Profiling Social Network Users with Machine Learning

Evis Trandafili
Polytechnic University of Tirana
Faculty of Information Technology
Sheshi Nene Tereza,
Tirana, Albania
etrandafili@fti.edu.al

Marenglen Biba
University of New York in Tirana
Faculty of Sciences
Rr. Komuna e Parisit,
Tirana, Albania
marenglenbiba@unyt.edu.al

Aleksandër Xhuvani
Polytechnic University of Tirana
Faculty of Information Technology
Sheshi Nene Tereza,
Tirana, Albania
axhuvani@fti.edu.al

ABSTRACT
Social Networks are becoming increasingly popular in our daily communication and rapid developments in data gathering technology have led to large amounts of data that are available from users' interactions. On the other side, the complexity of analyzing social interactions is not related only to the size of the network to be analyzed, but also to the nature of the interactions. One of the most important tasks in analyzing social networks is the user profiling (or clustering) which makes possible to design customized marketing strategies based on the type of the user. A user falling under a certain profile could then target with the same products used for other users in the same group. In this context, it is important to develop approaches that are able to efficiently and effectively profile users based on their interactions with other users. Machine learning methods have shown the capability to automatically discover patterns from data even in scenarios where complex relationships holds. In this paper, we show through experiments how machine learning algorithms can be effectively used to produce accurate profiling of real-world social network users. We show that users can be clustered in groups and that interesting patterns can be discovered among users not directly linked with decision tree learning.

Keywords
social networks, data mining, machine learning

1. INTRODUCTION
Social Network Mining (SNM) is a rapidly growing field that is increasingly receiving much attention in both the communities of data mining and in marketing and business strategies [1]. The main goal of this scientific area is the study of relationships between individuals regarding their social position, the analysis of their roles, the discovery of social structures and many other issues related to social behavior. The relationships between individuals have been cast as links in huge networks and these have been traditionally constructed based on interviews and responses given by social actors. However this has always led to limited scalability of the analysis due to the lack of an infrastructure where interaction logs could be produced and saved. After this an automated approach could then be used for data collection from the interactions between individuals.

Recently, SNM has intensively evolved into an outstanding area not only in social sciences but also in computer science, due to the success of online social networking and media-sharing sites, and moreover, due to the availability of large repositories of social network data. With the rapid development of Internet and Web 2.0, SNM has gained even more importance which is mainly due to the combination between social media sites and social networks. The more these two combine and merge with each other, the more individuals have additional alternative ways to connect and build online relationships among them. Online social networks have boosted the capability to collect data as shown by the growing number of individuals interacting in large scale online social network platforms such as Facebook, LinkedIn, Flickr, Instant Messenger, etc. With the amount of data coming from these platforms, SNM is even more powerful because large scale networks of social entities can yield patterns that are normally not observed in small networks. With millions or even more actors in a network, it is now possible to discover patterns that can have a valuable business usage.

Machine Learning and Data Mining have long dealt with the problem of inferring models for classification in many application domains [16, 17]. With the fast growing amount of available data, however, the capability of traditional approaches to learn useful models has reached the limit. Large environments are continuously posing new challenges to learning algorithms which have now to take into consideration the presence of many entities distributed in large systems. The possibility to involve in the learning process huge collections of documents and large databases, has led to new opportunities for discovering important relationships among apparently distant entities, but at the same time, has raised performance issues that the current machine learning methods have to deal with.

On the other side, the advantage of large amounts of data that can help to perform a thorough analysis comes together with a cost, which is the incapability of classical traditional machine learning and data mining methods to deal with this new scenario. This has given rise to challenges that are not only related to computational complexity, but also to the core methodological approaches of the learning and mining algorithms. These have to be redesigned under the new perspective of the online social networking data. With the algorithmic solutions developed in the machine learning and data mining areas, it is strongly related a set applications in online social networks. For this reason, it is essential for practitioners in the field of SNM, to understand the computational challenges that lie behind the analysis of social networks.

One of the applications in online social networks is the precise targeting of users based on their relationships with other

users. This process can be cast as a profiling process where users are partitioned into profiles. Each of these profiles identifies a certain category of users sharing common features. From a marketing point of view, this outcome can be used to design campaigns that can benefit from knowing which users fall in the same group. Based on this, products chosen by a user may probably be chosen by users belonging to the same group. In machine learning this task can be seen as clustering of instances where every instance is a user. In this paper, we show how clustering can be useful in large online social networks, by experimenting clustering algorithms such as the Expectation-Maximization algorithm on real world data extracted from the hi5 social network. We show that users apparently not related to each other, fall under the same group based on links with other common users.

In addition to clustering, marketing in social networks can benefit from predicting which user may be indirectly connected with another user from the point of view of social relationships even though there is no direct friendship among these users. In this case, we may apply machine learning algorithms that extract rules that express relationships among users. Since decision trees have been a successful machine learning algorithm applied to many tasks, we decided to apply this algorithm to data coming from the real world social network Hi5 to see whether interesting rules can be discovered regarding the relationship among users. We found that through this strategy, interesting patterns can be discovered that can later be used for marketing purposes.

The paper is organized as follows: Section 2 discusses social network mining and related work with interesting applications of machine learning and data mining algorithms to the task of analyzing these networks; Section 3 discusses clustering as a machine learning technique, and in particular it introduces the EM algorithm which is the one used in the experiments; Section 4 presents a brief introduction to decision tree learning; Section 5 presents the experimental setting and the results; Section 6 presents the experimental evaluation and we conclude in Section 7.

2. SOCIAL NETWORK MINING

Recently, a growing amount of effort in the data mining community has been dedicated to analyzing social networks. We review here some main approaches that have been developed recently.

One of the main problems in social network mining is community structure discovery. Targeting customers in an appropriate and customized approach has a long history in economics, statistics and marketing. In a business intelligence effort towards solving this problem, an important problem has been grouping or clustering of customers. In the context of social networks, actors in a social network form groups and the task of community structure discovery is to identify these communities through the study of network structures and topology. In other words, the challenge is to find groups of users for which, the set of edges is dense within the group and sparse outside the group. Social networks are usually characterized by a strong community effect. This means that in a group of people, these tend to interact with each other more than with people outside the group. A quantitative measure of the community effect is transitivity, that simplified, takes the form that friends of a friend are very likely to be also friends. An interesting coefficient is the proposed approach to measure the transitivity as the probability of connections between one vertex's neighboring friends [3].

For real social networks, computing the global clustering coefficient can become computationally intractable if we rely on exact counting. The exact counting of triangles has been shown to be computationally very expensive [4, 5]. Other approaches base the counting on approximations such as the work presented in [6]. Recently in [7] the authors presented a label propagation approach to community structure discovery. They introduce a semi–synchronous version of label propagation algorithms which aims to combine the advantages of both synchronous and asynchronous models. The authors prove that the proposed models always converge to a stable labeling. Moreover, the authors experimentally investigate the effectiveness of the proposed strategy comparing its performance with the asynchronous model both in terms of quality, efficiency and stability. Tests show that the proposed protocol does not harm the quality of the partitioning. Moreover it is quite efficient; each propagation step is extremely parallelizable and it is more stable than the asynchronous model, thanks to the fact that only a small amount of randomization is used by the proposed approach.

Another interesting problem is social network evolution which is an interesting and challenging task in machine learning. This task is mainly concerned with modeling and often discovering the dynamics of the social graph. An interesting work in this direction is the one presented in [8] where the authors present a detailed study of network evolution by analyzing four large online social networks. They exploit the full temporal information about node and edge arrivals. This study performed for the first time at a large scale, involves the analysis of individual node arrival and edge creation processes that jointly lead to macroscopic properties of networks. The authors, using a methodology based on maximum-likelihood, perform a thorough investigation of a wide variety of network formation strategies, and showed that edge locality plays a critical role in evolution of networks. The discovered patterns supplement earlier network models based on the inherently non-local preferential attachment. In addition, based on their observations, the authors develop a complete model of network evolution, where nodes arrive at a prespecified rate and select their lifetimes. The authors also show analytically that the combination of the gap distribution with the node lifetime leads to a power law out-degree distribution that accurately reflects the true network in all four cases.

Social interactions that occur regularly will typically correspond to significant yet often infrequent and hard to detect interaction patterns. To identify such regular behavior, the authors in [9] propose a new mining problem of finding periodic or near periodic subgraphs in dynamic social networks where scalability is also a major issue. They propose a practical, efficient and scalable algorithm to find such subgraphs that takes imperfect periodicity into account and demonstrate the applicability of their approach on several real-world networks and extract meaningful and interesting periodic interaction patterns.

Social networks often involve multiple relations simultaneously. People usually construct an explicit social network by adding each other as friends, but they can also build implicit social networks through daily actions like commenting on posts, or tagging photos. In [10] it is addressed this problem: given a real social networking system which changes over time, do daily interactions follow any pattern? The authors model the formation and co-evolution of multi-modal networks proposing an

approach that discovers temporal patterns in social interactions. They show the effectiveness of the approach on two real datasets (Nokia FriendView and Flickr) with 100,000 and 50,000,000 records respectively, each of which corresponds to a different social service, and spans up to two years of activity.

A very challenging task in dynamic social networks is link prediction. Interesting recent research has been dedicated to the link prediction task which is complex due to the inherent skewness of network data. Link prediction methods can be categorized as either local or global. Local methods consider the link structure in the immediate neighborhood of a node pair, whereas global methods utilize information from the whole network. An interesting approach is community (cluster) level link prediction method without the need to explicitly identify the communities in a network. This approach is presented in [11] where the authors define a variable-cost loss function to address the data skewness problem. They provide theoretical proof that shows the equivalence between maximizing the well-known modularity measure used in community detection and minimizing a special case of the proposed loss function. They design a boosting algorithm to minimize the loss function and present an approach to scale-up the algorithm by decomposing the network into smaller partitions and aggregating the weak learners constructed from each partition. The authors empirically evaluate the proposed algorithm by evaluating it on 4 real-world network datasets.

In a recent approach [12], the authors proposed a new method for characterizing the dynamics of complex networks with an application to the link prediction problem. The approach proposed is based on the discovery of network sub graphs (triads of nodes) and measuring their transitions during network evolution. The authors define the Triad Transition Matrix (TTM) containing the probabilities of transitions between triads found in the network, then they show how it can help to discover and quantify the dynamic patterns of network evolution. They also propose the application of TTM to link prediction with an algorithm (called TTM-predictor) which shows good performance, especially for sparse networks analyzed in short time scales.

Modeling event propagation is another important challenge in social networks. Handling this task appropriately leads to interesting applications for viral marketing. In [13], the authors propose a scalable framework for modeling competitive diffusion in social networks. In social networks, multiple phenomena often diffuse in competition with one another. Applications of this kind include, for instance, eventual results from multiple competing diffusion models (e.g. what is the likely number of sales of a given product). The authors in [13] define the most probable interpretation (MPI) problem which technically formalizes this need. They develop algorithms to efficiently solve MPI and show experimentally that their algorithms work on graphs with millions of vertices.

A very challenging task in online social networks is the discovery of the diffusion paths and the evolutionary process of a topic. Unlike explicit user behavior (e.g., buying a book) both these are implicit. An interesting approach has been recently proposed in [14] where the authors track the evolution of an arbitrary topic and reveal the latent diffusion paths of that topic in a social community. The proposed approach is based on a novel and principled probabilistic model which casts the task as a joint inference problem that considers textual documents, social influences, and topic evolution in a unified way. A Gaussian Markov Random Field is introduced to model the whole diffusion process. Experiments on both synthetic data and real world data show that the discovery of topic diffusion and evolution benefits from this joint inference, and the probabilistic model proposed performs significantly better than existing methods.

Another approach is presented in [15], where the authors develop techniques for identifying and modeling the interactions between social influence and selection, using data from online communities where both social interaction and changes in behavior over time can be measured. They find clear feedback effects between the two factors and there is a rising similarity between two individuals serving, in aggregate, as an indicator of future interaction. The results show that similarity continues to increase steadily, although at a slower rate, for long periods after initial interactions. The authors also consider the relative value of similarity and social influence in modeling future behavior. For instance, to predict the activities that an individual is likely to perform in the future, it is interesting to know whether it is more useful to know the current activities of their friends, or of the people most similar to them.

On large-scale networks, there is a need to perform aggregation operations. Unfortunately the existing implementation of aggregation operations on relational databases does not guarantee superior performance in network space, especially when it involves edge traversals and joins of gigantic tables. In [22], the authors investigate the neighborhood aggregation queries: Find nodes that have top-k highest aggregate values over their h-hop neighbors. While these basic queries are common in a wide range of search and recommendation tasks, surprisingly they have not been dedicated much attention. The work in [22] proposes a Local Neighborhood Aggregation framework, to answer these queries efficiently. The approach exploits two properties unique in network space: First, the aggregate value for the neighboring nodes should be similar in most cases;
Second, given the distribution of attribute values, it is possible to estimate the upper-bound value of aggregates. These two properties inspire the development of novel pruning techniques, forward pruning using differential index and backward pruning using partial distribution. Empirical results show that the proposed approach could outperform the baseline algorithm up to 10 times in real-life large networks.

In this paper, we exploit the potential of machine learning algorithms to perform clustering and that of decision trees models which can capture interesting relationships among social network users.

3. CLUSTERING

3.1 Clustering in Machine Learning

Clustering is an old problem in computer science and statistics in general, and in the field of data mining and machine learning it is an approach that has been intensively developed. Clustering techniques apply when there is no class to be predicted but rather when the instances are to be divided into natural groups [16, 17]. The generated clusters reflect a certain degree of relationship or similarity among the instances falling in the same cluster, while at the same time these groups bear some distinguishing features that give the possibility to discriminate between different clusters.

As mentioned in [16] there are different alternatives in expressing the result of clustering. The groups that are identified may be exclusive so that any instance belongs to only one group. Or they may be overlapping so that an instance may fall into

several groups. In addition, the clusters may also be probabilistic, whereby an instance belongs to each group with a certain probability. Or they may be hierarchical, such that there is a crude division of instances into groups at the top level, and each of these groups is refined further all the way down to individual instances.

3.2 Iterative distance-based clustering

The most well-known clustering approach is called *k-means*. This technique works by first you specifying in advance the number of clusters that are going to be generated: this is called the parameter *k*. At this point, the k points are chosen at random as cluster centers and for all the instances it is calculated the ordinary Euclidean distance metric to the closest cluster center. After this step, we compute the centroid or mean of the instances in each cluster, which is called "means". These centroids are taken to be new center values for their respective clusters. All these steps are repeated with the new cluster centers until the same points are assigned to each cluster in consecutive rounds, at which stage the cluster centers are stabilized and will continue to remain the same [16].

3.3 Probabilistic clustering

A principled approach to the clustering problem comes from statistics. From a probabilistic point of view, the goal of clustering is to find the most likely set of clusters given the data and any prior expectations. Since in many cases the evidence is not enough to place categorically the instances in one cluster or the other, it is often convenient to have a certain probability of an instance to belong to each cluster. This helps to eliminate the non-flexibility that is often associated with methods that make hard judgments.

The basis for statistical clustering is a statistical model called *finite mixtures.* A *mixture* is a set of *k* probability distributions, representing *k* clusters, that govern the attribute values for members of that cluster. Each of these distributions gives the probability that a particular instance would have a certain
set of attribute values if it were *known* to be a member of that cluster. Any particular instance belongs to one and only one of the clusters, but it is not known which one. In addition, the clusters are not equally likely: there is some probability distribution that reflects their populations. The simplest finite mixture situation occurs when there is only one numeric attribute, which has a Gaussian or normal distribution for each cluster but with different means and variances. The clustering problem is to take a set of instances and a prespecified number of clusters, and work out each cluster's mean and variance and the population distribution between the clusters [16].

3.4 The Expectation-Maximization Algorithm

The problem in probabilistic clustering is that it is not known the distribution that each training instance comes from and the parameters of the mixture model. One way to proceed is that of adopting the procedure used for the *k*-means clustering algorithm and iterate. Initially start with guesses for the parameters, use them to calculate the cluster probabilities for each instance, use these probabilities to reestimate the parameters, and repeat.
This is called the *EM algorithm,* for *expectation–maximization* [18]. The first step, calculation of the cluster probabilities (which are the "expected" class values) is "expectation"; the second, calculation of the distribution parameters, is "maximization" of the likelihood of the distributions given the data.
In this paper we use the EM-algorithm in order to generate clusters of social network users.

4. DECISION TREES

One of the most successful models in machine learning are decision trees. Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree. Learned trees can also be re-represented as sets of if-then rules to improve human readability [17]. In this model, it is followed a "divide-and-conquer" approach to the problem of learning from a set of independent instances. Nodes in a decision tree involve testing an attribute of the instances. Usually, the test at a node compares an attribute value with a constant. However, some trees compare two attributes with each other, or use some function of one or more attributes. Leaf nodes provide a classification that applies to all instances that reach the leaf. To classify an unknown instance, it is routed down the tree according to the values of the attributes tested in successive nodes, and when a leaf is reached the instance is classified according to the class assigned to the leaf [16].
A decision tree is presented in Fig. 1, where the tree has been generated from contact lenses data in order to help in prescribing the correct contact lense for a patient. As we can see, the classification is performed starting from the top at the root of the tree and testing an attribute at every node in the tree.

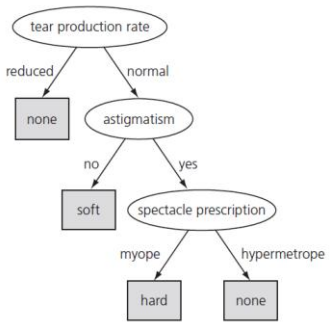

Figure 1. Decision tree for contact lense [16].

As mentioned in [17], appropriate problems for decision tree learning are those that present features such as: Instances are represented by attribute-value pairs; the target function has discrete output values; disjunctive descriptions may be required; the training data may contain errors; and the training data may contain missing attribute values.

Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. This approach is exemplified by the ID3 algorithm [18] and its successor C4.5 [19]. ID3 learns decision trees by constructing them top-down, beginning at the root of the tree and deciding which attribute should be tested. To perform this decision, each instance attribute is evaluated using a statistical test to determine how well it alone classifies the training examples. The best attribute is selected and used as the test at the root node of the tree. The test is based on a measure called information gain which expresses the expected reduction in entropy caused by knowing

the value of an attribute. Put another way, the measure gives the information provided about the target function value, given the value of some other attribute.

In this paper we will use the decision trees in a different setting. We would like to discover rules that govern the relationships among users in the online social network. We are not interested in classifying the single user but in extracting decision trees from the data that can lead to interesting patterns among the users.

5. EXPERIMENTS

5.1 Experimental Setting

The experiments performed here, have the goal of answering these two questions:

> **Q1.** Are there any commonalities between the 20 friends of a certain friend?
>
> **Q2.** Is there any pattern or relationship among the 20 friends that can be discovered from the data?

To answer question **Q1,** we perform clustering on the users' data and check whether users not directly related to each other (but indirectly through other users) fall under the same cluster. For example, if user U_1 has as friends the users U_2 and U_3, but these two users are not connected together, is there any common feature between U_2 and U_3 that makes them fall in the same group?

To answer question **Q2,** we perform decision tree learning and check whether there are interesting patterns among the users who, based on the data are not related to each other but have in common other users.

5.2 Preprocessing and input engineering

The data used in our experiment have been extracted from the online social network Hi5[1] and first presented in [21]. This dataset is composed of 4928 users where for every user there are 20 other users considered as his closest links. In Figure 2 it is shown the table of users where every user is in a row and the friends are in columns.

```
%% 4928 users (rows) with maximum 20 friends (columns) for each user
93          194         195         196         100
23          259         383         384         385
70          17          571         572         573
56          757         758         759         43
941         942         943         944         945
```

Figure 2. Dataset of users from Hi5

In order for this table to be processed in Weka, we should translate the matrix in the .arff format which is compatible with the requirements of the Weka program input. In this context we need one single record for each user, whose attributes are all the remaining users (4928 in total) and each attribute has a value of "*yes"* in case there is a link between the two users and "no" in case there is not. For this reason we have developed a program in the Java language that translates the data into the required format of Weka.

[1] http://www.hi5.com

```
@relation hi5
@attribute user1 {yes, no}
@attribute user2 {yes, no}
@attribute user3 {yes, no}
@attribute user4 {yes, no}
@attribute user5 {yes, no}
@attribute user6 {yes, no}
@attribute user7 {yes, no}
@attribute user8 {yes, no}
@attribute user9 {yes, no}
@attribute user10 {yes, no}
@attribute user11 {yes, no}
@attribute user12 {yes, no}
@attribute user13 {yes, no}
```

Figure 3. Part of the attributes section on the .arff file

The content of the .arff file's data regarding the record of one single user is shown in Figure 4. Each value matches the attributes listed in Figure 3. The overall content of the .arff file contains the data regarding 4928 users.

```
@data
yes,yes,yes,yes,yes,yes,yes,yes,yes,yes,yes,yes,yes,yes,no,n
,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no
no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,no,
```

Figure 4. User record in the .arff format

5.3 Models generated

The clustering algorithm used is expectation-maximization (EM) clustering as implemented in Weka [16, 18]. We performed two experiments: one where the algorithm is left to find the number of clusters automatically and one other where the number of clusters is given as input parameter to the algorithm. In the second experiment we used as number of clusters k=10, while in the first case the algorithm was run with default parameters and found the optimal k=8. Figure 5 shows the clusters generated in the second experiment. Due to the memory limitations we took 1000 users (rows) and for each of these users we took links with 500 other users. In addition we also used values {1, 0} instead of {yes, no}.

```
Number of clusters selected by cross validation: 8


               Cluster
Attribute          0      1      2      3      4      5      6      7
               (0.02) (0.07)    (0)    (0)  (0.9)    (0)    (0)    (0)
=================================================================================
user1
  mean             0 0.0135      0      0 0.0044      1      0      0
  std. dev.   0.0773 0.1155 0.0773 0.0773 0.0666 0.0773 0.0773 0.0773

user2
  mean         0.111      0      0      0 0.0011      0      0      0
  std. dev.   0.3141 0.0547 0.0547 0.0547 0.0333 0.0547 0.0547 0.0547

user3
  mean             0      0      0      0 0.0033      0      0      0
  std. dev.   0.0547 0.0547 0.0547 0.0547 0.0577 0.0547 0.0547 0.0547

user4
  mean             0      0      0      0 0.0056      0      0      0
  std. dev.   0.0706 0.0706 0.0706 0.0706 0.0744 0.0706 0.0706 0.0706
```

Figure 5. The clusters found automatically, with k selected with cross-validation

Regarding the decision tree learning, we followed this approach: we selected for each case the class attribute to be one user, and use all the other users as normal attributes. Figure 6 shows the decision tree learned for user 16 and Figure 7 shows the decision tree learned for user 6.

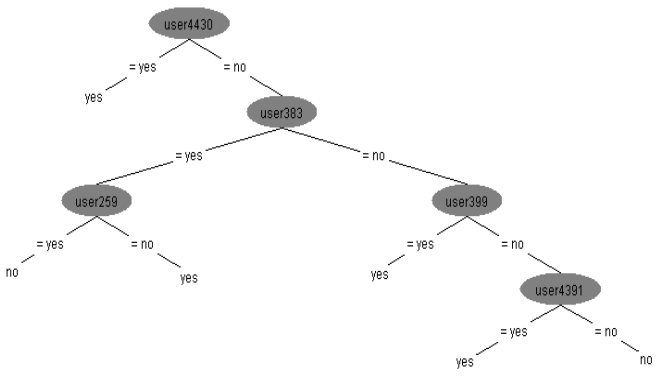

Figure 6. Decision tree generated for user 16

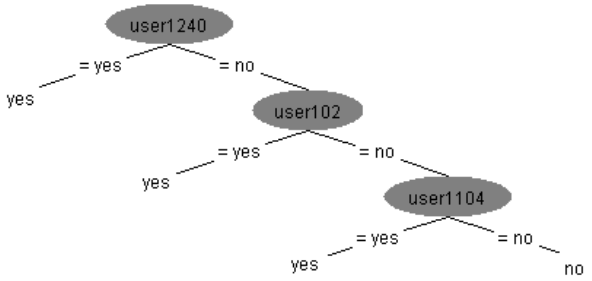

Figure 7. Decision tree generated for user 6

6. EVALUATION OF RESULTS

The generated clusters give insight into interesting grouping of apparently unrelated users. We discovered that many users apparently not related to each other through direct common shared friends, fall under the same cluster. This result shows that it is not sufficient to consider only the direct friends of users in order to group them, but we need to consider the whole dataset. In this case a user record is composed of 4928 attributes each expressing the presence or not among the twenty friends of a user. The investigation of the clusters can be interesting under a marketing point of view where users that were never considered to be target of a certain campaign, may now be clustered together with other users who have been targeted normally. The potential of clustering is therefore important from a business point of view.

In the decision tree learning experiments, we discovered some interesting patterns. For example, as can be seen in Figure 6, the model generated for user 6 leads to some useful patters regarding other users. In this case, users 4430, 383, 399 and 4391 are never present altogether in the dataset jointly with user 6. This means that among 4928 users there is no single user where among the twenty closest links we can find the above users. On the other side, we can see that whenever user 6 is present as a link, 4430 is present also. However, among the twenty links of user 6 we do not find the user 4430. The fact that user 6 is associated in the decision tree always with user 4430 is a strong evidence that these two might probably be recommended to each other for adding the link. The second case is that of user 383 whose presence implies the exclusion of user 4430. If user 383 is present, then the presence also of user 259, will also exclude user 6. In other words, users 4430, 383 and 259 should not be recommended to each other. In Figure 7 it is shown the decision tree learned for user 6. An interesting pattern here is that users 1240, 102 and 1104 in relation with user 6, will always appear in lists as excluding each other. In other words, these users should not be recommended to each other as potential links.

7. CONCLUSION

Social Networks are becoming increasingly popular and it is important to investigate whether well-established fields such as machine learning and data mining can help in automatically processing large amounts of data that are being gathered every day. In this paper, we investigated how machine learning algorithms can be effectively used to produce accurate profiling of real-world social network users. We showed through experiments on real-world data that users can be clustered in groups and that with decision tree learning, interesting patterns can be discovered among users not directly linked. As future work we intend to apply association rule mining that due to high memory and computational demand, requires a more powerful computing infrastructure than the currently available for the authors.

8. ACKNOWLEDGMENTS

Our thanks to the Weka development team that has made it possible with an open source software to work on machine learning algorithms. We also would like to thank the authors of [21] for sharing the dataset and providing explanations about it.

9. REFERENCES

[1] Trandafili, E. and Biba, M. 2013. A Review of Machine Learning and Data Mining Approaches for Business Applications in Social Networks. In International Journal of E-Business Research, January 2013, Vol. 9, No. 1, IGI-Global.

[2] Trandafili, E. and Biba, M. 2013. Scalable and High Performing Learning and Mining in Large-Scale Networked Environments: A State-of-the-art Survey. In N.T. Nguyen (Ed.): Transactions on CCI X, LNCS 7776, pp. 162–176, Springer, 2013.

[3] Tang, L. & Liu, H. 2010. Graph Mining Applications to Social Network Analysis. In *Managing and Mining Graph Data. The Kluwer International Series on Advances in Database Systems*, 2010, Volume 40, 487-513.

[4] Schank ,T., & Wagner, D. 2005. Finding, counting and listing all triangles in large graphs, an experimental study. In Sotiris E. Nikoletseas, ed., *Workshop on Experimental and Efficient Algorithms, Lecture Notes in Computer Science, 3503*, (pp. 606-609), Springer.

[5] Latapy, M. 2008. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theor. Comput. Sci.,* 407(1-3):458–473, 2008.

[6] Buriol, L. S., Frahling, G., Leonardi, S., Marchetti-Spaccamela, A., & Sohler, C. 2006. Counting triangles in data streams. *In PODS '06: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (pp. 253–262), ACM New York, NY, USA.

[7] Cordasco, G. & Gargano, L. 2012. Label propagation algorithm: a semi-synchronous approach. International Journal of Social Network Mining, Vol. 1, No. 1.

[8] Leskovec, J., Backstrom, L., Kumar, R., & Tomkins, A. 2008. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD International Conference textiton Knowledge Discovery and Data Mining*. ACM, New York, (pp. 462–470).

[9] Lahiri, M., & Berger-Wolf, T.Y. 2008. Mining Periodic Behavior in Dynamic Social Networks. In *Proceedings. of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, December 15-19, 2008, Pisa, Italy. IEEE Computer Society.

[10] Du, N., Wang, H., and Faloutsos. Ch. 2010. Analysis of Large Multi-modal Social Networks: Patterns and a Generator. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010*, Proceedings, Part I. LNCS 6321, Springer.

[11] Comar, P. M., Tan, P. A. & Jain, K. 2011. LinkBoost: A Novel Cost-Sensitive Boosting Framework for Community-Level Network Link Prediction. In D. J. Cook, J. Pei, W. Wang, O. R. Zaane, X. Wu (Eds.): *11th IEEE International Conference on Data Mining, ICDM 2011*, Vancouver, BC, Canada, December 11-14, 2011, IEEE Computer Society.

[12] Juszczyszyn, K., Musial, K., & Budka, M. 2011. Link Prediction Based on Subgraph Evolution in Dynamic Social Networks. Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom).

[13] Broecheler, M., Shakarian, P. & Subrahmanian, V.S. 2010. A Scalable Framework for Modeling Competitive Diffusion in Social Networks. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom / IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2010*, (pp. 295-302), IEEE Computer Society.

[14] Xide Lin, C., Mei, Q., Han, J., Jiang, Y., and Danilevsky, M. 2011. The Joint Inference of Topic Diffusion and Evolution in Social Communities. In D. J. Cook, J. Pei, W. Wang, O. R. Zaane, X. Wu (Eds.): *11th IEEE International Conference on Data Mining, ICDM 2011*, Vancouver, BC, Canada, December 11-14, 2011, IEEE Computer Society.

[15] Crandall, D. J., Cosley, D., Huttenlocher, D. P., Kleinberg, J. M., & Suri, S. 2008. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*.

[16] Witten, I.H., Frank, E. and Hall, M.E., 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. (third edition), Morgan Kaufmann.

[17] Mitchell. T. 1997. *Machine Learning*, McGraw Hill, 1997.

[18] Dempster, A.P.; Laird, N.M.; Rubin, D.B. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society. Series B (Methodological) 39 (1): 1–38.

[19] Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1(1):81–106.

[20] Quinlan, J. R. 1993. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.

[21] Symeonidis P., Tiakas E., Manolopoulos Y.: Transitive Node similarity for Link Prediction in Social Networks with Positive and Negative Links, Proceedings of the 4th International Conference on Recommender Systems (RecSys'2010), pp. 183-190, Barcelon, Spain, 2010.

[22] Yan, X., He, B., Zhu, F., Han, J. 2010. Top-K Aggregation Queries Over Large Networks.In: IEEE 26th International Conference on Data Engineering (ICDE), pp. 377–380, 2010.