Finding unexplained human behaviors in Social Networks

F.Persia, F.Amato, F.Gargiulo, S.R.Poccia, and A.De Santo

Dip. di Ingegneria Elettrica e Tecnologie dell'Informazione,
University of Naples Federico II
via Claudio 21, 80125, Naples, Italy
{fabio.persia@unina.it,flora.amato@unina.it,f.gargiulo@cira.it,silvestro.poccia@gmail.com,aniello.desanto@gmail.com}

Discussion Paper

Abstract. Detection of human behavior in On-line Social Networks (OSNs) has become a very important challenge for a wide range of applications, such as security, marketing, parent controls and so on, opening a wide range of novel research areas, which have not been fully addressed yet.
In this paper, we present a two-stage method for finding unexplained (and potentially anomalous) behaviors in social networks. First, we use Markov chains to automatically learn from the social network graph a number of models of human behaviors (*normal* behaviors); the second stage applies an activity detection framework based on the concept of *possible words* to detect all unexplained activities with respect to the well-known behaviors. Some preliminary experiments using Facebook data show the approach efficiency and effectiveness.

Keywords: Social Network Analysis, Anomaly Detection, Data in social networks

1 Introduction

This paper refers to [1], which has been just published in the IEEE Eighth International Conference on Semantic Computing.

Online Social Networks (OSNs) have become extremely popular in recent years, leading to the presence of huge volumes of users' personal information on the Internet.

The ever increasing number of social networks' users on one hand and the massive amount of information being shared daily on the other hand, has encouraged attackers to develop and use different techniques to collect and analyze such information for a number of malicious purposes, including spear-fishing attacks and identity theft.

In this paper, we focus our attention on the problem of anomaly detection on OSNs. For such a reason, we are specifically interested in defining a framework for detecting users' *anomalous behavior* within OSNs, that would allow an application to identify behavior patterns still unknown to social networks experts,

and to add them to an increasing body of knowledge. For example, monitoring OSN interactions in a primary school network could help identify anomalous users, such as bullies and sexual harassers.

We consider users' behaviors from a "network traffic perspective": observations from traffic analysis (server or client side *logs*) provide information about users' activities, that a *social graph* alone is not able to capture, thus significantly contributing to a better understanding semantics [7, 11] behind the interactions between users and OSN [3, 4, 8].

In [5], Kammenhuber *et al.* present an interesting methodology for extracting client side logs from the traffic generated by a large users' group on Internet on the base of the related full *clickstream* (i.e. the result pages users viewed and the subsequently visited hyperlinked pages). They also propose a Markov model that captures users' browsing behavior and allows to infer prevalent search patterns.

In turn, Wang *et al.* [6] propose clickstream models as a new tool to detect fake identities in OSNs. Similarly to the described approaches, we exploit the users' clickstream data to derive information about user behavior. But differently from them, we define a more complex probabilistic model based on the concept of *possible worlds* to model normal and *unexplained* behaviors, exploiting apposite reasoning techniques to detect anomalous activities [2]. From this viewpoint, our work is more similar to several approaches based on several formalism to model *normal behaviors* and on different algorithms to detect anomalies that have been proposed not for OSNs, but for video surveillance domain [9, 10].

Thus, we present a two-stage method for anomaly detection in humans' behavior while they are using a social network. The first stage uses *Markov Chains* to automatically learn from a social network graph all models of known human behaviors that can be considered *normal*; the second stage applies the described activity detection framework based on the concept of *possible worlds* to detect all *unexplained* behaviors - subsequences of the time-stamped social data that known models are not able to explain with a certain confidence.

2 Modeling Unexplained User Behaviors in OSNs

Our model has been expired by the works [2, 10] in order to derive a formal definition of *User Behavior* for OSNs. In particular, we first discuss possible ways to identify significant user actions in OSNs and show how models of known behaviors can be built from a set of training social data. Then, we specify the probability for a sequence of data to be unexplained with respect to a set of known user behaviors and finally discuss some detection algorithms.

2.1 Basic Definitions

Definition 1 (User Behavior) *A User Behavior is a labeled directed graph* $B = (V, E, \delta, \rho)$ *where: (i)* V *is a finite set of nodes labeled with action symbols from* S*; (ii)* $E \subseteq V \times V$ *is a set of edges; (iii)* $\delta : E \rightarrow \mathbb{N}^+$ *associates with each edge* $\langle v_i, v_j \rangle$ *an upper bound of time that can elapse between* v_i *and* v_j*; (iv)*

$\rho : E \rightarrow (0,1)$ *is a function that associates a probability distribution with the outgoing edges of each node, i.e.* $\forall v \in V \sum_{\langle v,v' \rangle \in E} \delta(\langle v,v' \rangle) = 1$*; (v) there exists an initial node* I *in the behavior definition, i.e.* $\{v \in V \mid \nexists\ v' \in V \ s.t.\ \langle v', v \rangle \in E\} \neq \emptyset$ *; (vi) there exists a final node* F *in the behavior definition, i.e.* $\{v \in V \mid \nexists\ v' \in V \ s.t.\ \langle v, v' \rangle \in E\} \neq \emptyset$.

As shown in Definition 1, we assume the existence of a finite set S of *action symbols* representing particular interactions (e.g. Share Photo, Write/Read Message, Check Notification) between users and the OSN. Figure 1 shows a user behavior model representing simple interactions between a user and a common OSN like Facebook.

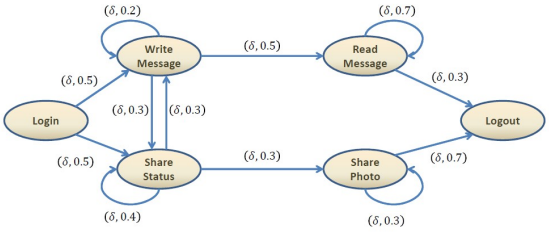


Fig. 1: An example of User Behavior Model.

Then, we define an *instance* of a user behavior as a specific path in B from the initial node to the end node.

Definition 2 (User Behavior Instance) *An instance of a User Behavior (V, E,* δ*,* ρ*) is a finite sequence* $\langle v_1, ..., v_m \rangle$ *of nodes in V such that: (i)* $\langle v_i, v_{i+1} \rangle \in E$ *for* $1 < i < m$*; (ii)* $\{v \mid \langle v, v_1 \rangle \in E\} = 0$*, i.e.* v_1 *is the start node I; (iii)* $\{v \mid \langle v_m, v \rangle \in E\} = 0$*, i.e.* v_m *is the final node F. The probability of the instance is* $\prod_{i=1}^{m-1} \rho(\langle v_i, v_{i+1} \rangle)$.

We work with sequences of time-stamped events. Let us assume that the number of observable actions in our domain is finite, each action can then be associated to a different action symbol in the set S. We define a *log entry* as a pair $\lambda = (s, ts)$, where $\lambda.s$ is the action symbol associated to the event and $\lambda.ts$ is the time stamp at which s was observed.

We call a *Social Log* Λ a finite sequence of log entries λ_i.

Now, we are in the position of defining the concept of *Behavior Occurrence.*

Definition 3 (Behavior Occurence) *Let* Λ *be a* Social Log *and B=(V, E,* δ*,* ρ*) a User Behavior. An* occurrence *of B in* Λ *is a sequence* $\langle (\lambda_1, v_1) ... (\lambda_m, v_m) \rangle$ *where: (i)* $\langle \lambda_1, ..., \lambda_m \rangle$ *is a subsequence of* Λ *such as* $\lambda_i = (\lambda_i.ts, \lambda_i.s)$*,* $\lambda.s$ *being an action symbol from S and* $\lambda.ts$ *the associated time-stamp; (ii)* $\langle v_1, ..., v_m \rangle$ *is an*

instance of B; (iii) $v_i = \lambda_i.s$ *for* $1 < i < m$[1]*; (iv)* $\lambda_{i+1}.ts - \lambda_i.ts \leq \delta(\langle v_i, v_{i+1}\rangle)$ *for* $1 < i < m$.

The probability $p(o)$ of the occurrence o should be the probability of the instance $\langle v_1, ..., v_m\rangle$.

2.2 Extimating Typical Behaviors

One of the problems we face is how known behavior models can be acquired. Here, we propose to identify significant models of typical behaviors through first-order *Markov Chains* built from sets of training data.

Let us consider a Markov Chain X_1^n with m states, where m is finite. Hence the transition probability matrix is unknown, we impose no restriction on it and want to infer the p_{ij} entries of such matrix $\hat{P}$.

Given the sequence of training data, an estimate of $\hat{P}$ is obtained by computing:

$$\hat{p}_{ij} = \frac{f_{ij}}{\sum_{j=1}^{m} f_{ij}}$$

where $\hat{p}_{ij}$ is the estimate of the probability of the transition from state i to state j and f_{ij} denotes the number of transition from i to j. It can be shown that this is equivalent to the maximum likelihood estimate of the transition probability matrix.

2.3 Probabilistic Model and Unexplained Behaviors

We define a theory for searching *unexplained* behaviors with respect to a set of behavior models $\mathcal{B}$. In order to do that, we design a probabilistic model and define the *totally and partially unexplained behaviors* similarly to [2, 10].

2.4 Detection Algorithms

The unexplained behavior detection problem is opportunely faced through the Top-k TUB (Totally Unexplained Behaviors) and Top-k PUB (Partially Unexplained Behaviors) algorithms that are a proper adaptation of those presented in [2].

Given a social log, the output of the Top-k TUB algorithm is a set of k unexplained behaviors having maximum probability. In turn, the Top-k PUB algorithm computes a set of top-k partially unexplained behaviors in a social log. More details about the related implementation are in [2].

3 System Architecture

The theoretical model has been exploited to develop a framework for the detection of unexplained behaviors in OSN crawled data streams. The structure of the system is based on a modular architecture, as shown in Figure 2.

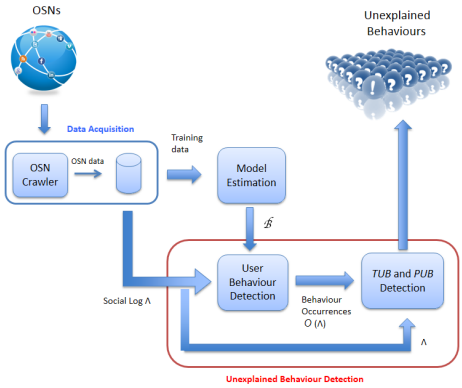


Fig. 2: System Architecture

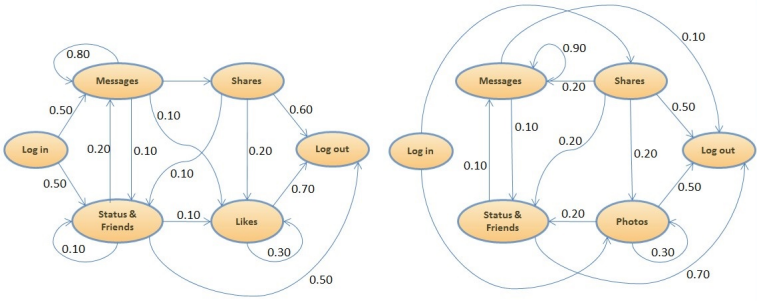


Fig. 3: Known Behavior Models

The *Data Acquisition* component includes a *Data Crawler* that captures information about user sessions from a specific OSN and saves the data in a format suitable to the detection framework (i.e. the Social Log).

The *Unexplained Behavior Detection* component - composed by the *User Behavior Detection* and *TUB and PUB Detection* modules - has in charge the identification of unexplained behavior patterns in time-stamped social data, using a set of known behavior models. The known models are produced by a *Model Estimation* module having as input the stream of training data.

In particular, the User Behavior Detection Module takes as inputs time-stamped user data collected in a social log and a set of behavior models to find the behavior occurrences matching the known models. Behavior occurrences in a data stream are efficiently detected using *tMagic* [12], which allows to solve the problem of finding occurrences of high-level activity model in an observed data stream.

[1] v_i refers both to the node v_i in B and the action symbol s_i labeling it

The *TUB* and *PUB* Detection Module takes as input the behavior occurrences previously found into the social log, and finally discovers the Unexplained User Behaviors.

The whole system has been implemented using JAVA technologies, while the crawling of data has been performed by realizing an *ad hoc* application directly integrated in the Facebook Social Network.

4 Preliminary Experimental Results

This Section shows a preliminary experimental evaluation of our framework. We first describe our methodology to collect data from Facebook. Then, we present the experimental protocol and evaluate Top-k TUB and Top-k PUB algorithms both in terms of execution time scalability and detection accuracy.

4.1 Dataset Preparation

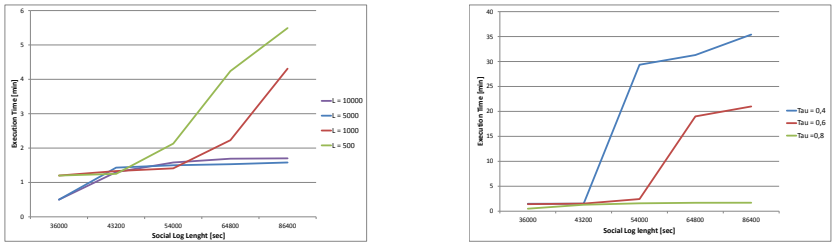
We decided to acquire data from Facebook users, using the built-in Facebook API developer environment. In particular, the environment has been used to develop an application able to collect users' data [2].

4.2 Experimental Protocol and Results

We use a sub-set of the collected data to feed to the Estimation Module. It produced several models (two of them are reported in Figure 3) of known behaviors to be used as input of the Unexplained Detection.

We tested our algorithms with logs covering from a minimum time span of 6 hours to a maximum of 24 hours.

[2] The data set is completely anonymous and freely available. For more information, please contact the corresponding authors



(a) Varying L (b) Varying τ

Fig. 4: Top-k TUB Running Times

Evaluating Execution Time We decided to measure[3] the execution time of Top-k TUB and Top-k PUB for different values of τ and L when varying the length of the input log and using the previously computed set of known behavior models.

Figures 4 and 5 show the processing time of TUB and PUB as a function of the input log length for different values of L and τ, by fixing $\tau = 0.8$ and $L = 10000$ respectively.

Finally, we evaluated accuracy both for the Top-k TUB and the Top-k PUB algorithms, as described in [2, 10].

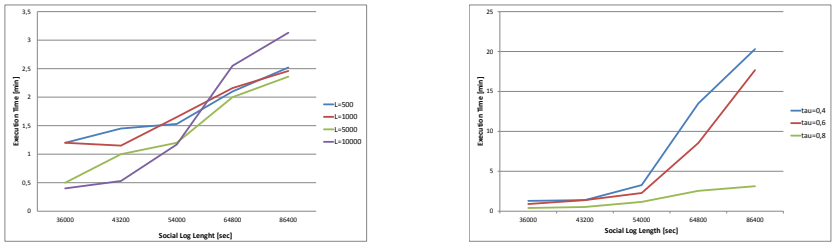
Accuracy Results Accuracy values for Top-k TUB and Top-k PUB when considering different models are shown in Figure 6 .

5 Conclusions and Future Work

This work provided formal definitions and a methodology to detect unexplained behavior in time-stamped OSN user data. We showed a prototype framework implementation and conducted experiments to provide an initial evaluation of our model detection performance that show quite good and encouraging results concerning accuracy and scalability.

Future work will be devoted to enlarge our experimentation, make a detailed study about matchings between unexplained and malicious behaviors and compare our approach with other different techniques.

[3] All experiments presented in this Section were conducted on a machine running Mac OS X 10.9.1, and mounting a *2GHz Intel Core i7* processor with a *8 GB, 1600 MHz DDR3*.



(a) Varying L (b) Varying τ

Fig. 5: Top-k PUB Running Times

τ	Accuracy
0.4	70, 75%
0.6	64, 33%
0.8	64, 33%

(a) Top-k TUB, when ignoring B_1

τ	Accuracy
0.4	87, 77%
0.6	78, 27%
0.8	72, 5%

(b) Top-k TUB, when ignoring B_2

τ	Accuracy
0.4	62, 3%
0.6	56, 42%
0.8	56, 42%

(c) Top-k PUB, when ignoring B_1

τ	Accuracy
0.4	70, 88%
0.6	66, 35%
0.8	65, 23%

(d) Top-k PUB, when ignoring B_2

Fig. 6: Accuracy

References

1. F. Amato, A. De Santo, V. Moscato, F. Persia and A. Picariello: "Detecting unexplained human behaviors in Social Networks", 2014 IEEE Eighth International Conference on Semantic Computing (ICSC), June 2014.
2. M. Albanese, C. Molinaro, F. Persia, A. Picariello, and V. Subrahmanian: "Discovering the top-k unexplained sequences in time-stamped observation data", IEEE Transactions on Knowledge and Data Engineering, vol. 26, pp. 577-594, March 2014.
3. F. Amato, A. Chianese, A. Mazzeo, V. Moscato, A. Picariello, F. Piccialli, (2013). The Talking Museum Project. Procedia Computer Science, 21, 114-121
4. L. Jin, Y. Chen, T. Wang, P. Hui, and A. Vasilakos: "Understanding user behavior in online social networks: a survey", Communications Magazine, IEEE, vol. 51, no. 9, pp. 144-150, 2013.
5. N. Kammenhuber, J. Luxenburger, A. Feldmann, and G. Weikum, "Web search clickstreams", in Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, IMC 2006, (New York, NY, USA), pp. 245-250, ACM, 2006.
6. G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao, "You are how you click: clickstream analysis for sybil detection", in Proceedings of the 22nd USENIX conference on Security, SEC 2013, (Berkeley, CA, USA), pp. 241-256, USENIX Association, 2013.
7. F. Amato, A. Mazzeo, V. Moscato, A. Picariello, "A system for semantic retrieval and long-term preservation of multimedia documents in the e-government domain"'. International Journal of Web and Grid Services, 5(4), pp. 323-338, 2009.
8. F. Amato, A. Mazzeo, V. Moscato, and A. Picariello, "Exploiting cloud technologies and context information for recommending touristic paths", in Intelligent Distributed Computing VII, pp. 281-287, Springer, 2014.
9. M. Albanese, V. Moscato, A. Picariello, V. S. Subrahmanian, and O. Udrea, "Detecting stochastically scheduled activities in video", IJCAI, pp. 1802-1807, 2007.
10. M. Albanese, C. Molinaro, F. Persia, A. Picariello, and V. S. Subrahmanian, "Finding unexplained activities in video", IJCAI,pp. 1628-1634, 2011.
11. F. Amato, A. Mazzeo, A. Penta, A. Picariello. "'Building RDF Ontologies from Semi-Structured Legal Documents"'. In Int. Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), pp. 997-1002, 2008.
12. M. Albanese, A. Pugliese, and V. Subrahmanian, "Fast Activity Detection: Indexing for Temporal Stochastic Automaton-Based Activity Models", IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 2, pp. 360-373, 2013.