} 1.0 \\ 0.5 \\ 0.0 \end{pmatrix}, \; RP_{u_2,t_1} = \begin{pmatrix} 0.6 \\ 0.8 \\ 0.2 \end{pmatrix} \quad (5)$$

The Manhattan Distance is calculated as follows:

$$d_{\text{Manhattan}}(RP_{u_1,t_1}, RP_{u_2,t_1}) = \quad (6)$$

$$= |1.0 - 0.6| + |0.5 - 0.8| + |0.0 - 0.2| = 0.9$$

where 0.9 is the distance of the interests of the both users, i.e., the similarity.

Therefore, our focus is on a large group of users (i.e., the whole web community) and not only on a single user or on a single topic. The following sections should clarify research questions such as:

- Which users are important for the web community?
- Which users have similar interests?
- How similar are the interests of the users of the community?
- How is this specific community structured?

[4] https://www.google.com/

 ISBN: 978-1-61208-242-4

By answering these questions we want to give the web site owner a useful tool to enhance her/his marketing strategies and rise as consequence the click rates of her/his portal.

IV. Analysis of Similarity Graphs

We developed a new tool for building and analyzing similarity graphs. We integrated several algorithms from different research areas for the analysis of the graphs. This tool is written in R[5]. R is an open source project with a huge developer community. The archetype of R is the statistic programing language S[6] and the functional programming language Scheme[7]. R has a big variety of libraries with many different functions for statistical analytics. For graph analysis R provides two common libraries: the Rgraphviz[8] and the igraph[9] library. We are using the later for our implementation because it provides more graph analytics algorithm[10] [26] and it is better applicable for large graphs. The igraph library is also available for other programming languages (e.g., C, Python).

Our graph analytics tool follows a two phases work flow. In the first phase the similarity graph is built and in the second the built graph can be analyzed with different algorithms. The next paragraphs describe the work flow in more detail.

A. Building Similarity Graphs

In the first work flow phase, the similarity graph of RPs of the users of the web community has to be build. We use an undirected, vertices and edges labeled, weighted graph without loop to build a model for the similarity of the web community users. The weighted edges represent the similarity between the vertices which stand for the users. The edges are labeled with the similarity value, that is the Manhattan Distance between the RPs of the users. The labels of the vertices are the user IDs. We use an undirected graph because the similarity of two users can be interpreted in both directions. Figure 2 and 3 show examples of a similarity graph. As mentioned before, in the research field of social networks graph analysis is used to detect social structures between the users, like in [27]. These graphs represent the friend relationship of the users and is in comparison to our work different. We use *weighted* graphs to embody the similarity of users where the edge weights represent the similarity between the interests of the users. So we are not able to use the graph analytics algorithm tools from the social network analysis.

In our tool, the web site owner can chose different alternatives to build a similarity graph for the analysis. The

[5]http://www.r-project.org/

[6]http://stat.bell-labs.com/S/

[7]http://www.r6rs.org/

[8]http://www.bioconductor.org/packages/release/bioc/html/Rgraphviz.html

[9]http://igraph.sourceforge.net/

[10]http://igraph.sourceforge.net/doc/html/index.html

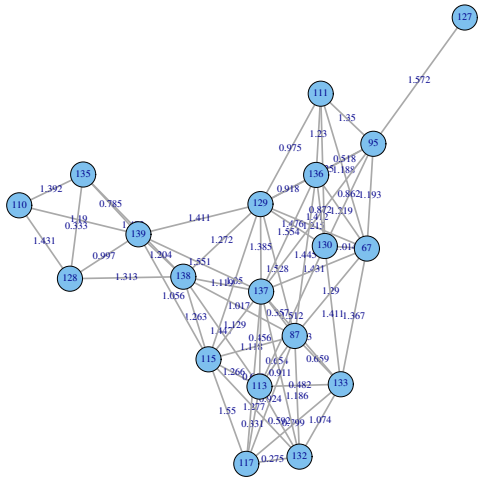

Figure 2. Smallest connection graph.

vertices of the graph (the users) are connected via edges that represent the similarity. It is possible to connect every user to all other users so that a complete graph represents the similarity between all users. This graph is huge and not easy to understand. To reduce the complexity of this graph we introduce a *similarity threshold*. This threshold defines how similar the users must be to be connected together. Only users are connected via vertices whose Manhattan Distance of their RPs is smaller (remember: the smaller the distance the more similar users are) than the chosen threshold. Our analysis tool provides several predefined options to build different graphs with different thresholds. All these graphs are subgraphs of the complete similarity graph of the whole web community:

- **Smallest connected graph**: with this option the threshold increases until every user has at least one connection to another user. In Figure 2, user no. 127 was added last to the graph and has a Manhattan Distance of 1.572. Accordingly all connected vertices have a similarity smaller or equal to 1.572. The result is **one** connected graph.
- **Closest neighbor graphs**: here users are only connected with their most similar neighbors. Every vertex has at least one edge to another vertex. If there exist more most similar neighbors with the same edge weight, the vertex is connected to all of them. This can result in **many** independent graphs as displayed in Figure 3. The difference to the nearest neighbor algorithm is that the nearest neighbor algorithm calculates a path through an existing graph by choosing always the nearest neighbor of the actual vertex.
- **Minimum spanning tree** [28]: is a subgraph where all users are connected together with the most similar users. In contrast to the "closest neighbor graph" we have **one** connected graph.

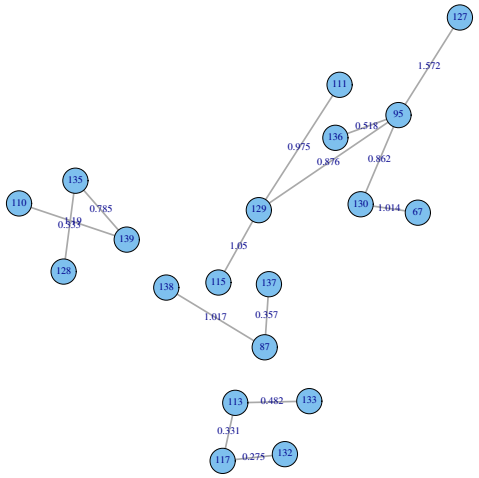

Figure 3. Closest neighbor graphs.

- **Threshold graph**: at last the web site owner can chose a similarity threshold on her/his own. To simplify the choice, the tool suggests two thresholds to the owner: a minimum threshold and a maximum threshold. With the minimum threshold only the most similar users are connected together and with the maximum threshold all users are connected together with every user. So the owner can chose a value between the suggested thresholds to get meaningful results.

B. Similarity Graph Mining Algorithms

In the second work flow phase the web site owner can analyze the graph, generated in the first phase of the work flow, with different algorithms. The aim here is to detect the important users in the graph.

What is an important user? There exists not only one valid definition because it depends–as always–on the point of view. In social networks, e.g., the importance of users often stands for their reputation. The reputation of a user can be measured, e.g., by its number of connectors to other users. Therefore a connector in social networks has another meaning, i.e., the friendship, like in our similarity graphs, we can not use this definition of user importance.

In a social graph a user could be important if she/he is central in respect to the graph. Centrality means that from this very user all other users should be not far away–it should be the nearest neighbor. These highly connected users are often referred as *Hubs* or *Authorities* [24]. Hubs have many outgoing edges while Authorities have many incoming edges.

In a weighted similarity graph high importance could mean that this user is the most similar to other users–she/he should have many edges to other vertices and the edges weights should be as low as possible.

Accordingly, we provide nine algorithms to discover the importance of users. Therefore the importance is defined by the used algorithm which are explained below.

- **PageRank**: The vertex with the highest "PageRank" is the most important user.
- **Jaccard similarity coefficient**: We interpret the most similar vertex as the most important user.
- **Dice similarity coefficient**: Like above we interpret the most similar vertex as the most important user.
- **Nearest neighbors degree**: If a vertex has many neighbors it can be considered as important.
- **Closeness centrality**: Vertices with a low closeness centrality value are important.
- **Hub score**: Vertices with a high score are named hubs and should be important.
- **Eigenvector centrality**: Vertices with a high eigenvector centrality score are considered as important users.

As these seven algorithms above are not *extra* designed to find the important vertices, i.e., users, in similarity graphs of user interests, we developed two new algorithms:

- **Weighted degree**: This simple algorithm choses the vertex with the most connections. Vertices with many connections are important users because they are similar to other user. Actually they are connected with other users cause of their similarity. If there are vertices with the same number of connections it takes the vertex with the lowest edge weights. Therefore the most unimportant vertex has fewer connections to other vertices and the highest edge weights.
- **Range centrality**: The idea behind this algorithm is that a user is important who has many connections in comparison with the other users of the graph, short distance to her/his neighbors, and low edge weights. The range centrality is defined as follows:

$$C_r = \frac{range^2}{aspl + aspw} \tag{7}$$

The *range* is the fraction of the number of users that are reachable from the analyzed vertex and of all users of the graph. We take the square of the range because we consider a user as very important that is connected with many other users:

$$range = \frac{\#reachable\ user}{\#all\ user} \tag{8}$$

The average shortest path length (*aspl*) is the average length of all shortest paths divide by the number of all shortest paths. The shortest paths are calculated with the analyzed vertex as starting point:

$$aspl = \frac{average\ shortest\ paths\ length}{\#shortest\ paths} \tag{9}$$

Table I
EVALUATION RESULTS: IDS OF THE USERS WITH MAXIMUM AND MINIMUM IMPORTANCE OF EVERY GRAPH TYPE (ROWS) FOR DIFFERENT ALGORITHMS (COLUMNS).

		Page Rank	Nearest N.D.	Dice S.C.	Jaccard S.C.	Closeness C.	Hub Score	Eigen-vector C.	Weighted D.	Range C.
SCG	Max	220	222	241	232	93,220	93	106	93	220
	Min	104	138	104	104	64	104	104	104	104
CNG	Max	220	169	79,80,121,200	79,80,121,200	87, 213	66	204	66	87,213
	Min	68	67,127,...	67,127,...	67,127,...	67,127,...	100,246	244	104	67,127
MST	Max	213	169	200	68,80,121	156	261	129	66	156
	Min	104	170	112,126,166	112,126,166	189	189	88	104	189
CG	Max	213	213	all	all	all	all	104	213	213
	Min	104	104	all	all	all	all	241	79	104

With the average shortest path weight (*aspw*) we take into account that the weight of the connected vertices should be very low, i.e., the vertices should be very similar. It's the fraction of the sum of all shortest paths weights and of the number of all shortest paths:

$$aspw = \frac{sum\ of\ all\ shortest\ paths\ weights}{\#shortest\ paths} \quad (10)$$

In the next section we will use our analysis tool with real usage data and compare our new algorithms with the established ones.

V. EVALUATION

A. Material and Methods

To evaluate our algorithms, we use the real usage data from our institute web site[11], i.e., the users' session log files of the site community. We observed 191 registered users over two years. For each user an RP is calculated. Next, we use our analytics tool to build similarity graphs from the RPs of the users and calculate for every graph type the most important and the most unimportant user.

B. Results

Table I displays the results of our calculations. The rows present the different graph types: *SCG* stands for *S*mallest *C*onnection *G*raph, *CNG* for *C*losest *N*eighbor *G*raph, *MST* for *M*inimum *S*panning *T*ree, and *CG* for *C*omplete *G*raph. For every graph type, the user with maximum and minimum importance is displayed. Every column presents one importance algorithm. We can observe the following fact in the dataset in respect to our algorithms, the weighted degree and the rang centrality:

[11] http://www.dbis.cs.uni-frankfurt.de/

In the SCG, the range centrality calculates the same un-/important users like the PageRank and eigenvector closeness, the weighted degree algorithm like the hub score. The majority of algorithms calculate the same unimportant user, only the nearest neighbor degree and closeness centrality differs.

In the CNG, the range centrality and the closeness centrality calculates the same two important users. But the unimportant users are different. The results of the weighted degree for the important user are like the hub score, but the unimportant user is different.

In the MST, the results of the range centrality equals the closeness centrality, while the weighted degree calculates the same unimportant user as the PageRank.

In the CG, user no. 213 is the most important user for both, the weighted degree and the range centrality. The PageRank and the nearest neighbor degree have the same result, only the eigenvector centrality differs. The user no. 104 is the most unimportant user for the rang centrality, the PageRank, and the nearest neighbor degree. The Dice similarity coefficient, Jaccard similarity coefficient, closeness centrality, and hub score are not able to find an un-/important user in the complete graph, because these algorithm do not include the edge weights into their calculation.

C. Discussion

Since there is no objective measurement for importance, we compare established algorithm with our approach. Every algorithm calculates importance in a different way, because every algorithm author has another definition of importance. Most of the algorithms are not designed for similarity or even weighted graphs. Therefore a comparison is not easy.

The weighted degree algorithm firstly focuses on the number of connected neighbors and secondly on the weights of the connected edges. The results of the weighted degree algorithm are very different from the results of the other

algorithm, only the hub score seems to be comparable. In contrast to the hub score the weighted degree algorithm is able to find an important user in a complete graph because it considers the edge weights of the connections (if there are users with the same number of connections which is alway the case in a complete graph).

Similarly, the range centrality focuses on the number of connections, but also on the reachability of the user and the path length. In other words, it considers the whole graph. In comparison to the other algorithms the range centrality is very similar to the closeness centrality but the results differs at the complete graph. Here, our range centrality algorithm calculates important and unimportant users, which is similar to the PageRank algorithm, but the closeness centrality can not calculate any similarity. This is an advantage of our algorithm.

In summary, we think that our new algorithms are a good alternative for finding important users in similarity graphs.

VI. Conclusion

With the results of graph analysis we are now able to answer the research questions of Section III:

- Which are the important users of the web community?
  We provide several algorithms (see Section II-C) to calculate the important user(s) of the community. The definition of importance is dependent from the used algorithm. For example, vertices with many low weight connections can be considered as the important users of the community. These users are very similar to the other users, expressed by the low edge weight.
- Which users have similar interests?
  All users are connected via weighted edges. Users with similar interests have connections with low weights. The web site owner can also define which users are connected together by selecting a similarity threshold (see work flow phase one, Section IV-A). As result only similar users are connected via edges.
- How similar are the interests of the users of the community?
  The lower the weight of the edges the more similar are the users of the community. We give the web site owner the possibility to set thresholds to identify quickly the similarity of her/his community (see Section IV-A).
- How is the community structured? Is it a homogeneous community where every user has similar interests or is it heterogeneous?
  The visualized graph of the community will give the web site owner an overview over the structure of the whole community of her/his web portal.

With answers to these questions, a web site owner is now able to start more focused marketing campaigns. To test new contents or features for her/his web site she/he could start with the most similar users, these users can be considered as an archetype for her/his community.

Acknowledgment

We would like to thank Roberto V. Zicari, Natascha Hoebel, Karsten Tolle, Naveed Mushtaq, and Nikolaos Korfiatis of the Gugubarra team, for their valuable support and fruitful discussions.

References

[1] I. Esslimani, A. Brun, and A. Boyer, "Detecting leaders in behavioral networks," in *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, ser. ASONAM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 281–285. [Online]. Available: http://dx.doi.org/10.1109/ASONAM.2010.72

[2] N. Mushtaq, P. Werner, K. Tolle, and R. Zicari, "Building and evaluating non-obvious user profiles for visitors of web sites," in *IEEE Conference on E-Commerce Technology (CEC 2004)*. Los Alamitos, CA, USA: IEEE Computer Society, 2004, pp. 9–15. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/ICECT.2004.1319712

[3] N. Hoebel and R. V. Zicari, "Creating user profiles of web visitors using zones, weights and actions," in *Tenth IEEE Conference On E-Commerce Technology (CEC 2008) And The Fifth Enterprise Computing, E-Commerce And E-Services (EEE 2008)*. Los Alamitos, USA: IEEE Computer Society Press, 2008, pp. 190–197.

[4] B. Weischedel and E. K. R. E. Huizingh, "Website optimization with web metrics: a case study," in *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet*, ser. ICEC '06. New York, NY, USA: ACM, 2006, pp. 463–470. [Online]. Available: http://doi.acm.org/10.1145/1151454.1151525

[5] S. Jung, J. L. Herlocker, and J. Webster, "Click data as implicit relevance feedback in web search," *Information Processing and Management: an International Journal*, vol. 43, no. 3, pp. 791–807, 2007.

[6] N. Hoebel, N. Mushtaq, C. Schefels, K. Tolle, and R. V. Zicari, "Introducing zones to a web site: A test based evaluation on semantics, content, and business goals," in *CEC '09: Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 265–272.

[7] C. Schefels and R. V. Zicari, "A framework analysis for managing feedback of visitors of a web site," *International Journal of Web Information Systems (IJWIS)*, vol. 8, no. 1, pp. 127–150, 2012.

[8] L. Yi and B. Liu, "Web page cleaning for web mining through feature weighting," in *Proceedings of the 18th international joint conference on Artificial intelligence*, ser. IJCAI'03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 43–48. [Online]. Available: http://dl.acm.org/citation.cfm?id=1630659.1630666

 ISBN: 978-1-61208-242-4

[9] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional spaces," in *Proceedings of the 8th International Conference on Database Theory*, ser. ICDT '01. London, UK: Springer-Verlag, 2001, pp. 420–434. [Online]. Available: http://dl.acm.org/citation.cfm?id=645504.656414

[10] S.-H. Cha, "Comprehensive survey on distance / similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.154.8446\&rep=rep1\&type=pdf

[11] L. Euler, "Solutio problematis ad geometriam situs pertinentis," *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, vol. 8, pp. 128–140, 1736.

[12] R. J. Wilson, *Introduction to Graph Theory*. Longman, 1979. [Online]. Available: http://books.google.com.ag/books?id=5foZAQAAIAAJ

[13] D. Jungnickel, *Graphen, Netzwerke und Algorithmen (3. Aufl.)*. BI-Wissenschaftsverlag, 1994.

[14] B. Bollobás, *Modern Graph Theory*, ser. Graduate texts in mathematics. Springer, 1998. [Online]. Available: http://books.google.com/books?id=SbZKSZ-1qrwC

[15] M. A. Rodriguez and P. Neubauer, "Constructions from dots and lines," *CoRR*, vol. abs/1006.2361, 2010.

[16] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[17] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the Seventh International World Wide Web Conference*, vol. 30, no. 1-7, 1998, pp. 107–117. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016975529800110X

[18] D. Nemirovsky and K. Avrachenkov, "Weighted pagerank: Cluster-related weights," in *TREC*, 2008.

[19] J. M. Pujol, R. Sangüesa, and J. Delgado, "Extracting reputation in multi agent systems by means of social network topology," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, ser. AAMAS '02. New York, NY, USA: ACM, 2002, pp. 467–474. [Online]. Available: http://doi.acm.org/10.1145/544741.544853

[20] P. Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, February 1912. [Online]. Available: http://www.jstor.org/stable/2427226

[21] G. Csardi, *Network Analysis and Visualization*, 0th ed., http://igraph.sourceforge.net, August 2010, package 'igraph'.

[22] A. Barrat, M. Barthèlemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 11, pp. 3747–3752, 2004. [Online]. Available: http://www.pnas.org/content/101/11/3747.abstract

[23] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[24] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, Sep. 1999. [Online]. Available: http://doi.acm.org/10.1145/324133.324140

[25] P. Bonacich, "Power and centrality: A family of measures," *American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, March 1987.

[26] J. Marcus, "Rgraphviz," Presentation, 2011, accessed: November 4th 2011. [Online]. Available: http://files.meetup.com/1781511/RgraphViz.ppt

[27] L. A. Adamic and E. Adar, "Friends and Neighbors on the Web," *Elsevier Social Networks*, vol. 25, no. 3, pp. 211–230, July 2003.

[28] R. C. Prim, "Shortest connection networks and some generalizations." *Bell Systems Technical Journal*, pp. 1389–1401, November 1957. [Online]. Available: http://www.alcatel-lucent.com/bstj/vol36-1957/articles/bstj36-6-1389.pdf

 ISBN: 978-1-61208-242-4