Grouping Facebook Users Based On The Similarity Of Their Interests

Novak Boškov*, Marko Jocić*, Đorđe Obradović*, Miloš Simić*
* University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
gnovak.boskov@gmail.com, m.jocic@uns.ac.rs, obrad@uns.ac.rs, milos.simic@uns.ac.rs

***Abstract* — In this paper we propose a simple method for measuring the similarity of interests between Facebook users and grouping them with this measure in mind. The ultimate goal is to build a web application which will integrate with Facebook and offer users the ability to find other users similar to them, starting from the premise that users are similar as much as their interests match. A user of this application will be able to look through a list of similar users (sorted by descending degree of mutual similarity) and see the particular shared interests with each of those similar users.**

I. INTRODUCTION

Social networks have emerged as a large factor in information dissemination, search, marketing, expertise and influence discovery, and are potentially an important tool for mobilizing people. Social media has made social networks ubiquitous, and has also given researchers access to massive quantities of data for various types of analyses. These data sets are a strong foundation for studying the dynamics of individual and group behavior, the structure of networks and global patterns of the flow of information between them. Sometimes, the shape of underlying networks is not directly visible, but can be inferred from the flow of information from one individual to another [1]. With the great expansion of Facebook and Twitter (which appeared in 2004 and 2006, respectively), various parties have shown their need for tracking the behavior and activities of social media users, from the founder companies to data scientists and enthusiasts. Data collected from social media can be used for a variety of purposes, so every party which investigates it may have its own point of view and method of interpretation.

Large amounts of data collected from social networks have found their role in social media marketing. By some definitions, "social media marketing" encompasses advertising and promotional efforts that use social media Web sites [2]. It is a form of viral marketing, a term coined by Harvard professor Jeffrey F. Rayport in 1996 to illustrate how a message spreads through an online community rapidly and effortlessly [3]. Marketing agencies are heavily involved in social media data analysis, and are mostly occupied with collecting large amounts of such data, which is more or less publicly available. The majority of such agencies base their insights into the state of the market on knowledge gathered from a variety of social networks. Groups of users which show similar aspirations on the network may adopt a particular product with greater success than a group which is composed of people with completely different viewpoints.

Social media companies also have the benefit of collecting data for themselves. Twitter and Facebook make a large part of their profit through the sale of such data, or conclusions made from the analysis of it. On the other hand, they offer some kind of recommendation services to their users, which rely on paid advertising. Also, users are given the opportunity to find new connections inside or outside of the boundaries of a particular social network, and this mechanism is also based on the analysis of user data which could be collected by the companies themselves.

In increasingly frequent situations nowadays, data originating from social media can be an object of interest to public authorities, such as the police, as well as a range of government agencies, because of a possibility of fraud such as impersonation, pedophilia and sex trafficking.

In this paper, the focus is on Facebook, because it is the world's most popular social network, and makes available an application programming interface, or API - Graph API [4], through which various data from Facebook can be easily consumed, and later analyzed. Also, this research uses some user information which is not publicly available, and for that purpose utilizes appropriate Access Tokens [5].

This paper is laid out in the following chapters: the first chapter gives an introduction into the research conducted in this paper, as well as the motivation for the research. Chapter II gives an overview of some related work. Chapter III outlines basic facts about data model and the possibilities it offers, together with a brief description of Facebook's Graph API, while Chapter IV describes the chosen method of determining the measure of similarity between Facebook users. Chapter V shows the results of the proposed method, concluding with directions for future research.

II. RELATED WORK

As noted, this section deals with existing related work in the area of finding similar social network users. Since the people who have conducted research in this field so far have different backgrounds, they have yielded few possible solutions distinct by approach. In this chapter we will present three classes of approaches: sociological approaches in subsection A, approaches based on queries in subsection B, and finding clones in subsection C.

A. Approaches relying on social relationships

This class is based on social relationships existing among users. In many cases, this information is instrumental in producing suggestions (e.g. for friendship relationships or community membership). In particular, the approach shown by Spertus, Saham & Buyukkokten in 2005 [6] analyzes the affiliation of users to multiple virtual

communities and tells them if they should join new ones. For this purpose, their approach considers Orkut, once a big social network operated by Google, as a reference scenario, and experimentally compares the effectiveness of a range of techniques for computing user similarities (e.g. tf-idf coefficients, or parameters coming from Information Theory).

The *AboutMe* system by Geyer, Dugan, Millen, Muller & Freyne from 2008 [7] is able to complete the profile of a user *u* by examining the list of topics used by his acquaintances in a social network. Resulting profiles are more accurate, and ultimately, are relevant to enhancing user participation in social activities.

The approach of Groh & Ehmig from 2007 [8] suggests using the friendship lists to identify resources relevant to them. In particular, this approach handles the friendship list of a user *u* and the ratings of the users on these lists assigned with an object *o* to predict the rating that u would assign to *o*.

The effectiveness of these approaches, however, crucially depends on the number of social relationships created by users. In fact, if a user is involved in few friendship relationships, the information at our disposal is poor, and thus the quality of suggestions will inevitably be poor [9].

B. Approaches based on queries

To model similarity, this class of approaches assumes the existence of a set of queries, and two users are deemed similar if their answers to these queries are (mostly) identical. Technically, each user has a vector of preferences (answers to queries), and two users are similar if their preference vectors differ in only a few coordinates. The preferences are unknown to the system initially, and the goal of the algorithm is to classify the users into classes of roughly the same preferences by asking each user to answer the least possible number of queries. This type of method can prove nearly matching lower and upper bounds on the maximum number of queries required to solve the problem. Specifically, it presents an "anytime" algorithm that asks each user at most one query in each round, while maintaining a partition of the users. The quality of the partition improves over time: for **n** users and time **T**, groups of $\tilde{O}\ (n/T)$ users with the same preferences will be separated (with high probability) if they differ in sufficiently many queries. Also, it presents a lower bound that matches the upper bound, up to a constant factor, for nearly all possible distances between user groups [10].

C. Finding clones

The viewpoint of this approach is that social networks provide a framework for connecting users, often by allowing one to find his preexisting friends. Some social networks even allow users to find other users based on a particular interest or tag. However, users would ideally like the option of finding people who share many of their interests, thus allowing one to find highly similar, like-minded individuals whom they call "clones". This approach explores a means for finding "clones" within a large, sparse social graph, and it presents its findings that concern patterns of shared interests. Additionally, holders of these ideas explore how this correlates with connectivity and degrees of separation in a social graph. Since analyzing social media deals with large amounts of data, the number of possible clones in a given dataset is enormous (for **n** users in dataset there is $\frac{n^2}{2}$ possible clones). Computing similarity values for each of these pairs is practically infeasible. Hence, this approach offers a way to first detect candidate pairs for clones, and later compute the actual similarity. To generate candidate pairs Min-Hashing and Locality Sensitive Hashing (LSH) may be employed [11]. This improvement is a valuable contribution of this approach, and can be used in other methods as well.

III. PRELIMINARIES

The approach proposed in this paper consists of using n-dimensional vectors to represent Facebook users and calculate mutual similarity between each possible pair using a few simple methods for measuring the distance between two vectors in n-dimensional space. Since this paper is focused on Facebook, it is necessary to give an overview of the fundamentals of the way Facebook content (especially pages) categorization works.

A. Facebook pages categorization

Facebook categorizes its content by employing a wide range of methods, and the most important method in this paper is Facebook Page categorization. When a user or organization wants to make a Facebook page, it must provide a short description and a couple categories which best suit it. Hinging on entered categories, each page may be classified as one or more categories. Each category can succeed some categories, as it can be placed as the ancestor of none or plenty of other categories. Thus, if we pay attention to Facebook's official page, we will see that it belongs to two categories - Product and Service. Since Facebook users are inclined to "like" pages, Facebook records such behavior. This implies that every page "like" a user has executed is stored somewhere in Facebook's system, and can be accessed through its services, which are available to third-party Facebook applications. Using the appropriate permissions, we are capable of using that data to represent a user. Hence, the set of all known Facebook user interests in this work consists of all categories currently present in the system and dynamically increases through the registration of new users. When a new user joins the system, categories of his likes are filtered, and all the categories that are not present in system yet will be added afterwards.

B. Modeling user

All approaches mentioned in the previous chapter can utilize n-dimensional vectors as a model of a concrete user on Facebook. N-dimensional vectors are encouraged as the main data structure through this work, and all algorithms will be executed against them. The problem of measuring the distance between two n-dimensional vectors comes as a logical continuation. The following subsections deal with

the aforementioned issue by presenting the advantages and disadvantages of each algorithm.

C. *Representing users similarity in terms of geometry*

Presume that our system currently knows only about a few Facebook categories, given in the form of the vector below:

$$[Movies, Sport, Television, Music, Activism, Software]$$

This vector can be treated as a set of dimensions in 6-dimensional Euclidean space. Thus, each entity that we want to compare, or possibly sort, become a point positioned in the considered coordinate system, whereby each coordinate of this entity (represented by a corresponding category) may take a value of 1 if the corresponding category is present, or 0 if it is absent. Suppose now that one user registered in our system has "liked" a page of some movie, and a page of some band. We can then represent him with the vector below:

$$[1, 0,0,1,0,0]$$

In accordance with that, we can now also assume that a user who has "liked" 3 different movies, 2 distinct sport clubs, and 7 bands, in that case its corresponding vector can consist from some other positive integers and zeros. One such vector could be:

$$[3,2,0,7,0,0]$$

The next issue is how to calculate the distance between two points in the described n-dimensional coordinate system, in which n is representing number of categories currently known to the system. In order to achieve a measure of similarity, we can measure the angle formed by two vectors representing users, or directly calculate the Euclidean distance between them. In a two-dimensional coordinate system, this problem (shown graphically in Figure 1) is easily recognizable.

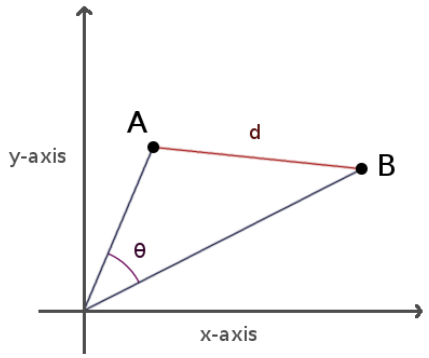

Figure 1. Distance between two points in two-dimensional Euclidean space

D. *Euclidean distance*

The formulation of Euclidean distance is straightforward, and if we assume that we have only two points, the Euclidean distance is given by following equation:

$$distance(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + \cdots + (a_n - b_n)^2}$$

Now, we can conclude the following: as the distance between two vectors decreases, the mutual similarity of users represented by those vectors is increases. This approach imposes a question which follows:

- *The problem (or strength) of Euclidean distance*

Presume that the system currently possesses structure categories that represent all known Facebook categories:

$$categories = [Algorithms, Programming, Mining, Python, Ruby,$$

$$Ruby, Publishing, Server, Cloud, Heroku, Jekyll,$$

$$GAE, Web, Design, UX, Android, API]$$

And that there is one referent vector against which we want to measure distance (given as referent in the following equation) in the coordinate system defined by the *categories* structure:

$$referent = [Publishing, WEB, API]$$

And vectors A, B, C and D are those whose distance is going to be measured:

$$A = [Algorithms, Programming, Mining, Python, Ruby]$$
$$B = [Publishing, Server, Cloud, Heroku, Jekyll, GAE]$$
$$C = [Web, Design, UX]$$
$$D = [Python, Android, Mining, Web, API]$$

Then, assuming that the distance is calculated using Euclidean distance, we can notice that vectors D and C have exactly the same distance from the referent vector, although vector D shares two categories with the referent vector, while C shares only one. Such behavior is not preferable for our goal, and because of that, Euclidean distance is not chosen in our method.

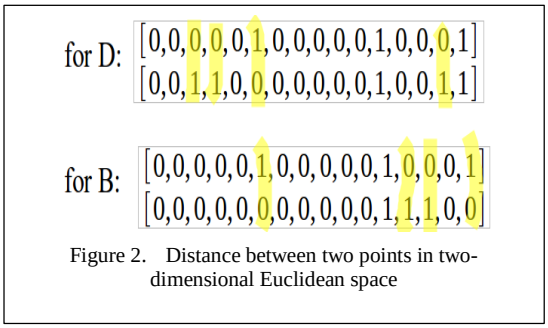

Figure 2. Distance between two points in two-dimensional Euclidean space

Let's see what evidence induces this kind of behavior. In Figure 2 the referent vector is shown above and vectors D and B are shown below.

It is evident that vectors D and B both differ from the referent vector at four positions. Therefore, we can conclude that Euclidean distance actually measures mutual diversity rather than similarity. Those coordinates that are the same are less significant than the differing ones.

E. Cosine Similarity

This method is comparable with the previous one, but produces different results since it measures mutual similarity rather than diversity. This way of calculating similarity is given by the following equation:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_1^n A_i B_i}{\sqrt{\left(\sum_1^n {A_i}^2\right)}\sqrt{\left(\sum_1^n {B_i}^2\right)}}$$

This means that the mutual similarity of two considered vectors is equivalent to the cosine of the inclination angle between them. This method yields much better results for this study.

The value of the similarity in this approach is between -1 and 1. If the similarity is equal to -1, then the considered vectors are completely different, and if it is equal to 1, then they are completely similar.

F. Pearson Correlation Coefficient

Pearson Correlation Coefficient (sometimes referred to as as the *PPMCC* or *PCC* [12]) when used as a method for measuring the similarity of n-dimensional vectors, represents a slightly more sophisticated technique which actually measures the linear correlation of the considered vectors. Actual similarity is calculated by the following equation:

$$similarity = \frac{\sum_1^n a_i b_i - \frac{\sum_1^n a_i \sum_1^n b_i}{n}}{\sqrt{\left(\sum_1^n {a_i}^2 - \frac{(\sum_1^n a_i)^2}{n}\right) \cdot \left(\sum_1^n {b_i}^2 - \frac{(\sum_1^n b_i)^2}{n}\right)}}$$

G. Social graph

The social graph in the Internet context is a graph that depicts the personal relations of Internet users. In other words, it is a social network, with the word “graph” taken from graph theory to emphasize the rigorous mathematical mathematic analysis applied (as opposed to the relational representation in a social network). In the words of Mark Zuckerberg, the founder of Facebook, the

social graph has been referred to as “the global mapping of everybody and how they're related” [13]. The term was popularized at the Facebook F8 conference on May 24, 2007, when it was used to explain how the newly introduced Facebook Platform would take advantage of the relationships between individuals to offer a richer online experience [14]. The definition has been expanded to refer to a social graph of all Internet users. Figure 3 given below to illustrate the structure of a social network. [15].

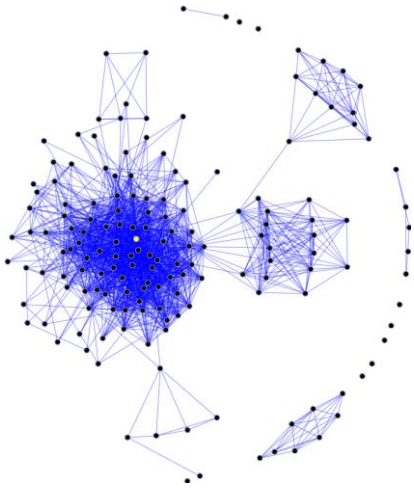

Figure 3. A social graph in which each person is represented by a dot and the friendship relationship is represented by a line.

H. Facebook Graph API

Facebook's Graph API is the primary way for applications to read and write to the Facebook social graph [4]. The Graph API has multiple versions available, and for this research, version 2.2 is used. This API is Facebook's official mechanism for making it possible to perform many different queries, such as fetching data about certain events, apps, groups, pages, users, as well as as user-specific data, such as a user’s timeline, his friends, photos, etc. A complete list of root nodes of the Graph API version v2.2 is available on https://developers.facebook.com/docs/graph-api/reference/v2.2.

Graph API is secured by an authentication flow based on the OAuth 2.0 protocol exposed through Facebook Login. Therefore, every application using the Graph API must provide a valid access token digested through noted authentication flow. The received access token is a so-called short-lived access token, and it expires after approximately 2 hours. However, this short-lived access token can be exchanged with a long-lived access token, which expires after 60 days.

I. Collecting the data used in this research

This work uses User Access Tokens, which means that users must give their permission explicitly at the end of the Facebook Login process. Generally, at the first page of our application, users are faced with the proposed privacy policy, and only if they decide to agree with it and provide us their tokens are we able to obtain information from Graph API. This information consists of a list of the user's

liked pages and friends. Around 250 users have decided to give us their permission so far, and that is the data set we currently have.

IV. PROPOSED METHODS

This work proposes two possible methods, and currently both are offered to end users of the resulting application. The first one is based on the Pearson correlation coefficient, and the other utilizes a combination of cosine similarity and the Jaccard index [16]. Afer login, on the first page of our application, users are able to choose from these two algorithms, in order to be able to later compare the calculated results on their own. Giving this selection to the users is important for us because it provides us with a nice possibility of examining user impressions, which are valuable for analyzing the success of the final algorithm.

- *Method based on PCC*

In order to clarify this method, it would be helpful to presume a possible system state. One such state is described in the part that follows.

The system possesses knowledge of some of the categories, and a vector which describes all known categories is given below:

$$categories = [Algorithms, Programming, Mining, Python, Ruby]$$

There are also four users currently using the system:

- User A, who has liked 4 pages that Facebook categorized as "Algorithms", 5 pages categorized as "Programming", 6 pages in the "Mining" category, 7 "Python" pages, and 8 "Ruby" pages. This user is then described with following vector:

$$A = [4,5,6,7,8]$$

- User B liked same pages as A, except those which are categorized as "Mining". B is absolutely not interested in that sort of thing, it so is described with:

$$B = [4,5,0,7,8]$$

- User C has the same interests as B, but C liked one less page from the "Ruby" category. User C is described with:

$$C = [4,5,0,7,7]$$

- User D has the same set of interests as A, but his time spent on particular pages is inverted in relation to A, so D shows the most interest in "Algorithms", and the least on "Ruby". This user is described with:

$$D = [8,7,6,5,4]$$

For each of these four users, we want to measure his similarity with a fifth user R using the Pearson correlation coefficient. User R has the same set of interests as the previous four users, but his interests are distributed in the following order: 1 liked page from the "Algorithms" category, 2 from "Programming", 3 from "Mining", 4 from "Python" and 5 from "Ruby". Finally, User R is described with the vector:

$$R = [1,2,3,4,5]$$

This method yields the following results:

$$similarity(R, A) = 1.0$$
$$similarity(R, B) = 0.5063696835418333$$
$$similarity(R, C) = 0.4338609156373132$$
$$similarity(R, D) = -1$$

These results show that PCC can serve satisfactorily well for the purposes of this work.

- *Combination of Cosine similarity and Jaccard index*

All methods described in the previous parts of this paper do not care about exactly the same pages which are liked by both compared users, but only about page categories. This fact was the motivation for the introduction of the Jaccard index.

The Jaccard index, also known as the Jaccard similarity coefficient (originally coined coefficient de communauté by Paul Jaccard), is a statistic used for comparing the similarity and diversity of sample sets. It measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets as given in equation below:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Facebook itself uses algorithms which recommend users to each other if they have liked the same pages. The idea of combining cosine similarity with the Jaccard index strives to merge the good parts of Facebook's solution into the results of this work

V. RESULTS

In this chapter we show the results of applying the proposed methods to the sample of around 250 Facebook users. After examination of some user impressions we are concluded that both proposed methods are producing mostly the same results, people who are shown to be highly similar by using one method are also shown to be similar using the second one. A notable exception is that

combination of cosine similarity and Jaccard index is turned out to be more alike Facebook's algorithms for friend recommendations, in the most of cases this method finds that user which is listed as friendship recommendation by Facebook itself is also that one who is listed highly similar by applying this method. Next to that second proposed algorithm is notably slower because of a non optimized method of calculating Jaccard index. Performance of algorithms is shown in the table below:

	Page categories	Shared same-page likes	Facebook friendship suggestions	Overall algorithm speed
Pearson correlation coefficient	Important	unimportant	Rare	Higher
Cosine Similarity & Jaccard index	Important	Important	Often present in result	lower

VI. Conclusion

In this paper we proposed a method for grouping users on Facebook based on similarity of their interests. Two possible methods are taken in consideration: one based on Pearson correlation coefficient and another based on a combination of Cosine similarity and Jaccard index. Both of them can be used to establish groups of Facebook users who are similar by their interests, each of them has its advantages and disadvantages so the right one can be chosen according to requirements and conditions. Results can be useful for sociologist as well as marketers who explore behavior of people on social networks. Further advancement of the proposed methods could be developing of a categorization system independent from Facebook's own categorization which will result in more righteous results. Also, mentioned algorithms and user-representing data structure could be optimized in order to increase the overall speed.

References

[1] Lerman K., Ghosh R., Information Contagion: An Empirical Study of the Spread of News on Digg and Twitter Social Networks, 2011, [Online]http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1509/1839/ [Accessed 30-Jan-2016]

[2] Bosman J., Chevy Tries a Write-Your-Own-Approach, and the Potshots Fly, April 4, 2006, [Online]http://www.nytimes.com/2006/04/04/business/media/04adco.html?_r=1&;oref=slogin [Accessed 31-Jan-2016]

[3] Rayport J., The Virus of Marketing, [Online]http://www.fastcompany.com/27701/virus-marketing [Accessed 31-Jan-2016]

[4] Facebook Graph API ,[Online]https://developers.facebook.com/docs/graph-api [Accessed 31-Jan-2016]

[5] Access Tokens , [Online]https://developers.facebook.com/docs/facebook-login/access-tokens/ [Accessed 31-Jan-2016]

[6] Spertus E., Sahami M. & Buyukkokten O., Evaluating Similarity Measures: a Large-scale Study Inthe Orkut Social Network. In Proc. of the ACM SIGKDD international conference on Knowledge discoveryin data mining (KDD '05), pages 678-684. ACM Press., 2005.

[7] Geyer, W., Dugan, C., Millen D.R., Muller M. & Freyne J., Recommending Topics For Selfdescriptions in Online user profiles. In Proc. of the ACM conference on Recommender Systems (RecSys'08), pages 59-66, Lausanne, Switzerland, 2008.

[8] Groh G. & Ehmig C. , Recommendations in Taste Related Domains: Collaborative Filtering vs.Social Filtering. In Proc. of the International ACM conference on Supporting Group Work (GROUP '07), pages 127-136, 2007.

[9] Pasquale De Meo, Giacomo Fiumara (Department of Physics, Informatics Section, University of Messina, Italy), Emilio Ferrara (Department of Mathematics, University of Messina, Italy), Finding Similar Users in Facebook, , [Online]http://cogprints.org/7634/1/Finding_similar_users-CR.pdf / [Accessed 31-Jan-2016]

[10] Aviv Nisgav , Boaz Patt-Shamir (School of Electrical Engineering, Tel Aviv University), Finding Similar Users in Social Networks, , [Online]http://groups.csail.mit.edu/tds/papers/Patt-Shamir/TCS.pdf [Accessed 31-Jan-2016]

[11] Chris Tanner, Irina Litvin, Amruta Joshi (Department of ComputerScienceLos, Angeles), Social Networks: Finding Highly Similar Users and TheirInherent Patterns, , [Online]http://www.chriswtanner.com/papers/SocialNets_Finding_Highly_Similiar_Users_02_2008.pdf [Accessed 31-Jan-2016]

[12] Pearson product-moment correlation coefficient , [Online]https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient [Accessed 23-Jan-2016]

[13] Facebook: One Social Graph to Rule Them All? , [Online]http://www.cbsnews.com/news/facebook-one-social-graph-to-rule-them-all/ [Accessed 23-Jan-2016]

[14] Facebook Unveils Platform for Developers of Social Applications , [Online]http://newsroom.fb.com/news/2007/05/facebook-unveils-platform-for-developers-of-social-applications/ [Accessed 23-Jan-2016]

[15] Social graph , [Online]https://en.wikipedia.org/wiki/Social_graph [Accessed 30-Jan-2016]

[16] Jaccard index , [Online]https://en.wikipedia.org/wiki/Jaccard_index [Accessed 30-Jan-2016]