Exploiting digital DNA for the analysis of similarities in Twitter behaviours

Stefano Cresci*, Roberto Di Pietro†§, Marinella Petrocchi*, Angelo Spognardi‡ and Maurizio Tesconi*

* Institute for Informatics and Telematics, National Research Council (IIT-CNR), Pisa, Italy [name.surname@iit.cnr.it]
† Nokia Bell Labs, Paris, France roberto.di_pietro@nokia-bell-labs.com
‡ Department of Computer Science, Sapienza Università di Roma, Rome, Italy spognardi@di.uniroma1.it
§ Department of Mathematics, University of Padua, Padua, Italy

Abstract—Recently, DNA-inspired online behavioral modeling and analysis techniques have been proposed and successfully applied to a broad range of tasks. In this paper, we employ a DNA-inspired technique to investigate the fundamental laws that drive the occurrence of similarities among Twitter users. The achieved results are multifold. First, we demonstrate that, despite apparently showing little to no similarities, the online behaviors of Twitter users are far from being uniformly random. Then, we perform a set of simulations to benchmark different behavioral models and to identify the models that better resemble human behaviors in Twitter. Finally, we demonstrate that the number and the extent of behavioral similarities within a group of Twitter users obey a log-normal distribution. Our results shed light on the fundamental properties that drive behaviors of groups of Twitter users, through the lenses of DNA-inspired behavioral modeling techniques. Our datasets are publicly available to the scientific community to further explore analytics of online behaviors.

I. INTRODUCTION

Online behavioral analytics has recently seen the emergence of a new stream of research that draws a parallel with bioinformatics, by applying techniques and methods typically involved in the analysis of biological DNA. Such studies offer fresh insights into the characteristics and fundamental properties of online behaviors and actions [1]. According to this new line of research, behaviors of online social network (OSN) users are modeled through strings of characters. Specifically, each action a user performs is associated with a character, similarly to the A, C, T, G bases of biological DNA. The chronological sequence of actions of a user leads to the creation of a string of characters, its digital DNA, which encodes the user’s behavioral timeline. Given a digital DNA string, well-known string mining and biological DNA analysis techniques [2], [3], [4] have been applied to study the characteristics of online user behaviors, for tasks such as anomalous behavior detection.

Recent applications of DNA-inspired techniques for spambot detection [5] highlighted that legitimate (human-operated) accounts share much lower behavioral similarities between one another, with respect to automated accounts. The latter, instead, commonly share longer behavioral sequences of actions. For instance, Figure 1 shows the difference between the amount of behavioral similarity (in terms of their actions) measured for legitimate and spambot accounts. As the number of accounts grows, the amount of similarity among legitimate accounts rapidly drops to very low levels, meaning that very few similarities are found among a large group of human-operated accounts. Conversely, the amount of behavioral similarity among automated accounts remains noticeably high, even when considering large groups of accounts. Further to the discovery of major differences between automated and legitimate accounts, these results raised questions regarding the amount of behavioral similarities of human-operated accounts themselves. In fact, the very limited similarity measured for legitimate accounts could suggest that human behaviors in OSNs are so diverse and heterogeneous that they could be considered completely unrelated between one another, with their actions almost uniformly random. Yet, to date, few results have been achieved to characterize diversity – or, conversely, similarity – of online human behaviors. A recent example is in [6], where the authors show that the interval patterns of online activities are representative of individuals.

Here, we investigate the fundamental characteristics of behavioral similarities among groups of users in OSNs. Our study is founded on recent results on behavioral modeling obtained via the adoption of digital DNA [5]. We focus on Twitter users and we measure the level of similarity between real behavioral sequences extracted from groups of Twitter accounts and synthetic sequences derived from the same groups, but randomly generated according to predefined behavioral models. Results of our analyses show that human behaviors are very heterogeneous, but not random at all.

1 More details on how these results have been obtained are in Section II.
Contributions. Our contributions are as follows:

- We propose a general algorithm to create null models of online behaviors;
- We benchmark 11 null models comprising ~92 millions of DNA strings, demonstrating that human behaviors are not comparable to sequences of random actions;
- We analyze the best performing models, highlighting the lack of a memory effect in user actions. Instead, our models are characterized by a law of diminishing returns;
- We demonstrate that similarities among the digital DNA strings of human accounts obey a log-normal law.

The investigated models could be used in the future in order to generate synthetic data that resemble real-world data, for experimentation purposes. Finally, for the sake of reproducibility, all the data used in this study are publicly available to the scientific community.

II. BACKGROUND AND DEFINITIONS

A. Digital DNA sequences

We define a digital DNA sequence as a row-vector of characters (i.e., a string).

\[ s = (b_1, b_2, \ldots, b_n) \quad b_i \in \mathbb{B} \quad \forall \quad i = 1, \ldots, n \]

Characters \( b_i \) in \( s \) are drawn from a finite set, called alphabet, \( \mathbb{B} = \{B_1, B_2, \ldots, B_N\} \quad B_i \neq B_j \quad \forall \quad i, j = 1, \ldots, N \quad i \neq j \)

The \( B_i \) characters are also called the (DNA) bases of the alphabet \( \mathbb{B} \). We can represent a user’s behavior with a digital DNA sequence by encoding each action of the user with an alphabet base. Then, by scanning the user’s actions in chronological order and by assigning the appropriate base to each action, we can obtain the sequence of characters that makes up the digital DNA sequence of the user. For example, Figure 2 shows the process of extracting the digital DNA sequence of a Twitter user, by scanning its timeline according to the alphabet \( \mathbb{B}_{type} \) defined in Table 1.

Table 1 contains the definitions of all the alphabets used in the remainder of this paper. These alphabets represent different possible encodings for OSNs actions, and have been already adopted in previous studies based on digital DNA [5], [7]. In detail, the \( \mathbb{B}_{type} \) alphabet encodes user behaviors according to the type of tweets produced, either tweets, retweets, or replies. Alphabets \( \mathbb{B}_{content} \) and \( \mathbb{B}_{content} \) provide a way to model Twitter actions, with different granularities, by looking at the content of tweets rather than the type. In order to easily classify a tweet based on its content we exploited Twitter’s notion of entities [8]. Instead, the alphabets \( \mathbb{B}_{interaction} \) and \( \mathbb{B}_{interaction} \) encode users behaviors by focusing on the popularity level of the users they interact with. Finally, in the alphabet \( \mathbb{B}_{account-age} \) the different DNA bases take into account the age of the accounts (i.e., time since the account’s creation) whom users interact with.

B. Similarity between digital DNA sequences

In order to analyze groups of users rather than single users, we need to study multiple digital DNA sequences as a whole. A group \( A \) of \( M = |A| \) users can be described by the digital DNA sequences of the \( M \) users,

\[
A = \begin{pmatrix}
S_1 \\
S_2 \\
\vdots \\
S_M
\end{pmatrix} = 
\begin{pmatrix}
(b_{1,1}, b_{1,2}, \ldots, b_{1,n}) \\
(b_{2,1}, b_{2,2}, \ldots, b_{2,m}) \\
\vdots \\
(b_{M,1}, b_{M,2}, \ldots, b_{M,p})
\end{pmatrix}
\] (1)

3For a complete reference of Twitter entities, see: [https://dev.twitter.com/overview/api/entities]
The group $A$ is defined as a column-vector of $M$ digital DNA sequences of variable length, one sequence for each user of the group.

To perform our analyses on digital DNA sequences, we can rely on recent advances in the fields of bioinformatics and string mining. In fact, several efficient algorithms and techniques for the analysis of biological strings have recently been proposed in such fields [8]. One of the possible means to quantify similarities between sequential data representations, such as our digital DNA sequences, is the longest common substring [3].

Intuitively, users that share long behavioral patterns are much more likely to be similar than those that share short behavioral patterns. Given two strings, $s_i$ of length $n$ and $s_j$ of length $m$, their longest common substring (henceforth LCS) is the longest string that is a substring of both $s_i$ and $s_j$. For example, given $s_i = \text{MASSACHUSETTS}$ and $s_j = \text{PARACHUTE}$, their LCS is the string ACHU and the LCS length is 4. The extended version of this problem that considers an arbitrary finite number of strings, is called the $k$-common substring problem [9]. In this case, given a vector $A = (s_1, \ldots, s_M)$ of $M$ strings, the problem is that of finding the LCS that is common to at least $k$ of these strings, for each $2 \leq k \leq M$. Notably, both the longest common substring and the $k$-common substring problems can be solved in linear time and space, by resorting to the generalized suffix tree and by implementing state-of-the-art algorithms, such as those proposed in [3].

Given that, in the $k$-common substring problem, the LCS is computed for each $2 \leq k \leq M$, it is possible to plot a LCS curve, showing the relationship between the length of the LCS and the number of strings. For example, figures 3a and 3b depict the LCS curves computed for a set of Twitter accounts via the $B^3_{\text{type}}$ and $B^3_{\text{content}}$ alphabets. On the $x$ axis is reported the number $k$ of accounts (corresponding to the $k$ strings, or digital DNA sequences, used to compute LCS values) and on the $y$ axis the length of the LCS common to at least $k$ accounts. Therefore, each point in a LCS curve corresponds to a subset of $k$ accounts that share the longest substring (of length $y$) among all those shared between all the other possible subsets of $k$ accounts.

A LCS curve is a representation of the behavioral similarities among a group of users, since it is an ordered sequence of substring lengths. To obtain a single value as measure of similarity for the whole group, we can compute the area under the LCS curve (AUC). Since LCS curves are discrete functions defined over the $[2, M]$ range, their AUC can be computed straightaway, without approximations, with the following trapezoid rule,

$$\text{AUC} = \frac{1}{2} \sum_{k=3}^{M} (\text{LCS}[k - 1] + \text{LCS}[k]) \Delta k$$

Compared to LCS, the definition of AUC given in Equation (2) allows us to quantitatively and directly compare the overall similarity among different groups.

### III. REAL-WORLD TWITTER DATASETS

In order to support the analyses presented in the remainder of the paper, we relied on both synthetic and real-world data from Twitter. While synthetic data are generated and documented on a per-experiment basis, in this section we introduce the details about the datasets used in all our experiments. Specifically, we spent several months collecting data about the activities of a random sample of legitimate accounts and of 3 different groups of bots. While not being the main focus of this work, data about the behaviors of the 3 bot groups might serve as reference in order to better understand the characteristics of online human behaviors.

In order to build our dataset of certified human accounts, we randomly contacted Twitter users by mentioning them in tweets about our research, and we further asked them simple questions in natural language. Possible answers to our questions were given by contacted users in the form of replies to our tweets. Thus, we exploited a Twitter crawler to collect all the tweets containing an answer to one of our questions. Such answers were then manually verified, and all the 3,474 accounts that answered were certified as legitimate.

Data about the bots is related to the activities of 3 groups of social spambots that have been recently detected in Twitter [1]. Specifically, the Bot1 group consists of a group of 991 spambots retweeters of a political candidate. Such accounts show a tweeting behavior which is apparently similar to those of genuine accounts, with a few tweets posted every day – mainly quotes from popular people. However, every time the political candidate posts a new tweet from his official account, all the automated accounts retweet it in a time span of just a few minutes. Spambot accounts belonging to the Bot2 group spent several months promoting the hashtag of a premium
app for mobile devices. Such spambots tweeted mentioning a specific legitimate (human) account and suggesting him to buy the VIP version of the app from a Web store. Finally, the Bot3 group consists of social bots whose intent is to advertise a subset of products on sale on an e-commerce platform. This time the deceitful activity is carried out by spamming URLs pointing to the advertised products.

Table [II] reports detailed statistics about the four real-world datasets used in this study.

IV. QUANTIFYING HETEROGENEITY AND RANDOMNESS OF ONLINE BEHAVIORS

Previous results with digital DNA showed evidence of high heterogeneity in legitimate behaviors [5], [7]. These results are also backed up by other recent studies conducted with different techniques [6], [10] and references therein. Thus, the question as to whether online behaviors are truly chaotic and driven by chance alone, or instead, whether it exists some law that drives them, resulting in minimal yet meaningful effects, it is yet to be investigated. A well known methodology to study this type of relations is based on null models and null hypotheses, introduced and developed for inferential statistics and graph theory [11], [12]. Namely, the differences between random and real instances of a phenomenon are used as evidences to confute the absence of a certain characteristic of the phenomenon. In our case, by relying on a large number of null models and statistical inference techniques, we aim at disentangling the actual properties of our LCS curves from the statistical effects due to alea.

To achieve this goal, in this section we propose a general and flexible algorithm for the construction of null models of digital DNA sequences. We leverage our algorithm to create 3 different kinds of null models, in which digital DNA sequences are randomly generated. Despite being composed of random sequences, the techniques used to create our models ensure that they preserve some important characteristics of the real sequences. In particular, our methodology can be described as follows. We firstly employ our proposed techniques to generate a large number of random sequences. Further, we gather such sequences so as to obtain groups that are comparable to our groups of both legitimate and bot accounts. We then analyze the groups of random sequences, following the same approach described in Section II for the analysis of legitimate and bot accounts. Finally, we compare results obtained from real data with those obtained from the null models and discuss the implications of our analysis.

A. Defining null models for digital DNA sequences

Given a group A of M accounts, defined in Equation (1), with a digital DNA derived from alphabet B, the task of defining a null model A for A lies in creating a set of M random sequences s so that,

\[
\hat{A} = \left( \begin{array}{c}
\hat{s}_1 \\
\hat{s}_2 \\
\vdots \\
\hat{s}_M
\end{array} \right) = \left( \begin{array}{c}
(\hat{b}_{1,1}, \hat{b}_{1,2}, \ldots, \hat{b}_{1,k}) \\
(\hat{b}_{2,1}, \hat{b}_{2,2}, \ldots, \hat{b}_{2,k}) \\
\vdots \\
(\hat{b}_{M,1}, \hat{b}_{M,2}, \ldots, \hat{b}_{M,k})
\end{array} \right)
\]

with \( \hat{b} \in B \)

In the above equation, the \( \hat{s} \) and \( \hat{b} \) symbols denote sequences and characters randomly generated. Thus, to create a null model, we have to generate a random sequence of characters for each account of the original group A. The characters used to create the random sequences are picked, with a given probability, among the bases constituting the B alphabet originally used to model the accounts of A. The pseudo-code of the general procedure for creating a null model is described by Algorithm 1.

```
input : A          // group of M DNA sequences
B            // alphabet of A
output: A      // group of M random DNA sequences
1 \( \hat{A} = ( ) \);
2 \( M = |A| \);
3 for \( i = 1 \) to \( M \) do
4 \( \hat{s}_i = ( ) \);
5 choose length \( \hat{n}_i \) for \( \hat{s}_i \) according to a given null model;
6 for \( j = 1 \) to \( \hat{n}_i \) do
7 choose base \( \hat{b}_{i,j} = B_k \in B \) according to a given null model;
8 \( \hat{s}_i.append(\hat{b}_{i,j}) \);
9 end
10 \( \hat{A}.push(\hat{s}_i) \);
11 end
12 return \( \hat{A} \);
```

Algorithm 1: General procedure for creating a null model A for the group of accounts A modeled through the alphabet B.

In Algorithm 1, the inner loop is responsible for choosing the characters that constitute a random sequence. At each iteration of the inner loop, each base \( B_k \in B \) is picked with probability \( Pr(B_k) \) and assigned to \( \hat{b}_{i,j} \). The character \( \hat{b}_{i,j} \) is then appended to the current random sequence \( \hat{s}_i \). The outer loop is responsible for creating the random sequences. At each iteration, it chooses the length \( \hat{n}_i \) of the current random sequence \( \hat{s}_i \) and, after the sequence is completed (i.e., the inner loop has terminated), it pushes \( \hat{s}_i \) inside the vector \( \hat{A} \). When the outer loop is completed the vector \( \hat{A} \) has length \( |\hat{A}| = |A| \), thus representing a null model for A.

The operations performed at lines 5 and 7 of Algorithm 1 determine the way in which a null model is constructed. To create a truly random null model we could just create sequences of random length by letting \( \hat{n}_i \) be a random variable following the discrete uniform distribution over the set \( \mathbb{N} \cap [1, \text{max}\{|s_i|\}] \) and by assigning each base \( B_k \in B \) to \( \hat{b}_{i,j} \) with an uniform probability \( Pr(B_k) = \frac{1}{|B|} \). However, this would generate a null model that resembles neither the lengths of the sequences s of A, nor the statistical distribution of the \( B_k \) bases in A. The usefulness of such a model as a contrast term with regards to the real sequences would therefore be very limited. Instead, following the same approach used in graph theory and network science for creating random graphs [13], we define 3 null models with similar characteristics to the original sequences of A, namely the average, bootstrap, and
In the average null model, each random sequence $\hat{s}_i$ has a length $\hat{n}_i^a$ equal to the mean of the lengths of the sequences $s_i$ of $A$,

$$\hat{n}_i^a = \frac{1}{M} \sum_{i=1}^{M} |s_i|$$

and the probability of assigning a base $B_k \in B$ to $\hat{b}_{i,j}$ is the ratio of characters $b = B_k$ among all the sequences of $A$,

$$Pr^a(B_k) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{M} f(b_{i,j}, B_k)}{\sum_{i=1}^{M} |s_i|}$$

with $f(b, B_k)$ defined as,

$$f(b, B_k) = \begin{cases} 1 & \text{if } b = B_k \\ 0 & \text{otherwise} \end{cases}$$

As defined by equations (3) and (4), the average null model creates random DNA sequences that resemble the average characteristics of the $M$ sequences of $A$. Instead, in the bootstrap and in the permutation null models, a specific random sequence $\hat{s}_i$ is created by performing a resampling of the corresponding real sequence $s_i$ of $A$. In detail, the length of a given random sequence $\hat{s}_i$ in both the bootstrap null model ($\hat{n}_i^b$) and in the permutation null model ($\hat{n}_i^p$), is equal to the length of the corresponding real sequence $s_i$,

$$\hat{n}_i^b = \hat{n}_i^p = |s_i|$$

Then, in the bootstrap null model the $\hat{b}$ bases of the $\hat{s}_i$ random sequence are assigned with a bootstrap of the corresponding real sequence $s_i$, i.e., by performing a sampling with replacement. Instead, in the permutation null model the $\hat{b}$ bases of the $\hat{s}_i$ random sequence are assigned with a permutation of the corresponding real sequence $s_i$, i.e., by performing a sampling without replacement. Intuitively, the bootstrap and the permutation null models will follow much more closely the characteristics of the real digital DNA sequences of $A$, with respect to the average null model.

B. Experimental results

Building on the definitions of the null models given in the previous Section IV-A, we ran a series of simulations in order to evaluate to what extent the similarity measured from real Twitter data and expressed by our LCS curves could be explained by our null models. In order to create a statistically reliable reference for our real DNA sequences, for each of the four groups of accounts of our dataset (legitimate accounts, Bot1, Bot2, and Bot3), we created 1,000 instances of each null model. Such instances of the null models are clearly different between one another, since they are created with the stochastic procedure described in Algorithm 1. This process led to the creation of 3,000 null models per group of accounts, counting a total number of 25,158,000 randomly generated DNA sequences. The 1,000 null models per type related to a group of accounts are then compared to the real LCS curve of that group.

In order to perform a qualitative comparison, in Figure 4 we show the real LCS curve of our groups of accounts (solid red lines) and the mean of the LCS curves of the 1,000 average, bootstrap, and permutation null models (dotted and dashed lines) created for each group of accounts. Figure 4 highlights significant differences between the LCS curves derived from real Twitter data and those derived from the 3 types of null models. As we were expecting, the bootstrap and permutation null models generate LCS curves that are much more similar to the real ones than those generated by the average null models. This is represented by both the values of the LCS curves which are overall lower, but comparable to that of the real sequences, and by the trends of the curves that are again comparable to the real ones. Notably, despite the highlighted similarities, the LCS curves derived from all the null models present lower values than the respective real curves. This suggests that the level of behavioral similarity measured for all the 4 groups of accounts is not due to alea. Creating synthetic sequences by filling them with random bases extracted from the same statistical distribution of the real sequences, clearly led to different results. Thus, a simple resampling of the real sequences erased some of the behavioral patterns that were present in our real data. This result is particularly relevant for
legitimate accounts. Indeed, qualitative results in Figure 4 and quantitative results introduced hereafter show that, although featuring limited common patterns, the behaviors of a group of human accounts are more similar than those measured by performing a series of random actions.

In addition to the qualitative study presented in Figure 4, we performed a quantitative analysis aimed at assessing the behavioral similarity among those accounts. For each group of accounts, we report the area under the real LCS curve. The higher is the AUC value of the LCS curve, we exploited the notion of area under the curve (AUC) introduced in Section II-B. The higher is the AUC value of the LCS curve, the behaviors of a group of accounts real data average bootstrap permutation null models for each block size and for each group of accounts. Sequences used for this experiment are obtained with the $\mathbb{B}_3$ alphabet.

![Fig. 5: Qualitative comparison between real LCS curves and 1,000 block-bootstrap null models for each block size and for each group of accounts. Sequences used for this experiment are obtained with the $\mathbb{B}_3$ alphabet.](image1)

![Fig. 6: Qualitative comparison between real LCS curves and 1,000 block-permutation null models for each block size and for each group of accounts. Sequences used for this experiment are obtained with the $\mathbb{B}_3$ alphabet.](image2)

Considering that the bootstrap and permutation null models are more representative than the average null models, for all the considered groups of accounts, we carried out more detailed experiments with the former null models. In this second batch of experiments, instead of performing the bootstrap and the permutation resampling one character at a time, we carried out our bootstraps and permutations at block level. That is, at every iteration of the null model algorithm, we build a random digital DNA sequence $s_i$ by picking groups of contiguous characters from the corresponding real sequence $s_i$, as shown in the toy example of Figure 7. The bigger the size of the block used for the bootstraps and permutations, the more behavioral patterns are likely to be preserved within the random DNA sequences $s_i$. Notably, block-based resamplings have already been adopted for the analysis of biological DNA [15]. Here, one of the aims of the experiment is to assess whether human behaviors in OSNs, as modeled by digital DNA sequences, feature a memory effect. That is, if an action that we perform at a given time occurs because of a given set (say $x$) of previous actions. In case it exists a memory effect of a given size $x$, then the AUC measured for null models with block size $= x$ should be greater than those measured for models with block size $\neq x$.

Table III: Quantitative comparison between AUC values of real LCS sequences and those measured for the null models.

<table>
<thead>
<tr>
<th>accounts</th>
<th>real data</th>
<th>average</th>
<th>bootstrap</th>
<th>permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bot1</td>
<td>$6.74 \times 10^5$</td>
<td>$1.43 \times 10^5$</td>
<td>$4.90 \times 10^5$</td>
<td>$4.63 \times 10^5$</td>
</tr>
<tr>
<td>Bot2</td>
<td>$5.37 \times 10^4$</td>
<td>$2.00 \times 10^4$</td>
<td>$2.19 \times 10^5$</td>
<td>$2.16 \times 10^5$</td>
</tr>
<tr>
<td>Bot3</td>
<td>$1.31 \times 10^8$</td>
<td>$0.18 \times 10^6$</td>
<td>$1.29 \times 10^6$</td>
<td>$1.29 \times 10^6$</td>
</tr>
<tr>
<td>Legitimate</td>
<td>$1.27 \times 10^5$</td>
<td>$0.31 \times 10^5$</td>
<td>$0.50 \times 10^5$</td>
<td>$0.50 \times 10^5$</td>
</tr>
</tbody>
</table>

Table III presents the results of this quantitative analysis. For each group of accounts, we report the area under the real LCS curve and the mean area under the LCS curves derived from the 3 types of null models. As shown in Table III, the AUC value for the real LCS curves is always higher than the mean values of the null models.
We experimented with block sizes ranging from 2 to 5, in addition to those with block size = 1 already presented in Figure 4 and Table III. To run this second experiment on block null models we created 67,088,000 additional sequences. Results of our experiments for block-bootstrap null models are presented in Figure 5, while results for block-permutation null models are presented in Figure 6. In addition, Table IV reports a comparison of AUC values measured for LCS curves derived from real data versus those derived from block-bootstrap and block-permutation null models. By increasing the block size, we are able to generate DNA sequences that are increasingly similar to the real ones. This is pictorially shown in figures 3 and 9 by the family of LCS curves of the null models that get closer to the curve obtained with real data, at each increase in block size. However, from the analytical data reported in Table IV we see that there is not a block size for which we measure a local maximum in AUC, but rather that the AUC keeps increasing together with the block size. This result seems to support the lack of a memory effect in the sequence of our online actions. To better analyze this facet for human-operated accounts, in figures 8a and 9a we report the percentage of AUC of the real sequences explained by the null models, as a function of the block size. In addition, figures 8b and 9b show the percentage increment in explained AUC for null models with block size = x, with respect to the null models with block size = x − 1. This helps to highlight the increment in AUC obtained when increasing the block size. As shown, rather than a memory effect for a specific block size, we witness to a law of diminishing returns, meaning that each subsequent increment in block size results in an increasingly small AUC improvement.

V. A FUNDAMENTAL LAW CHARACTERIZING ONLINE HUMAN BEHAVIORS

In the previous section we have shown that our online behaviors, although seemingly uncorrelated and highly heterogeneous, are not completely random. Thus, leveraging this finding, in this section we aim at verifying whether a fundamental law exists that drives the online behaviors of legitimate users, as described by LCS curves. Characterizing our behaviors is interesting for a number of reasons. In fact, a deeper understanding of our online behaviors opens up the possibility to leverage such information for innovative applications. In addition, models of our online behaviors can represent a benchmark when comparing the behaviors of unknown or suspicious accounts.

Figures 3a and 3b show the LCS curves of human accounts obtained with the $E_{type}$ and the $E_{content}$ alphabets, as defined in Table I. The low values of LCS in the figures testify that human accounts share a limited number of behavioral similarities. The depicted scenario is typical of heavy-tailed phenomena, where very few accounts feature common behaviors, while the vast majority share little to no common behaviors. Although similarities among the online behaviors of groups of users have never been investigated this way, the result is not unexpected. In fact, heavy-tailed phenomena have been witnessed in several expressions of human dynamics [16].

In the following, we investigate the link between the empirical data leading to the LCS curves obtained with all the DNA alphabets of Table I and well-known statistical distributions. Fitting empirical data to unveil their underlying statistical distribution is a frequent task in computer physics. Given the peculiar heavy-tailed shape of our LCS curves, we follow the approach in [17], [18] for fitting power law distributions. Such an approach can be extended to fit any kind of statistical distribution. The procedure we adopted is as follows:

1) choose a set of plausible theoretical models: in this step, we choose the statistical distributions that rep-
represent plausible hypotheses to describe our empirical data. Given the informal nature of this step, descriptive statistics and plots of the empirical data (such as the ones in Figure 3) can provide valuable hints;

2) estimate the model parameters: for each theoretical model proposed in the previous step, we estimate the parameters that best fit our empirical data;

3) evaluate the competing theoretical models: we assess the goodness-of-fit of the theoretical models – with the estimated parameters – to our empirical data. The theoretical model that shows the best fit is the one that best explains our empirical data.

As our theoretical models, we consider the Poisson distribution and three well-known heavy-tailed distributions, namely the power law (or Pareto), the log-normal, and the exponential distributions. As parameters, in addition to the characteristic parameters for a specific distribution (e.g., the scaling parameter $\alpha$ for a power law distribution), we also need to estimate the lower cutoff value – namely, the lowest LCS value for which our empirical data follows that distribution. In practice, few empirical phenomena follow a distribution for all values of their domain [17]. More often, such behavior applies only for values greater than some minimum $x_{\text{min}}$. Therefore, as suggested in [17], [18], for each theoretical model we first estimate the $x_{\text{min}}$ parameter, by means of the Kolmogorov-Smirnov statistic. Then, we employ a maximum likelihood estimation (MLE) method to estimate the characteristic parameters of the distribution, starting from $x_{\text{min}}$. We then fit our empirical data with the theoretical models defined by the estimated parameters. As done in [17], [18], we execute the model fitting and the goodness-of-fit evaluation on the complementary cumulative distribution function (CCDF) of our LCS curves. The CCDF of a random variable $X$ is denoted by $P_X(x)$ and is defined as the probability that the random variable $X$ takes on a value greater than or equal to $x$,

$$P_X(x) = \Pr(X \geq x)$$

The goodness-of-fit evaluation can be based on any well-known metric of distance between statistical distributions. In detail, we aim at measuring the distance between our empirical data $P_X(x)$ and each of the theoretical distributions introduced before and denoted hereafter by $\hat{P}_X(x)$. In our study, we exploited the Kolmogorov-Smirnov distance ($KSD$) and the Kullback-Liebler distance ($KLD$) as our evaluation metrics. Intuitively, $KSD$ measures the maximum distance between the two considered distributions, over their domain,

$$KSD(P_X, \hat{P}_X) = \max_x |P_X(x) - \hat{P}_X(x)|$$  \hfill (5)

$KLD$ is instead an information theoretic metric that measures how much information is lost when a given distribution $\hat{P}_X(x)$ is approximated by $P_X(x)$. In detail, $KLD$ is the symmetric version of the Kullback-Liebler divergence $KLd$ defined as,

$$KLd(P_X, \hat{P}_X) = \sum_x \ln \left( \frac{P_X(x)}{\hat{P}_X(x)} \right) P_X(x)$$

thus,

$$KLD(P_X, \hat{P}_X) = \frac{KLd(P_X, \hat{P}_X) + KLd(\hat{P}_X, P_X)}{2}$$  \hfill (6)

As a result of equations (5) and (6), the lower the $KSD$ and $KLD$ for a fitted model, the better the fit.

Table IV reports the quantitative results of the parameters estimation and model fitting for the LCS curves computed considering digital DNA sequences encoded according to the alphabets in Table I. Qualitative results of model fitting are shown in Figure 10. Both our quantitative and qualitative results suggest that LCS curves of human Twitter accounts obey a log-normal distribution. This result is in line with recent work showing evidence of log-normal distributions in online human behaviors [19], [20], [21]. A phenomenon following a log-normal distribution, as in the case of all heavy-tailed distributions, is characterized by a few events of exceptionally high magnitude and by a multitude of almost insignificant events, with all intermediate values still present. In fact, no characteristic or typical value (i.e., the mean) for this kind of phenomena can be (meaningfully) defined, as demonstrated by a standard deviation that is orders of magnitude larger than the mean. In our scenario, this translates into a few human accounts sharing long behavioral patterns, while the
remaining vast majority of accounts share short sequences in common. Heavy-tailed distributions of phenomena related to our online actions have recently been explained as being a direct consequence of human heterogeneity [16], [22]. Thus, apart from a few exceptions, the human accounts in our datasets show strong evidence of behavioral heterogeneity. Although high behavioral similarities are possible according to the log-normal distribution, the probability of witnessing to a large number of such events is extremely low, as demonstrated by the CCDFs of human LCS curves. Therefore, this strict characterization of the LCS curves of human accounts can also be exploited as a reference to highlight anomalous or suspicious behaviors.

VI. RELATED WORK

With the widespread adoption of OSNs, search engines, and e-commerce platforms, we assisted to the rapid proliferation of many studies on modeling and analysis of online behaviors [23]. However, such behavioral analytics studies present clear challenges, since “user behaviors depends on content, intentions, and contexts” in complex online scenarios [10]. Adding to these challenges are recent findings, such as those related to the online inter-event time distributions (referred to as interval patterns), showing that individuals have persistent and distinct interval patterns [6], and those related to our tweeting behaviors [5], [7]. All such studies highlighted that our online behaviors are very diverse and heterogeneous. As such, to date very few works focused on a high-level, general study of our online behaviors. Instead, the majority of works focused on specific tasks, such as analyzing specific behavioral patterns [24], [25], [26], predicting future behaviors [27], [28], and detecting anomalous ones [5], [29], [7].

As an example of this research trend, [29] considers the detection of Twitter social spambots through machine learning, and the quantification of the number of bots versus that of genuine accounts. The authors of [30] consider the adoption of positive habits in people’s life, by analyzing the changing behaviors of Fitbit users, as they report about their trainings on the Fitbit social network and they influence each other with such reports. Work in [31] considers pictures from WeChat, an extensively used instant messaging Chinese platform, by demonstrating the existence of correlations between self-posting behaviors and personal patterns of the users (e.g., preferences towards high-level attributes like travel, food, cosmetics, and children). The authors of [32] consider a large amount of web access logs, with the intent of categorizing and relating them to activities in real life. Results of the analyses show that it is possible to correlate a series of web access logs with the habits to travel, both domestically and internationally.

While the above list does not intend to be exhaustive, it nonetheless testifies the broad variety of investigated scenarios. In this paper, we study online behaviors from a different perspective since we aim at verifying if a general model exists for approximating our behaviors, regardless of the precise
contextual activities the users are carrying out. To this end, our study is similar to other recent works that studied online behaviors in a context-agnostic setting [19], [20], [21].

VII. CONCLUSIONS

In this paper, we provided several contributions. By analyzing the collective behaviors of Twitter users, we demonstrated that human behaviors, although extremely heterogeneous, are not comparable to sequences of random actions. Further, we provided a flexible, constructive solution to generate behavioral models of human accounts. Moreover, we proposed different models of collective behaviors, with the best performing model capable of explaining more than 50% of the behavioral similarity of a real group of users. We showed evidence that user online activities lack a memory effect. Instead, they are characterized by diminishing returns. Mostly important, we demonstrated that the emergence of behavioral similarities among a group of users follow a log-normal law.

As future work, we aim at verifying whether these findings hold for other types of OSNs. We will also consider more sophisticated models, able to better adhere to real behaviors. Indeed, the diminishing returns effect suggests that null models relying on block bootstraps and permutations fail in efficiently resembling the real LCS curves.

ACKNOWLEDGMENTS

This research is supported in part by the EU H2020 Program under the schemes INFRAIA-1-2014-2015: Research Infrastructures grant agreement #654024 SoBigData: Social Mining & Big Data Ecosystem and MSCA-ITN-2015-ETN grant agreement #675320 European Network of Excellence in Cybersecurity (NECS). Funding has also been received from the Registro.it IIT-CNR project MIB (My Information Bubble), from Fondazione Cassa di Risparmio di Lucca that partially finances the regional project Reviewland, and from the MIUR (Ministero dell’Istruzione, dell’Università e della Ricerca) and Regione Toscana (Tuscany, Italy) funding the SmartNews: Social sensing for Breaking News project: PAR-FAS 2007-2013.

REFERENCES

[31] T. Chen, Y. Chen, and J. Luo, “A selfie is worth a thousand words: Mining personal patterns behind user selfie-posting behaviours,” in WWW’17 Companion. ACM.