Computing User Importance in Web Communities by Mining Similarity Graphs

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

***Abstract*—The economic success of the World Wide Web makes it a highly competitive environment for web businesses. For this reason, it is crucial for web business owners to learn what their customers want. In this paper, we provide a useful tool to the web site owner for analyzing her/his web community. In particular, the web site owner can compute the importance of the users and analyze the structure of the specific community by comparing the interests of the users. Therefore, we present the conception and implementation of a tool 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. We introduce different graph types for analyzing profiles of web community users. Moreover, we propose two new algorithms for finding important users of an on-line community.**

Keywords-Computer aided analysis; World Wide Web; Data analysis; Graph theory; Application software.

I. Introduction

This paper is an extended version of [1] presented at the First International Conference on Data Analytics in Barcelona in 2012.

These days, web-based user communities enjoy great popularity. The social network Facebook[1] has more than 1 billion active users [2] and even the relatively new Google+[2] about 235 million [3]. In this highly competitive environment, it is crucial for web site owners to understand and satisfy their web community.

To reach this goal, we present the conception and implementation of 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 [4] [5], developed by the DBIS research group at the Goethe-University Frankfurt am Main, to compute interest profiles of web users. 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 web community structured?

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

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 similarity graph of her/his community. This threshold sets how similar the users must be to be connected together in the similarity graph. In addition to that, it reduces the complexity of the graph. 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.

In contrast to other researches that derive the importance of users from the social structures of web communities (e.g., Trusov et al. [6]), we calculate the user importance from their interests.

The rest of the paper is structured as follows: Section II outlines related work and Section III introduces basic concepts and definitions that will be used in the rest of the paper. Section IV presents the main contribution of this paper, our analysis tool for building and mining similarity graphs and an implementation of a prototype. After we evaluate our analysis tool with a real usage dataset in Section V, we integrate it into the Gugubarra Framework in Section VI. Section VII presents the conclusion and future work.

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

II. Related Work

Previous research discovered community structures in social networks, but focused on the pure friendship or relationship structure of these communities. E.g., Rongjing Xiang et al. developed in [7] an unsupervised model to estimate relationship strength from interaction activity (e.g., communication, tagging) and user similarity. In this work, we calculate the relationship structure from the interests of the web community users. Moreover, we use their interest profiles to determine the relationship strength between the users.

To evaluate the web user's level of expertise (i.e., importance) on a given topic, Jidong Wang et al. [8] propose a link analysis. They use a unified directed graph, where the nodes of the graph are users and web pages and the directed edges represent the hyper links between web pages and user's visit of the web pages. The link analysis algorithm is derived from the algorithm presented by Kleinberg in [9] that we also use to compute the importance of the users. Moreover, we use nine different algorithms to determine the importance of web users because we think there is not only one valid definition for importance.

The detection of important users, i.e., leaders in behavior networks, is the focus of the publication of Esslimani et al. [10]. The behavior network is a graph where the nodes represent the users and the edges represent the links between users. The navigational similarities are the weights of the edges. The detection of leaders relies on their high connectivity in these behavioral networks and their potentiality of propagating accurate appreciations. We also understand high connectivity of users in a network as an indicator for their importance. Both of our new algorithms to detect important users take the connectivity of the users into account.

In [11], Paliouras uses a similarity threshold to transform a weighted user graph into an unweighted graph. As a side effect, the connectivity of the graph is reduced. In our work, we use a similarity threshold to reduce the complexity of the web community graph too. In contrast to Paliouras, we keep the resulting graph weighted and use the edge weights as additional information to calculate the important users of the web community.

III. Basic Concepts and Definitions

In this section, we introduce the analytic framework Gugubarra, which is used to calculate the interest profiles of the web community users. Furthermore, we present the definitions of the user equality and the user similarity, concepts of the graph theory, and seven algorithms to determine the importance of users.

A. The Web Analytics Software Gugubarra

The web based analytic framework *Gugubarra*, also described in [4] [5] [12], is a prototype system developed by the Databases and Information Systems (DBIS) research group at the Goethe-University Frankfurt am Main. The purpose of the system is to help the owner or manager of a web site to more fully understand the interests of the registered users on her/his web site. We use the Gugubarra interest profiles of the users to build the similarity graphs. Therefore, we introduce the basic concepts of this framework.

In this project, a *web site* is a collection of web pages, where *visitors* or *users* can register and log on. The combined group of *registered* users of this web site are called the *web community*. This web site is maintained by a web site *owner* who controls the content and decides on the business strategies or goals. During a user *session*, which is defined as the time between the log-in and the log-out of a web user, all web page requests are stored in the log files of the web server and enriched with additional information, such as zones, topics, and actions, which are explained in [13]. All of these data are used to calculate profiles describing the interests of every web site user. 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, as follows.

The *explicit* user data are stored in *two* different profiles, in the *Obvious Profile* (OP) and in the *Feedback Profile* (FP) [4]. Explicit user data means, that the web site user is directly asked by the web site owner about the data, e.g., by an e-mail or a web form. The OP [13] contains identification and personal data, e.g., name, address of the user. The FP holds the explicit feedback of the user. The advantages of these types of data is that they come directly from the user and that the user is aware of being asked about her/his interests. Thus, the results can reflect the interests of a user very accurately. However, the disadvantages are that a user can misinterpret the topics and/or give inaccurate answers. The explicit user feedback is a valuable source for the calculation of user interest profiles.

In addition to the explicit user data, the Gugubarra Framework calculates user interests from the *implicit* user data, too. The sources of the implicit user data are the interactions of the visitors with the web site, particularly, the behavioral data. With these data, Gugubarra compensates for the constraints of the explicit user data mentioned above. The implicit user data are stored in the *Non-Obvious Profile* (NOP), which consists of the *Action Profile* (ActP) and the *Duration Profile* (DurP) [13]. In [14], the implicit user profiles of the Gugubarra Framework are extended with data form the mouse activities of the web site user.

The *Relevance Profile* (RP), introduced in [15] and [16], unites the explicit and the implicit feedback profiles of a user into a single interest 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 implicit and explicit feedback, showing a supposed high interest in topic T_2 (1.0), lower interest in topic T_1 (0.3), 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 topic T_1, T_2, T_3.

We use the RP to calculate the graphs of the web

community. Therefore, we provide the necessary definitions in the next sections.

B. 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 [17], 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 [18] that the *Manhattan Distance*, also known as *City Block Distance* or *Taxicab Geometry*, is very well suited for high dimensional data. We shared in [19] that web sites may have up to 100 topics. Thus, we have to deal with high dimensional feature vectors, i.e., one dimension per topic.

The Manhattan Distance (L_1-norm) [20] 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 $m \neq r$.

To calculate the *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:

Let us 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:

$$RP_{u_1,t_1} = \begin{pmatrix} 1.0 \\ 0.5 \\ 0.0 \end{pmatrix}, \; RP_{u_2,t_1} = \begin{pmatrix} 0.6 \\ 0.8 \\ 0.2 \end{pmatrix} \tag{3}$$

The Manhattan Distance is calculated as follows:

$$d_{\text{Manhattan}}(RP_{u_1,t_1}, RP_{u_2,t_1}) = \tag{4}$$

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

where 0.9 is the distance of the interests of the both users, i.e., the similarity. In general, the *smaller* the calculated distance is the *more similar* are the compared users to each other.

C. Graph Theory

In this section, we present the basic definitions of graph theory, which was founded by Leonhard Euler [21], that are necessary for our tasks.

A *graph* G [22] 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. Sometimes it is postulated [22] that $V(G)$ and $E(G)$ has to be finite but there exists also definitions about infinite graphs [23]. However, the number of web site users should be finite.

A graph G can be represented [24] 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:

$$a_{ij} = \begin{cases} 1 & \text{if } \{v, w\} \in E(G) \\ 0 & \text{otherwise.} \end{cases} \quad with \; v,\ w \in V(G) \tag{5}$$

In a *simple graph* an edge connects always *two* vertices [25]. This means that $E(G)$ consists of unordered pairs $\{v, w\}$ with $v, w \in V(G)$ and $v \neq w$ [22]. 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* [22] 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 [24]. 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 5 above) where a_{ij} is the weight of the connection of $\{v, w\}$. See Example 6 for an adjacency matrix of a similarity graph of five users:

$$A = \begin{pmatrix} 0.00 & 1.28 & 1.19 & 2.79 & 1.18 \\ 1.28 & 0.00 & 1.63 & 2.83 & 1.90 \\ 1.19 & 1.63 & 0.00 & 2.50 & 1.35 \\ 2.79 & 2.83 & 2.50 & 0.00 & 2.85 \\ 1.18 & 1.90 & 1.35 & 2.85 & 0.00 \end{pmatrix} \quad (6)$$

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* [26] 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* [24] 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.

D. 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 [27] used their *PageRank* algorithm to rank web pages with the link graph of their search engine Google[3] 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 [28]. Pujol et al. [29] 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* [30] 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 [31]. Here the pairwise similarity of all vertices is calculated.

The *Dice similarity coefficient* [31] 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 [32] Barrat et al. define a nearest neighbor degree algorithm for weighted graphs.

Closeness centrality [33] measures how many steps are required to access every other vertex from a given vertex.

Hub score [9] is defined [31] as the eigenvector of $A\ A^T$ where A is the adjacencies matrix and A^T the transposed adjacencies matrix of the graph.

Eigenvector centrality [34] [31] 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.

In Section V, we present an evaluation of these algorithms and compare the results with two new algorithms.

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. 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 web community?
- How is this specific web community structured?

By answering these questions, we want to give the web site owner a useful tool to enhance her/his marketing strategies, in respect of the work of Domingos and Richardson [35], and rise as consequence the click rates of her/his portal.

Before we integrate this tool in the Gugubarra Framework, we tested our concepts with a prototype written in *R*[4]. *R* is an open source project with a huge developer community. The archetype of *R* is the statistic programing language

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

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

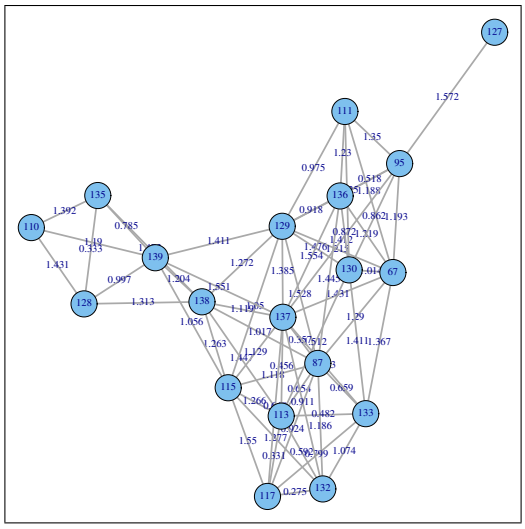


Figure 2: Smallest connection graph.

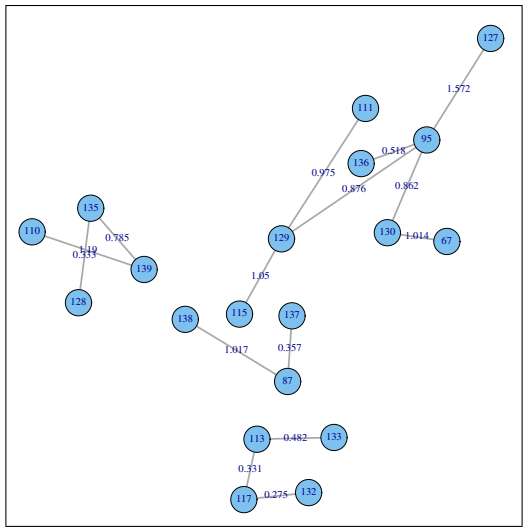


Figure 3: Closest neighbor graphs.

S[5] and the functional programming language Scheme[6]. *R* has a big variety of libraries with many different functions for statistical analytics. For graph analysis *R* provides two common libraries: the Rgraphviz[7] and the igraph[8] library. We are using the latter for our implementation because it provides more graph analytics algorithms[9] [36] 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 web community users 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. Figures 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 [37]. These graphs represent the friend relationship of the users and are different in comparison to our work. 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 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

[5] http://stat.bell-labs.com/S/
[6] http://www.r6rs.org/
[7] http://bioconductor.org/packages/release/bioc/html/Rgraphviz.html
[8] http://igraph.sourceforge.net/
[9] http://igraph.sourceforge.net/doc/html/index.html

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** [38]: 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.
- **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 of the similarity 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, as in our similarity graphs. Thus, 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* [9]. 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 or least important 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:

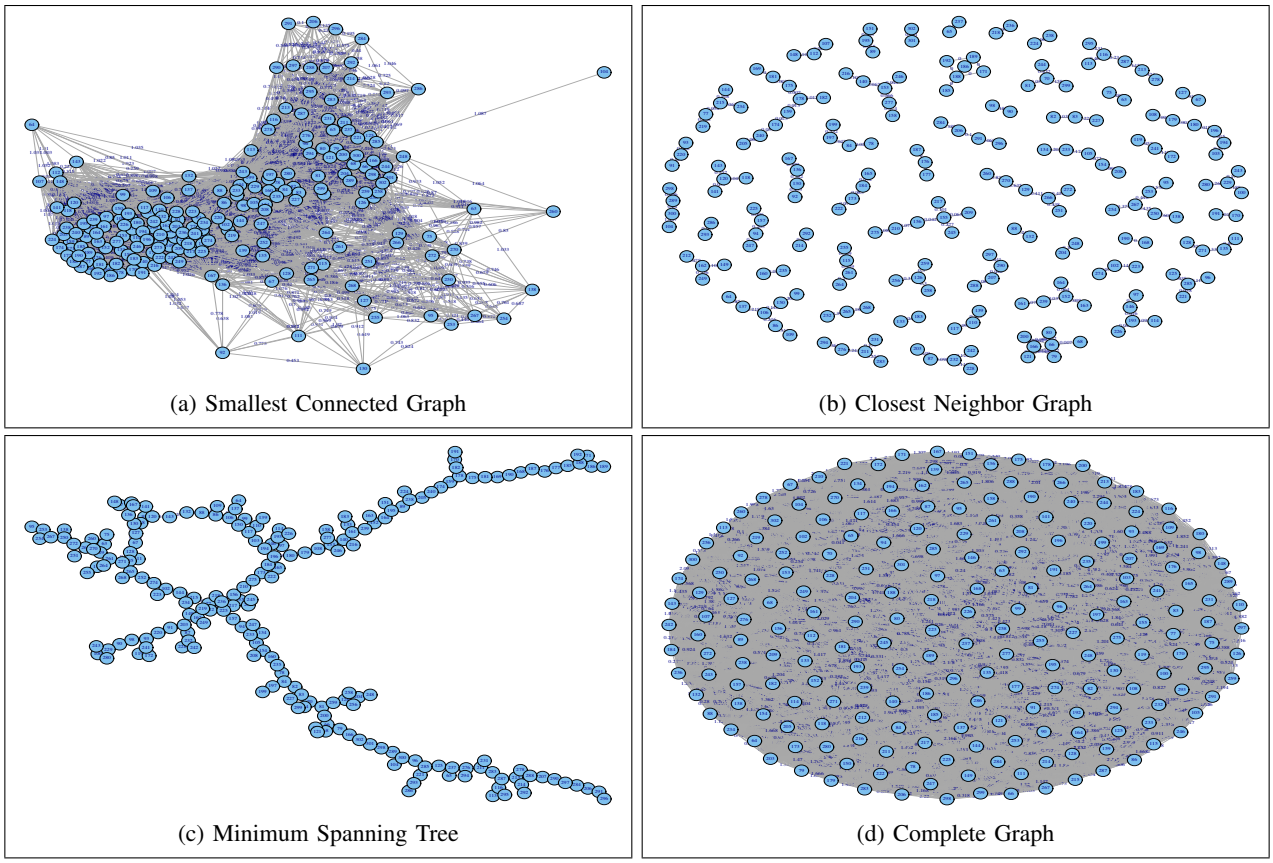
$$range = \frac{\#reachable\ user}{\#all\ user} \tag{8}$$

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:

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

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:

$$aspw = \frac{sum\ of\ all\ shortest\ paths\ weights}{\#shortest\ paths} \tag{10}$$



(a) Smallest Connected Graph

(b) Closest Neighbor Graph

(c) Minimum Spanning Tree

(d) Complete Graph

Figure 4: Similarity graphs of all users.

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

To evaluate our algorithms, we use the real usage data from our institute web site[10], i.e., the users' session log files of the site community.

A. Material and Methods

For this evaluation the data of all registered visitors of the DBIS web site were analyzed. We observed 213 registered users over two years. For each user an RP is calculated. The data were collected during the period between June 2010 and July 2012. We used the same settings for the Gugubarra Framework as described in [12]. For each topic, zone topic weights ware associated with different *zones* [13].

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.

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

B. Results

First phase: In the first phase of the analysis process, we generate the similarity graphs of the users. The four graphs are displayed in Figure 4.

Second phase: In the second phase, we analyze the graph, generated in the first phase, with different algorithms. The aim here is to detect the important users in the similarity graph.

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:

In the SCG, the range centrality calculates user no. 220 as most *important* user. The weighted degree, the closeness centrality, and the PageRank select user no. 93 as most *important*. User no. 223 is *important* for the eigenvector centrality and the Dice similarity coefficient. The hub score chose user no. 91 and the nearest neighbor degree user

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 Degree	Range C.
SCG	Max	93	216	223	232	93	91	223	93	220
	Min	104	138	104	104	104	104	104	104	104
CNG	Max	155	169	79,80,121,200	79,80,121,200	178	66	300	66	178
	Min	68	63,65,67,...	63,65,67,...	63,65,67,...	63,65,67,...	63,65,67,...	270	104	63,75
MST	Max	241	169	68,80,121,200	68,80,121	225	261	129	66	225
	Min	104	293	112,229,232	112,229,232	296	296	300	104	296
CG	Max	241	241	all users	all users	all users	all users	104	241	241
	Min	104	104	all users	all users	all users	all users	241	79	104

no. 216 as most *important*. The majority of algorithms calculate the same *unimportant* user (user no. 104), only the nearest neighbor degree centrality differs (user no. 138).

In the CNG, the range centrality and the closeness centrality calculates the same *important* user (user no. 178). The same *unimportant* users (user no. 63 and user no. 75) selects the range centrality, the hub score, the closeness centrality, the Jaccard and the Dice similarity coefficient, and the nearest neighbor degree. The results of the weighted degree for the most *important* user is no. 66 and for the most *unimportant* user no. 104.

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. The range centrality, the hub score, and the closeness centrality select user no. 296 as most *unimportant* one.

In the CG, user no. 241 is the most *important* user for all, except for the eigenvector centrality. User no. 104 is the most *unimportant* user for the rang centrality, the PageRank, and the nearest neighbor degree. The eigenvector centrality calculates exact the opposite results. The Dice similarity coefficient, the Jaccard similarity coefficient, the closeness centrality, and the hub score are not able to find an *un-/important* user in the complete graph, because these algorithms do not include the edge weights into their calculation.

C. Discussion

Since there is no method to measure the importance objectively, we compare established algorithms 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 difficult.

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 algorithms, only the hub score and the PageRank seem 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 computing the importance of users in similarity graphs.

VI. Integration into the Web Analytics Software Gugubarra

After the successful prototype testing, we integrate the tool for building and mining similarity graphs in the Gugubarra Framework. The framework consists of two parts [39] [12], the Gugubarra Designer and the Gugubarra Analyzer.

The *Gugubarra Designer* helps the web site owner to include the concepts of Gugubarra into the web site and stores the feedback data of the web site users. It is realized as a plugin for the content management system Joomla!.

The other part, the *Gugubarra Analyzer*, analyzes the data of the Gugubarra Designer, to build the user profiles, and

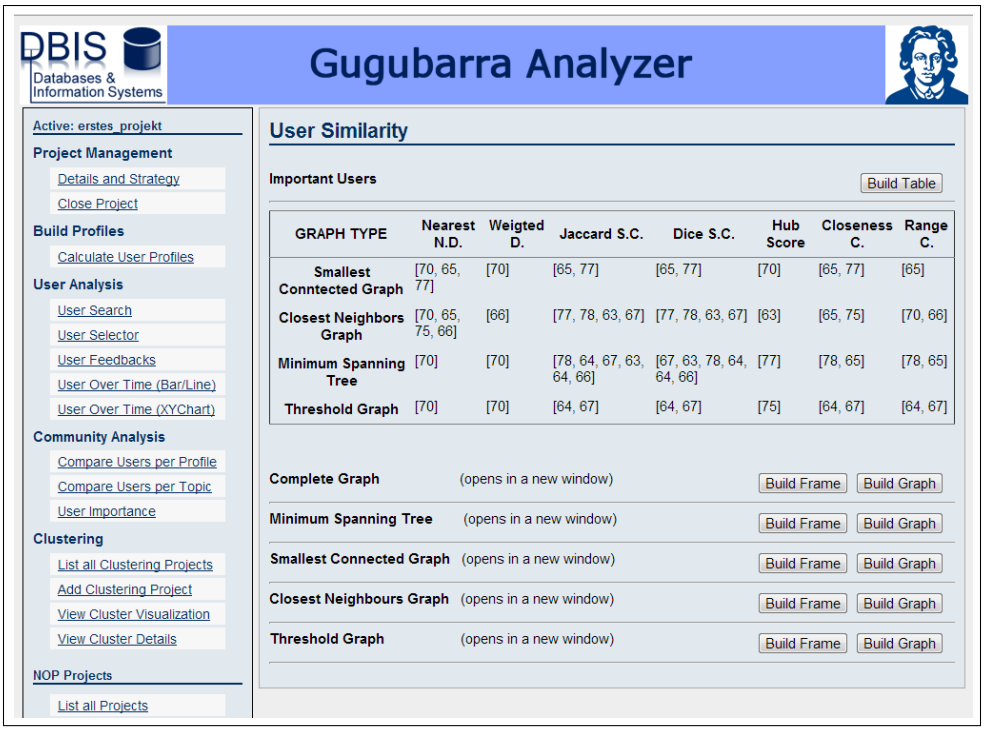


Figure 5: The GUI of the Gugubarra service for building and analyzing similarity graphs [40].

to provide the web site owner with a web application to analyze her/his web community. The Gugubarra Analyzer is a separate service and can be installed on the same or a different machine than the Gugubarra Designer.

Tapestry[11], an open source framework for creating dynamic web applications in Java[12], is used in this application to build the simple HTML[13] pages, such as the configuration dialogs. For more complex pages, the Java libraries Swing and JavaFX are used. Within the GUI configuration dialogs, the web site owner can influence the user profile calculations by changing different parameters. After configuration, the different user profiles are calculated and presented to the web site owner.

The Gugubarra Analyzer comes with a few of different analysis services, which allow the web site owner to examine some statistics about the web site community. The user profiles and the different services are provided using Spring[14], a Java framework for the development of enterprise applications. The business objects, i.e., zones, topics, and users are queried from the database and mapped to objects using Hibernate[15], a framework for the storage and retrieval of Java domain objects via object/relational mapping.

We integrated the tool for building and analyzing similarity graphs as new service into the Gugubarra Analyzer as described in detail in [40]. We used the JGraphT[16] Java library because it provides data structures, graph algorithms as well as support for the visualization of graphs. Figure 5 shows the web GUI of the Gugubarra Analyzer with the new service. The web site owner can now analyze her/his community and compute the importance of the users. The most important users are show in the table on the top ordered by the different importance algorithms. On the bottom of the page, the web site owner can choose between different graph visualizations. With the "build graph" button the selected graph representation will be calculated and presented as a static png-image. The "build frame" button will present the graph as Java-applet where the web site owner can order the single vertices manually and adapt the appearance of the graph to her/his needs.

VII. Conclusion and Future Work

With the tool for building and analyzing similarity graphs, we provide a useful service to web site owners for analyzing their web community. We showed with an evaluation the applicability of our approach. We extended the web analytics software Gugubarra with this tool. Now, with the results of graph analysis, we are able to answer the research questions of Section IV:

- Which are the important users of the web community? We provide several algorithms (see Section III-D) to

[11] https://tapestry.apache.org/
[12] http://www.java.com/
[13] HyperText Markup Language, http://www.w3.org/html/
[14] http://www.springsource.org/
[15] http://www.hibernate.org/
[16] http://jgrapht.org/

calculate the important user(s) of the community. The definition of importance is dependent on the used algorithm and on a subjective point of view. 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 web community?
  The weights of the edges of the similarity graph represent the similarity of the users. These weights are calculated with the Manhattan Distance. Therefore, the lower the weights of the edges are 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 web 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 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 because these users can be considered as an archetype for her/his community.

Besides the extension of the tool with more algorithms for the similarity calculation, in future, the exploration for similarity (or importance) metrics would be helpful. With this type of metrics it would be possible to evaluate the similarity algorithms objectively.

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. Furthermore, our appreciation goes to Joanna Pieper and Mitra Shamloo for their implementation work.

REFERENCES

[1] C. Schefels, "How to find important users in a web community? mining similarity graphs," in *Proceedings of the First International Conference on Data Analytics (DATA ANALYTICS 2012) / NexTech 2012*. International Academy, Research and Industry Association (IARIA), 2012, pp. 10–17.

[2] "Facebook reports fourth quarter and full year 2012 results," Menlo Park, USA, January 2013, http://investor.fb.com/releasedetail.cfm?ReleaseID=736911, accessed: June 12, 2013.

[3] V. Gundotra, "Google+: Communities and photos," Mountain View, USA, December 2012, http://googleblog.blogspot.de/2012/12/google-communities-and-photos.html, accessed: June 12, 2013.

[4] N. Mushtaq, P. Werner, K. Tolle, and R. V. Zicari, "Building and evaluating non-obvious user profiles for visitors of web sites," in *IEEE Conference on E-Commerce Technology (CEC 2004)*. Washington, DC, USA: IEEE Computer Society, July 2004, pp. 9–15.

[5] 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)*. Washington, DC, USA: IEEE Computer Society, July 2008, pp. 190–197.

[6] M. Trusov, A. V. Bodapati, and R. E. Bucklin, "Determining influential users in internet social networks," *Journal of Marketing Research*, vol. 47, no. 4, pp. 643–658, 2010.

[7] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *Proceedings of the 19th international Conference on World Wide Web*, ser. WWW'10. New York, USA: ACM, 2010, pp. 981–990.

[8] J. Wang, Z. Chen, L. Tao, W.-Y. Ma, and L. Wenyin, "Ranking user's relevance to a topic through link analysis on web logs," in *Proceedings of the 4th International Workshop on Web Information and Data Management*, ser. WIDM '02. New York, USA: ACM, 2002, pp. 49–54.

[9] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, September 1999.

[10] 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 (ASONAM '10)*. Washington, DC, USA: IEEE Computer Society, 2010, pp. 281–285.

[11] G. Paliouras, "Discovery of web user communities and their role in personalization," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1-2, pp. 151–175, 2012.

[12] C. Schefels, "Analyzing user feedback of on-line communities," Ph.D. dissertation, Goethe-University Frankfurt am Main, 2012.

[13] N. Hoebel, S. Kaufmann, K. Tolle, and R. V. Zicari, "The gugubarra project: Building and evaluating user profiles for visitors of web sites," in *HotWeb 2006 - First IEEE Workshop on Hot Topics in Web Systems and Technologies*. Washington, DC, USA: IEEE Computer Society, November 2006, pp. 1–7.

[14] C. Schefels, S. Eschenberg, and C. Schöneberger, "Behavioral analysis of registered web site visitors with help of mouse tracking," in *Proceedings of the 14th IEEE International Conference on Commerce and Enterprise Computing (CEC2012)*. Los Alamitos, USA: IEEE Computer Society Press, 2012, pp. 33–40.

[15] C. Schefels and R. V. Zicari, "A framework analysis for managing explicit feedback of visitors of a web site," in *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS2010)*. New York, USA: ACM, November 2010, pp. 481–488.

[16] 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.

[17] 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 (IJCAI'03)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 43–48.

[18] 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 (ICDT '01)*. Berlin, Germany: Springer, 2001, pp. 420–434.

[19] 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 *Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing (CEC2009)*. Washington, DC, USA: IEEE Computer Society, July 2009, pp. 265–272.

[20] 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.

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

[22] R. J. Wilson, *Introduction to Graph Theory*. London, UK: Longman, 1979.

[23] D. Jungnickel, *Graphen, Netzwerke und Algorithmen*, 3rd ed. Mannheim, Germany: BI-Wissenschaftsverlag, 1994.

[24] B. Bollobás, *Modern Graph Theory*, ser. Graduate Texts in Mathematics. Berlin, Germany: Springer, 1998.

[25] M. A. Rodriguez and P. Neubauer, "Constructions from dots and lines," *Bulletin of the American Society for Information Science and Technology*, vol. 36, no. 6, pp. 35–41, August 2010.

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

[27] 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 (WWW7)*, vol. 30, no. 1–7. Amsterdam, The Netherlands: Elsevier, 1998, pp. 107–117.

[28] D. Nemirovsky and K. Avrachenkov, "Weighted pagerank: Cluster-related weights," in *Proceedings of The Seventeenth Text REtrieval Conference (TREC 2008)*. Gaithersburg, USA: National Institute of Standards and Technology (NIST), November 2008.

[29] 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 (AAMAS '02): Part 1*. New York, USA: ACM, 2002, pp. 467–474.

[30] P. Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, February 1912.

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

[32] 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.

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

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

[35] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '01. New York, USA: ACM, 2001, pp. 57–66.

[36] J. Marcus, "Rgraphviz," Presentation, 2011, http://files.meetup.com/1781511/RgraphViz.ppt, accessed: June 12, 2013.

[37] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, July 2003.

[38] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Systems Technical Journal*, pp. 1389–1401, November 1957.

[39] N. Hoebel, "User interests and behavior on the web: Measurements and framing strategies," Ph.D. dissertation, Goethe-University Frankfurt am Main, 2011.

[40] M. Shamloo and J. Pieper, "Ähnlichkeitsgraphen und wichtige nutzer einer web-community," bachelor thesis, Goethe-University Frankfurt am Main, December 2012.