Data Mining for Social Networks

Proseminar Data Mining

Jona Neumeier
Deparment of Informatics
Technical University Munich
Email: neumeiej@in.tum.de

***Abstract*— In this paper we want to discover some of the key concepts of data mining or rather graph mining for social networks. We therefore first introduce some key elements of graphs and of social networks, especially online social networks. After that we will look at some basic techniques used to extract information out of massive network graphs. We will see how subgraph mining, link mining and other methods can give us useful information about community structures or influential people in the network. We also provide brief use cases about how the revealed facts and figures can be used, be it in line with the users' interests or not.**

***Index Terms*— data mining, graph mining, online social networks, social network analysis**

I. Introduction

Ever since humanity exists people have gathered and have formed networks. In the modern technology age these networks also appear in the online world, in form of *online social networks (OSN)*. Online social networks could be considered one of the main causes of the massive modern data production. Everyday billions of new data are generated by the users of OSN by liking, sharing and commenting. In this paper we want to give an overview of what a social network nowadays consists of. Never before humanity has been able to gather so much information about so many severe different people ranging from young to old in one place, as Simonite concluded. This carries huge opportunity, as we can study human behavior better than ever before [1]. OSN are also an effective tool for communication. People in the Arabian Spring used *Facebook* to organize demonstrations which later resulted in a revolution, is just one example [2]. Simultaneously, there are risks regarding privacy and anonymity. That is because never before had one company so much information about its clients [1]. Fact is that OSN are increasing in popularity and become a more and more important component of our day to day life. For this reason, we will discover how these massive amounts of information can be mined and used. Therefore, we look at graph representations of OSN and how these graphs can be mined for our cause. Finding frequent subgraphs, link mining or finding communities inside of a network are only a few examples of techniques used. All these techniques have a solid base in traditional graph theory [3], while some of them, at the same time, are very specific for graphs representing social networks.

In the first part of this paper, we present some necessary definitions for graphs and social networks. The following part will contain more detail about how data mining and graph mining are used for social networks and how the extracted data is or can be used.

II. Social Networks

A social network is a gathering of people with the same interests, motivations or goals [3]. These networks can appear offline in form of face-to-face interactions between individuals [4] or in the form of an online social network. In both forms hierarchical structures can appear as well as social roles for every individual [3]. The driving force behind such networks is that people like to associate with other people who they think are like them [3]. In this paper we concentrate on OSN or social network sites like they are referred to in [5]. The author Boyd et al. continued to describe social network sites as a web-based service in which individuals can display different information about them on a public or private profile. They further can form a variety of connections with other users and traverse through them [5].

Although each network is different and has their own terminology for each feature, nearly all of the large OSN share the following points:

- *Profile page*: On this page users can describe themselves. They can display their current work, education, housing and relationship situation as well as upload profile pictures and write short messages called "*status updates*" or "*tweets*".
- *Relations between the users*: They can range from befriended to one another, "*like*" something another person did or they "*follow*" someone or something to get their status updates.
- *Chats and Comments*: Two or more users of the network can exchange messges via a private chat or carry out a public debate in a comment section.

There is a OSN for everyone, ranging from the general public to niche groups. *Facebook* and *Twitter* are one of the most famous ones, the first of the two is used by hundreds of millions of people worldwide [3] and the number keeps growing. There are also networks to share solely pictures or videos (*Snapchat*, *Instagram* or *YouTube*), networks for business and work (*LinkedIn*), networks for programming and computer issues (*StackOverflow*) and many more to name. According to [5] many of the large OSN are not used to "network" in the original sense, they are rather used to stay in contact with people already in their social network, online or otherwise. The data generated by the users of an OSN by liking, sharing and connecting with others, is of great value for the owner of the social network. With the help of *data mining* this can be used to gather information for a specific user or for a whole community inside the network. The network then can place advertisements specifically targeted for one community or generate predictions of whom you could also befriended with or like [4]. This topic is highly controversial, because of the privacy concerns some user groups and institutions have. Certainly, it is a subject of many occupations as it combines fields like psychology, sociology, statistics and graph theory in one [3].

Figure 1 is an example of a very small network. Each node represents an individual of the network and the connections between them show the relationships two nodes share. Therefore node *A* and *B* are friends and node *C* is following *A*. There are only two types of relationships in this example, "*following*" and "*friendship*". In a real network there are thousands of similar connections. Further, we can see that two nodes can share multiple connections.

In the following we will present what types of information can be extracted out of user profiles, generated metadata and pictures. There seems to be a focus on social networks concentrating on uploading and sharing pictures and videos, so we can use these techniques to gather a lot of information. Of course there are far more things to mine out of a social network, some of them are for example the movement pattern of individuals or simply the information that the user displays at his or her own profile. The data also reveals information the user usually is not aware of [3], for instance the relationship "friend of a friend of a friend". The next three sections discuss three methods that can be used to gather information from the users's activities. Afterwards we will concentrate on graphs and mining them for the same purpose.

1) Texts: Reddit is an example of a network which is highly focused on simple texts. Further, on every OSN the user can "*comment*", "*tweet*" about something or *chat* with other individuals. As said by Akaichi et al., these unstructured texts can be mined. The method used for this is called text mining [2]. The goal of this is to retrieve patterns and information out of these texts [2]. With the help of sentiment analysis, we can mine the opinions of the participants of the OSN, by dividing texts into positive, negative or neutral ones [2]. Therefore, distinguishing the texts to mine between facts and opinions [2]. A practical example given in [2] is that studies can be done without a huge amount of money, because we can get a lot of information by simply publishing our product or hypothesis and then improve it with the help of the resulting criticism. Therefore Akaichi crafted a five step method to analyze texts in [2], we now briefly look at each step:

1) Collect and extract the raw data (comments and status updates) from the network.
2) Build three lexicons: one for social acronyms, one for emoticons and one for interjections, where each item in the lexicon is mapped to a sentiment.
3) Preprocessing phase: extract key features out of the text.
4) Cunduct a training set.
5) Apply machine learning methods to mine and categorize the remaining texts in the network.

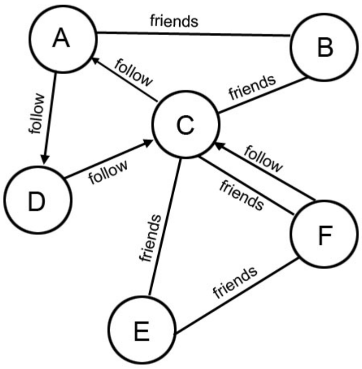

Figure 1. Example graph for a small social network with directed edges (follow-relationship) and undirected edges (friends-relationship).

The performance of this technique is described as high in accuracy [2] and is therefore a valid method to analyze statuses in OSN.

2) Pictures: Image mining aspects are rising in importance, especially when we look at the large amount of images, present in an OSN. Examples for this are *Instagram* or *Snapchat*. Thus, we will outline a few methods and applications for image mining in social networks. The goal of image mining is the discovery of patterns in a set of pictures [6]. One area of use is the recognition of a user in a photo uploaded by another user, without tagging any user on the picture. This is accomplished through face recognition. With this the network can link images to certain profiles to gather further information about the individuals. Object recognition is the process of finding known objects to the system in other images [6]. This could be seen as a labeling process [6] and is also the technique used for the example above. A second example of image mining applied on OSN is the detection of unauthorized uploaded pictures by users [6], so the owner of the network can take the photo down or initiate legal consequences. This can be achieved through the technique of classifying and clustering pictures with the same features [6]. This is done with a pre-labeled set of images, in terms of the contents of the picture [6]. Then machine learning algorithms are used to cluster unlabeled images to the right pre-clustered ones [6].

3) Metadata: Metadata is data about other data [7]. It can include timestamps, sender and receiver of a message, a location and the type of data that has been sent [7], but not the data itself. This can be a useful tool if we want to show that two people had contact, for instance. We can also form a movement pattern or can predict where one individual is living or working, when the metadata reveals the location when the user interacts with the network. The only thing we have to do is to analyze from which location the user most often logs on to the network.

III. Graphs

One way to visualize a social network is by using *graphs*, as we will in this paper. The reason for this is that the structure of a social network with its individuals and connections between them can easily be mapped to a graph, since a graph informally is a collection of nodes, which might be connected through edges [8]. As mentioned in [3], we are more interested in graphs as a datatype and not as a mathematical entity. Consequently, a graph $G = (V, E)$ consists of a set of vertices $V(G)$ and a set of edges $E(G)$ between some of the vertices [3]. Furthermore, a graph is made up of one or more subgraphs $H \subseteq G$ defined with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ [9]. We can distinguish between two aspects in relation to graphs: First, we can differentiate between directed and undirected graphs [3], [9]. In a graph with undirected edges there is no information about the direction of flow between nodes [3], which is different to a graph with directed edges where the flow can only go from the tail node to the head node [9]. Both forms appear in a social network. A directed relation could be that one user follows an important or famous individual, without the important person following back. Other than the ”friends” relationship, where the relation is bi-directed. Second, there are graphs with weights and without weights [3], [9]. These are just numbers written on the edges that represents information according to the domain we are working in - for example the distance or the costs [3]. Characteristics of a graph are the *average degree of the vertices*, with the degree of one vertex defined as the number of edges incident to this vertex [9]. The *diameter* of the graph, as $\max_{u,v \in V(G)}$ $\text{distance}(u, v)$ or the number of *isomorphisms*, as a graph H is isomorphic to a graph G if there is a bijection $f : V(G) \to V(H)$ and if $f(u)f(v) \in E(H)$ with $u, v \in E(G)$ [3], [9]. This simply means that a graph H is isomorphic to a graph G if they match exactly [3]. Further characteristics are the *average path length* between the nodes and the *cluster coefficient*, which indicates a nodes level of connectivity with its neighbors [3]. Nettleton further describes the cluster coefficient as a central property for a community within a social network and that a general graph of an OSN is formed out of many small *communities* [3]. With an algorithm that can find the vertex with the highest cluster coefficient, we can unveil the most connected person in a social network, which can be of great importance as we will discuss later on.

Because of the large networks and thus the large graphs we want to analyze, the representation of graphs inside a computer is a key point for efficiency and computational cost [3]. Mostly used are the *adjacency lists* and the *adjacency matrices* [3]. The adjacency matrix is an n-by-n matrix, with n as the number of nodes in the graph, where the entries are either the weight of the edge or a 1 if there is an edge or 0 if there is no edge between to vertices [9]. The adjacency list holds the exact same information only represented as a list [3]. Similar to adjacency matrices and lists there are incidence matrices and lists with the difference that they store the information if one vertex is an endpoint of an edge [3], [9]. There are far more characteristics which can be found in [9], but these are the most important ones for our task to analyze social networks.

Again, we can see most of the characteristics of graphs in Figure 1. For example, the nodes and the edges between them. We can distinguish the directed edges, between node *A* and *D* for instance, and undirected edges, between node *E* and *F*. In this graph the edges do not have weights. In the real world the weight of the edges might indicate the level of friendship according to various factors. Further, we can easily see that the node *C* has the most in- and outgoing edges, with a $\text{degree}(C) = 6$.

In the following we will see, how we can extract the information we want out of the data type described in this section.

IV. Graph Mining

With data mining we try to gain knowledge of different data sets, like the structure or general rules that apply [3], [7]. Generally, the information that can be gathered is not obvious for humans [3]. Graph mining is a subsection of data mining, simply applied on graphs [3]. Next we want to discover how these large amount of information can be mined from graphs. We mainly look at methods that are applicable for OSN, therefore not every method is shown below.

A. Mining Frequent Subgraphs and Substructures

As stated earlier, an OSN is formed by a high number of communities. In consequence it is interesting for us what these groups of people might have in common [8]. For that reason we have to find subgraphs, which can represent such communities, that occur multiple times in the whole graph. We can also search for abnormal subgraphs, which alter from the other frequent subgraphs that we found [8], and investigate them. According to the ninth chapter of [7], the discovery of frequent subgraphs is made up of two steps. Within in the first step we try to find candidates for a frequent subgraph. The candidate is then used to check how often the candidate occurs in the overall graph. An example is given in Figure 1, a potential subgraph of a larger graph might be the subgraph that is build of the nodes *A*, *D* and *C*. The second step involves an isomorphism test and is therefore very expensive in terms of computational costs (NP-hard), which is why most studies focus on the first step [7]. There are several algorithms and principles to detect and mine frequent isomorphic subgraphs and subpatterns, like the a-priori based approach, the pattern-growth approach, greedy search approach or the inductive logic programming (ILP) approach [3], [7], [8]. The techniques used for this are either the *breadth-first search* or the *depth-first search* [3], [7]. Because it is hard to find exact matches of the same subgraph there are also techniques that find invariants of frequent subgraphs [7]. An invariant of a subgraph is similar to the frequent subgraph candidate, but not isomorphic [3]. This can also be used as a pre-filter for the actual isomorphic subgraph detection, mainly used for reducing the complexity and therefore reducing the computational cost [3]. For example, a similar subgraph might distinguish through a different number of vertices or edges in a specific range, the degree of certain vertices or the number of cyclic loops in the graph [3], [7]. Two example techniques for mining similar frequent subgraphs of sub-patterns, as reported by Han et al., are:

- *Mining Closed Frequent Substructures*: The problem this method tries to solve is that in very dense graphs there tends to be a high number of subgraphs, especially because of the recursive behavior of the subgraph. A solution to this is to mine closed frequent subgraphs. This technique tries to mine only subgraphs which are not inside of a bigger subgraph. So a graph is closed if it is the largest subgraph in the close environment.
- *Mining approximate frequent substructures*: Here we can declare one approximate substructure which we like to mine, the algorithm then searches for other subgraphs with slight structural changes. This is to reduce the number of patterns to mine.

Stated by Han et al., indexing is a general known way to perform searches in an effective way. The following solutions for the indexing problem are proposed in [7]. Amongst other things the mining of frequent subgraphs and substructures can be used to reduce the complexity of indexing a graph. An index based on the vertices or edges is very ineffective, because the number will explode in a fairly large network. Indexing on frequent subgraphs is also not an optimal solution, however a better one than the one formerly described.

Also given in [7] is the following instance of using frequent subgraph mining. The analysis and classification of clusters inside a network graph can also be realized through the detecting of frequent subgraphs. We can examine features that are highly present in one subgraph but rather non-existent in another, which can form the basis of a classification of these subgraphs/clusters. In addition, if two or more subgraphs share the same patterns or smaller subgraphs it can be a qualified property to keep these graphs in one cluster class [7].

In the following part of this paper, we want to take a closer look on one approach to find frequent isomorphic subgraphs, namely the *a-priori based approach*.

As seen in the Algorithm 1, taken from [7], the algorithm takes a graph data set, a minimum threshold and returns the frequent substructure [7]. This is just the overall outline for an algorithm with the a-priori based approach. Specific algorithms like the AGM or the FSG operate using a variation of the above shown approach [7]. The algorithm starts with a small sized graph and adds another vertex, edge or path with each iteration [7]. As we can see on line 4 the algorithm merges the current and the previously detected subgraph together and then checks the frequency of this new candidate for a frequent subgraph [7]. We stated earlier that the creation of a useful candidate is the hard part in this approach The computational cost is very high, as there are three interleaved for-each-loops and a recursive call in each run of the algorithm, which is the reason the algorithm is in the complexity class $O(n^3)$.

Algorithm 1 AprioriGraph: Finding frequent subgraphs with the a-priori based approach [7].

Input:

- D, a graph
- minSup, the minimum support threshold (indicator of how dense the subgraph should be)

Output:

- S_k, the frequent subgraph

Method:

- S_1, frequent single-elements in the graph data set

```
CALL AprioriGraph($D, minSup, S_1$)
 1: procedure APRIORIGRAPH($D, \mathrm{minSup}, S_k$)
 2:     $S_{k+1} \leftarrow \emptyset$
 3:     for each frequent $g_i \in S_k$ do
 4:         for each frequent $g_j \in S_k$ do
 5:             for each $\mathrm{size}(k+1)$ graph $g$
 6:               formed by the merge of
 7:               $g_i$ and $g_j$ do
 8:                 if $g$ is frequent in $D$ and $g \notin S_{k+1}$ then
 9:                     insert $g$ into $S_{k+1}$;
10:     if $S_{k+1} \neq \emptyset$ then
11:         AprioriGraph($D, minSup, S_{k+1}$);
12:     return;
```

B. Link Mining

With Link Mining we try to gather information from the relationships formed by the participants of the network, instead of getting the knowledge from the object or an individual in a social network [7]. Hence we use graphs to represent our social network we can mine the edges for this purpose [7]. There are several tasks than can be accomplished with this method. First, we can classify an object with its links [7]. Unlike normal classification, link-based classification predicts the category of an object, in our case a user of a social network, upon its links instead of its attributes [7]. In a network for academic papers, for example, we can predict the topic of a paper by simply looking on the citations [7]. In this case, the citations of the paper are the links, which can also be formed if the paper gets cited by another paper [7]. Other than the next approach where we try to predict the type of the link rather than the type of the object [7]. With this we can predict if two people in a network are ”related”, ”friends”, ”friends of friends” or ”coworkers” for example. We can also use link mining to predict if there even is a connection between two entities or not [7]. This can be useful for marketing, as one instance, if we know someone associates to a certain topic, we can target him or her directly with advertisements about this topic. Further, we can estimate the cardinality of an entity in our social network, which we can divide in out- and ingoing links [7]. We can drive solutions from that such as identifying a famous or influential person on the network. If this person has a lot of ingoing links, we then can investigate this person.

Information that can also be disclosed from link mining is the object reconciliation [7]. Here we try to identify if two objects are the same, based on their attributes and links [7]. This is a useful tool for eliminating duplicates [7], which can occur when a user creates multiple accounts without deleting the old one, for instance.

C. Communities

Finding and extracting communities from a social network graph is a highly discussed topic with regard to graph mining [3]. We will see how we can define a community, what algorithms exist for extracting them and what we can use this technique for in relation to gathering information. We first want to define what a community is, which is not as easy as it may appear [10]. Nonetheless, a general definition could be that a community is a part of the graph where the connections between the vertices are denser than in or towards the other parts of the graph [3], [10]. But if we want adequate information we have to know the field in which the network is active, so that we can define which connections are significant for our purpose [10]. For instance, if we like to extract a community out of a OSN whose property is that every user within the community likes a specific movie, we do not want to look at all the other possible connections. We only want to consider the connections a user likes the movie, which is indicated through a like or something similar. Information we do not want to regard is if there are friends or a whole subnetwork of friends within this community. The authors of [10] have worked out two mathematical definitions for defining a community, which we will consider in the following. They define a community in a strong sense as

$$k_i^{in}(V) > k_i^{out}(V), \forall i \in V. \tag{1}$$

With Graph $V \subset G$ as a subgraph of Graph G, k_i as a degree of a node i within V, k_i^{in} as the degree of ingoing edges and k_i^{out} of outgoing edges. This simply describes that the nodes in a community are more connected than they are to nodes outside of the community. Similar to the strong community definition, a weak community is defined by the authors, as

$$\sum_{i \in V} k_i^{in}(V) > \sum_{i \in V} k_i^{out}(V), \tag{2}$$

which indicates that the sum of all degrees in the community is greater than the sum of the degree of the rest of the nodes towards the network. It is also worth mentioning that the community in a strong sense is a subset of the community in a weak sense [10]. Basically, there are two main methods for algorithms to detect a community inside of a OSN graph [3], [10]. There is the *agglomerative* technique, which is a hierarchical clustering process. Here we start with all nodes but no edges between them and add an edge with each iteration, so that the community grows larger and larger [3], [10]. The other technique, called *divisive*, is the exact opposite [10]. We start with the whole graph and cut out edges with each iteration, until we have found our community [3], [10]. These two methods form the basis for almost every other algorithm. The study in [10] presents a new approach for a divisive algorithm, it only focuses on local quantities. A quantity is a measure to single out unwanted edges [10]. They tested their method on a network with $|V| = 15616$ (scientists) but focused on a subset of the graph with $|V| = 12722$. They claim that they constructed a dendrogram of this network in just 3 minutes on a desktop computer with a 800 MHz CPU. A *dendrogram* is a hierarchical tree, which represents the intra and inter community links [3], [10]. If we find one or more of such communities, we can extract a variety of different information about them and use them in several different ways. One example might be, that we examine core principles which are present in one or more communities in our network and apply these principles to an unpopular community, which we want to push in popularity. Another use case could be a feature that occurs on nearly every website, a recommendation system [8]. For instance, if we have found a community in which every user likes rock music, we can suggest interpreters to an individual based on the interpreters that other participants of this community listen to. We then can provide them with this suggestion, alongside with a possible shopping link. Paired with the next technique the finding of communites can be a powerful tool for marketing purposes.

D. Finding influential Individuals

Finding the most influential person in a OSN graph can be an important subject for several different fields, if we see it from a marketing point of view, for example. We can detect a community of a certain topic, then find an influential individual in this community and target this ”celebrity” for marketing purposes. With this we reach a whole network of people with the same interests by only addressing one person. We could then craft a deal with him and sponsor him with a product we want to sell, for example. We might also grow in popularity and influence ourselves and therefore mirror other popular individuals in the social network. Similar to communities, we are here looking for just one node with a very high degree, instead of a whole network with high connectivity [3]. But this problem is very high in computational cost, since it is a NP-hard problem [3]. Two heuristics for measuring the degree of connectivity and therefore the influence on the network, are described by [3]. These are the *degree centrality*, which is the degree of one node, and the *distance centrality*. The later measure states that a vertex with short paths to other nodes has a more likely chance to influence these adjacent vertices [3]. Generally, there are four variants to find the most influential individuals, as reported by Nettleton. The first is the *greedy search*, we also have the search regarding how high the degree of a node is or regarding the centrality degree of the node. Lastly, we have a *random search*. The influence of a person in an OSN is seldom accidental, but one can achieve influence through various different methods. Though, there are many aspects to consider, like the psychological or sociological aspect [3]. Nettleton, for example, describes in [3] that in the case of Wikipedia a user can be promoted to an administrator, thus has more influence on the network, by constantly contributing to the network in form of adding value and content. In Figure 1 we can examine the node *C* as the most influential, according to the number of followers. We can see that this individual is followed by the indiviuals represented as the node *D* and *F*. We can also take it to a second stage in which we also count in the friends or the followers of somone who follows the indivdual *C*. In our case *C* would be followed by all other nodes in the graph. The power an influential person can have on a network is outlined in [1] with the example of Facebook founder Mark Zuckerberg. He added the feature that one can show that he or she is an organ donor by simply clicking a box on the profile. Many did exactly this and as a result the donor enrollment increased by a factor of 23 in 44 states in the USA [1]. It is most likely, that many individuals saw this organ donor checkbox first on a profile of an influential person, they were following, and then continued to become a donator themselves. This shows the power and impact people and especially influential ones can have on the overall network and the society.

E. Evolution and Predictions

There are several characteristics regarding the evolution of social network graphs [8]. In the following we want to discuss schemes, that seem to appear in nearly every social network. These are *power laws* and *shrinking diameters* of the network [7]. Since networks often evolve at a very quick rate, we want to be able to predict how the graph of the network will look like in a few days, months or years. This is why we need models that are built on the principles of how a social network evolves [7].

At first we want to look at the power law. This law describes the increasing density of the network graph over time, much like the "the rich get richer" principle [3], [7]. It was believed that with a linear growing number of vertices the number of degrees grows accordingly [7]. But the power law states that the average degree in the network increases rapidly [7]. In consequence the number of edges is superlinearly compared to the number of nodes [7]. For OSN this means, that the connections formed by the individuals grows exponentially with every new member in the network. The power law is given by the following function:

$$p(k) = Ak^{-\gamma} \tag{3}$$

[3], [8]. With $p(k)$ as the probability that a node will have degree k, A and γ are constants and γ is referred to as the power exponent [3], [8]. To compute the power exponent is not simple, it is often based on assumptions [8].

Next we want to look at the phenomenon of the *shrinking diameter*, which states that with an increase in size of the graph, the average diameter is decreasing [7]. In line with [3], [8] the most common diameter in OSN is ≈ 6, which is fairly small for a big social network [8]. The scenario A is a friend of B, who is a friend of C in an OSN, transforms, according to the *shrinking diameter*, to the relationship A is a friend of C, without a detour over the individual B. Another occurrence of this is in an weighted graph, where the weight describes the level of friendship between individuals of the network, with 0 as the highest friendship state. This weight decreases with the growth of the network, in consequence of the *shrinking diameter*.

When we get a snapshot of a graph, we want to be able to predict which links will be added in the future, this is known as the *link prediction problem* [7]. The basic approach to this is to assign a connection weight: $\mathrm{score}(X, Y)\ \forall X, Y \in V$ [7]. The algorithm should then produce a list of $\mathrm{score}(X, Y)$ in decreasing order of likelihood [7]. This list is produced upon different findings in the graph, one could be the length of the shortest path [7]. This method can be used to suggest other individuals, whom you might also like or know. This is represented by a high probability in the $\mathrm{score}(X, Y)$ list, where X represents you and Y represents the person you might know. Another example is, if there is a high probability that you will form some kind of connection - a "like" for example - with a certain brand, you could be targeted with advertisements in advance.

V. Conclusion

The techniques and concepts in this paper are all rather basic in the field of data mining and social network analysis. Nonetheless, these are the fundamentals and a lot of new research is built upon these methods. As new OSN will appear and with the growth in both size and influence of the existing ones, this topic will be more relevant than ever in the past. There are networks with over hundreds of millions of active users and the research, like the companies owning the networks, are interested in the data that can be derived from this massive gathering of people, for what purpose whatsoever. Like Nettleton commented employees and students of big companies and universities will have the most favorable position for this field of research, simply because it takes a lot of computational cost to store and analyze these huge amounts of data. It will be fascinating to see how this topic will grow and influence our society as the networks will grow continuously. Companies like *Facebook* do not exactly know what they themselves can do with this huge amount of data [1]. The challenges these companies are faced with are basically the same as humanity has to accomplish, because they study human psychology and behavior [1]. Even today companies that run these large networks know more about us than we know and might want to [1]. And maybe in the future they are able to predict what we will do even before we ourselves know.

References

[1] T. Simonite, "What facebook knows: The company's social scientists are hunting for insights about human behavior. what they find could give facebook new ways to cash in on our data—and remake our view of society." 2012, (Last accessed Sun, 29 May 2016 09:15:00). [Online]. Available: https://www.technologyreview.com/s/428150/what-facebook-knows/

[2] J. Akaichi, Z. Dhouioui, and M. Lopez-Huertas Perez, "Text mining facebook status updates for sentiment classification," 2013, pp. 640–645.

[3] D. F. Nettleton, "Data mining of social networks represented as graphs," *Computer Science Review*, vol. 7, pp. 1–34, 2013.

[4] M. Atzmueller, “Data mining on social interaction networks,” *JDMDH*, vol. 2014, 2014. [Online]. Available: http://jdmdh.episciences.org/11

[5] D. Boyd and N. B. Ellison, “Social network sites: Definition, history, and scholarship,” *J. Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007. [Online]. Available: http://dx.doi.org/10.1111/j.1083-6101.2007.00393.x

[6] J. Zhang, W. Hsu, and M. Lee, “Image mining: Issues, frameworks and techniques,” in *Proceedings of the Second International Workshop on Multimedia Data Mining, MDM/KDD’2001, August 26th, 2001, San Francisco, CA, USA*, 2001, pp. 13–20.

[7] J. Han and M. Kamber, *Data mining: Concepts and techniques*, 2nd ed., ser. The Morgan Kaufmann series in data management systems. Amsterdam: Elsevier/Morgan Kaufmann, 2010, chapter 9.

[8] D. Chakrabarti and C. Faloutsos, “Graph mining: Laws, generators, and algorithms,” *ACM Comput. Surv.*, vol. 38, no. 1, 2006. [Online]. Available: http://doi.acm.org/10.1145/1132952.1132954

[9] D. B. West, *Introduction to graph theory*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2001.

[10] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, “Defining and identifying communities in networks,” *Proceedings of the National Academy of Sciences*, vol. 101, no. 9, pp. 2658–2663, 2004.