Projet S5 2017 linkstream State of the Art : Online communities dynamics analysis

Amine AKKI
Frédéric TAMAGNAN
Khaoula GHOUIBI

March 4, 2018

1 Introduction

On-line communities are a pillar of the web and represent a huge part of the Internet traffic. The importance of on-line communities have risen in popularity, since they have become one of the most crucial part of the modern companies marketing strategy. In fact, in digital marketing client relationship management is fundamental to the success of companies. Hence, ensuring the development and the flourishing of these communities is a concern of community managers and owners. The real issue is that monitoring technics are still empirical. In fact, community managers still use simple indicators such as the frequency and the number of messages exchanged as well as the development of the number of participants. What companies need are tools that allows them to follow the dynamic of interactions within the community, and help them understand the real causes behind the tendencies within their own network. These tools have to deal with a much more complex context since these communities are different from social media where people clearly state their relations with other people. It is complicated to identify the tendency of interaction based only on empirical indicators, thus the new methods of analysis must automatically detect and qualify the interaction of interactions where participants share the same subject on a forum or co-fund the same project.

2 KPIs

Social media is quickly becoming one of the most important channels businesses can use to interact with their customers. Similar to every other form of communication out there, it is important to arm any company's business with a smart, nimble team to ensure every inbound message and outbound opportunity is addressed. This is exactly what the community manager do. In fact, a community manager is responsible for advocating the brand on social networks. They create their own social persona and actively go out within the online community to connect with customers and understand the dynamics of the online community. Community managers typically help build, grow and manage a company's or brand's online communities. Using analytics tools to monitor social media outlets, online forums and blogs, a community manager finds out what the community looks like and what it's potential evolution.

We decided to start from the lithium.com vision of community management. On their point of view we can analyze a community depending on different axis. Based on their axis, we have listed eight axis of analysis.

1. Reactivity
2. Interaction between members
3. Number of members and its variation
4. Quality of produced content
5. Traffic
6. Roles detection : superfans, influencers, followers
7. Anomalies detection
8. Pattern detection

It is hard to focus on the whole list so, we are going to focus in this paper on the tools that help a community manager to analyze an online community and to understand its dynamics through some KPIs : Δ-density, Δ-clique, entropy, etc,...

We denote a linkstream as $L = (T, V, E)$ with $T = [\alpha, \omega]$ and $E \in T \times V \times V$, as a set of interactions over time. $l = (t, u, v) \in E$ means that an interaction occurred between $u \in V$ and $v \in V$ at time $t \in T$.

2.1 Interaction between Members

- Average number of members each member has interacted with from the beginning.
  Let $m \in V$ then V_m =$\{\ v \in V \mid (t, v, m) \in E\ \}$
  γ=$\frac{\sum_{m=1}^{card(V)} card(V_m)}{card(V)}$
- Average number of members each member has interacted with during the last Δ period;
  Let $m \in V$ then $V_m\Delta$ =$\{\ v \in V \mid (t, v, m) \in E \mid \omega - \Delta \leq t \leq \omega\ \}$
  γ_Δ=$\frac{\sum_{m=1}^{card(V)} card(V_m\Delta)}{card(V)}$

2.2 Number of members and variations

- Number of members who have never been part of the linkstream N
  $N = card(\{v \in V | \nexists e \in E | e = (t, u, v) | u \in V | t \in T\})$
- Number of members who haven't made any interaction during 100 Δ
  $N_{100\Delta} = card(\{v \in V | \nexists e \in E | e = (t, u, v) | u \in V | \omega - 100\Delta \leq t \leq \omega\})$

2.3 Traffic and reactivity

- Number of interaction on a Δ period I_Δ
  $I_\Delta = card(\{e \in E | e = (t, u, v) | u, v \in V^2 | \omega - \Delta \leq t \leq \omega\})$
- Average time between two interactions on a Δ period
  Let $T_e = \{t \in T | \exists e \in E | e = (t, u, v) | (u, v) \in V^2\}$
  T_Δ=$\frac{\sum_{\forall t_i \in T_e} t_{i+1} - t_i}{I_\Delta}$

2.4 Anomalies detection : Variations of in-messages and out-messages

In this subsection we consider here directed links, i.e. we make a distinction between (t, u, v) and (t, v, u), the second member of the tuple is the sender and the third is the receiver.

- Average number of received interactions by the members who have received interactions on Δ period
  Let $V_i = \{v \in V | \exists e \in E | e = (t, u, v) | u \in V | \omega - \Delta \leq t \leq \omega\}$
  Let I_V = $\{e \in E | e = (t, u, v) | v \in V_i | uinV | \omega - \Delta \leq t \leq \omega\}$
  Average number of received interactions by the members who have received interactions on Δ period= $card(I_V)/card(V_i)$

  This KPI permits to have an idea of how many messages, active people have received, and to detect spam. It is better to compute this mean on the base of "active people on a Δ Period" (only people who have received messages on the delta period) rather than on the base of all people in the linkstream.

  We can imagine a lot of other KPI like interaction rate, number of interaction divided by the cardinal of all possible interactions etc

2.5 KPIs more complex

2.5.1 Density

We work on a linkstream of a total duration of L. We suppose a duration Δ between 0 and L is given. We first define the Δ-density of a pair of nodes u and v, that we denote by $\delta(u, v)$.

Density in a graph is the probability that a link exists between two randomly chosen nodes. Similarly, we define the Δ-density of (u, v) as the probability that a randomly chosen time-interval of size Δ contains (at least) an occurrence of (u, v) [NGL]. In other words, the Δ-density of (u, v) measures the extent at which (u, v) occurs (at least) every Δtime, or conversely the fraction of time-intervals of duration Δ that contain (at least) an occurrence of (u, v). We define the Δ-density by the following expression:

$$\delta_\Delta(u,v) = 1 - \frac{\sum_i \max(\tau_i - \Delta, 0)}{\omega - \alpha - \Delta}$$

In order to extend the notion of Δ-density to any set S of links, we define it as the average of the Δ-density of the elements of S:

$$\delta_\Delta(S) = \frac{\sum_{(u,v)\in S} \delta_\Delta(u,v)}{|S|}$$

Note: In order to study the delta-density in our data, we first have to choose an appropriate value of delta. This is usually done by testing different values of delta and choosing the most appropriate one.

2.5.2 Clique

A clique, C, in an undirected graph $G = (V, E)$ is a subset of the vertices, $C \subseteq V$, such that every two distinct vertices are adjacent. This is equivalent to the condition that the induced subgraph of G induced by C is a complete graph. In some cases, the term clique may also refer to the subgraph directly. We generalize the classical notion of cliques in graphs to such link streams: for a given delta, a Δ-clique is a set of nodes and a time interval such that all pairs of nodes in this set interact at least once during each sub-interval of duration Δ. We denote a linkstream as $L = (T, V, E)$ with $T = [\alpha, \omega]$ and $E \in T \times V \times V$, as a set of interactions over time. $l = (t, u, v) \in E$ means that an interaction occurred between $u \in V$ and $v \in V$ at time $t \in T$.

For a given duration Δ, a Δ-clique C of L is a pair $C = (X, [b, e])$ with $X \in V$ and $[b, e] \in T$ such that $|X| > 2$, and for all $u, v \in X$ and $t_0 \in [b, max(e - \Delta, b)]$ there is a link $(t, u, v) \in E$ with $t \in [t_0; min(t_0 + \Delta, e)]$. Notice that Δ-cliques necessarily have at least two nodes.

More intuitively, all nodes in X interact at least once with all others at least every Δ from time b to time e. Δ-clique C is maximal if it is included in no other Δ-clique, (i.e. there exists no $\Delta-$clique $C_0 = (X_0; [b_0; e_0])$ such that $C_0 \neq C$,$X \in X_0$ and $[b, e] \in [b_0, e_0]$). [TV16]

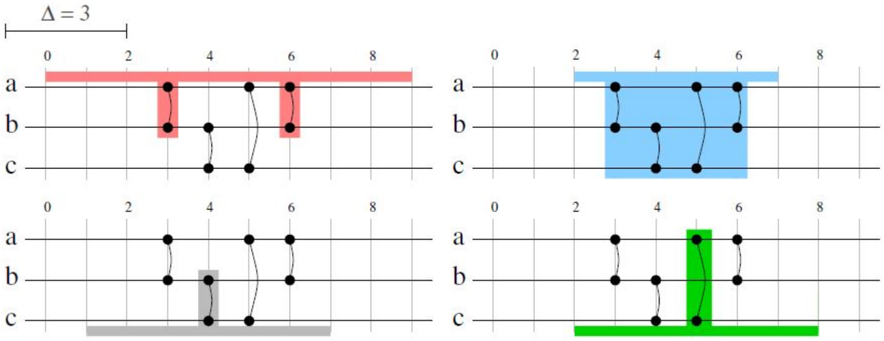
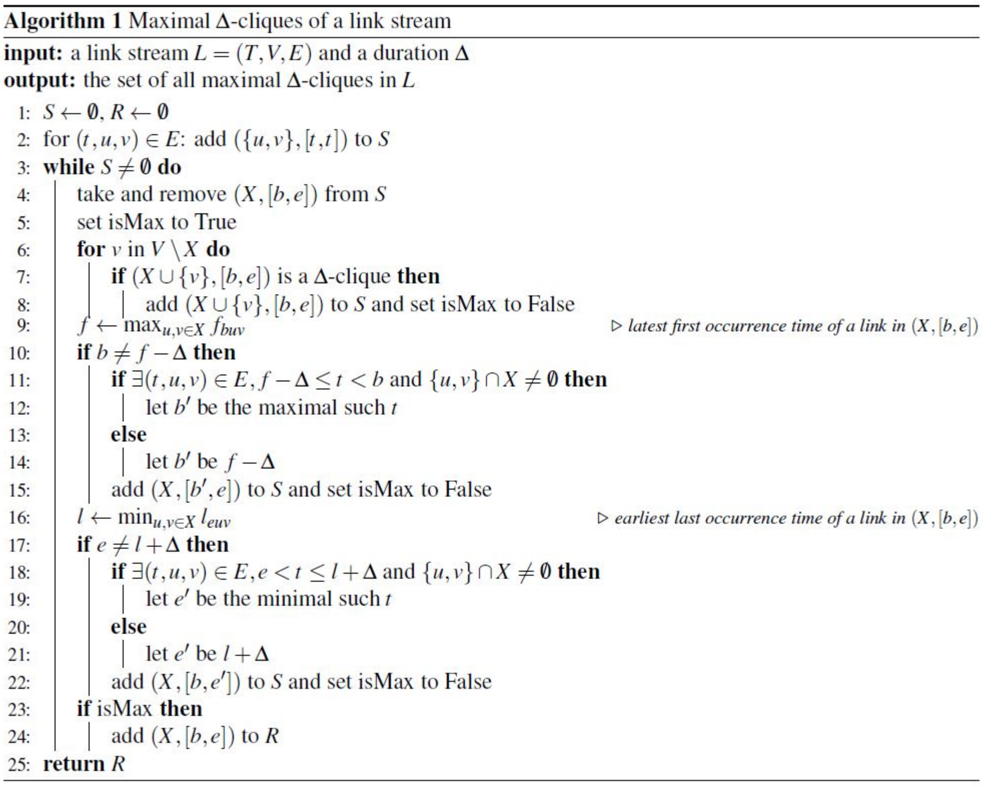


Figure 1: Examples of Δ-cliques

In real-world situations, Δ-cliques are signatures of meetings, discussions, or distributed applications for instance. Moreover, just like cliques in a graph correspond to its subgraphs of density 1, Δ-cliques in a link stream correspond to its substreams of Δ-density 1. Therefore, Δ-cliques in link streams are natural generalizations of cliques in graphs. An algorithm was proposed by Tiphaine Viard, Matthieu Latapy and Clémence Magnien [TV16] for listing all maximal Δ-cliques of a given link stream which is decribed as the following:

One may trivially enumerate all maximal cliques in a graph as follows. One maintains a set M of previously found cliques (maximal or not), as well as a set S of candidate cliques. Then for each clique C in S, one removes C from S and searches for nodes outside C connected to all nodes in clique C, thus obtaining new cliques (one for each such node) larger than C. If one finds no such node, then clique C is maximal and it is part of the output. Otherwise, if the newly found cliques have not already been found (i.e., they do not belong to M), then one adds them to S and M. The set S is initialized with the trivial cliques containing only one node, and all maximal cliques have been found when S is empty. The set M is used for memorization, and ensures that one does not examine the same clique more than once.

The algorithm for finding Δ-cliques in link stream $L = (T, V, E)$ (Algorithm 1) relies on the same scheme. We initialize the set S of candidate Δ-cliques and the set M of all found Δ-cliques with the trivial Δ-cliques $(a, b, [t, t])$ for all $(t, a, b) \in E$ (Line 2). Then, until S is empty (while loop of Lines 3 to 24), we pick an element $(X, [b, e]) \in S$ (Line 4) and search for nodes v outside X such that $(X \cup v, [b, e])$ is a Δ-clique (Lines 6 to 10). We also look for a value $b_0 < b$ such that $(X, [b_0, e])$ is a Δ-clique (Lines 11 to 16), and likewise a value $e_0 > e$ such that $(X, [b, e0])$ is a Δ-clique (Lines 17 to 22). If we find such a node, such a b_0 or such an e_0, then Δ-clique C is not maximal and we add to S and M the new Δ-cliques larger than C we just found (Lines 10, 16 and 22), on the condition that they had not already been seen (i.e., they do not belong to M). Otherwise, C is maximal and is part of the output (Line 24).



```
Algorithm 1 Maximal Δ-cliques of a link stream
input: a link stream $L = (T,V,E)$ and a duration $\Delta$
output: the set of all maximal $\Delta$-cliques in $L$
1:  $S \leftarrow \emptyset, R \leftarrow \emptyset$
2:  for $(t,u,v) \in E$: add $(\{u,v\},[t,t])$ to $S$
3:  while $S \neq \emptyset$ do
4:      take and remove $(X,[b,e])$ from $S$
5:      set isMax to True
6:      for $v$ in $V \setminus X$ do
7:          if $(X \cup \{v\},[b,e])$ is a $\Delta$-clique then
8:              add $(X \cup \{v\},[b,e])$ to $S$ and set isMax to False
9:      $f \leftarrow \max_{u,v \in X} f_{buv}$        ▷ latest first occurrence time of a link in $(X,[b,e])$
10:     if $b \neq f - \Delta$ then
11:         if $\exists (t,u,v) \in E, f - \Delta \leq t < b$ and $\{u,v\} \cap X \neq \emptyset$ then
12:             let $b'$ be the maximal such $t$
13:         else
14:             let $b'$ be $f - \Delta$
15:         add $(X,[b',e])$ to $S$ and set isMax to False
16:     $l \leftarrow \min_{u,v \in X} l_{euv}$        ▷ earliest last occurrence time of a link in $(X,[b,e])$
17:     if $e \neq l + \Delta$ then
18:         if $\exists (t,u,v) \in E, e < t \leq l + \Delta$ and $\{u,v\} \cap X \neq \emptyset$ then
19:             let $e'$ be the minimal such $t$
20:         else
21:             let $e'$ be $l + \Delta$
22:         add $(X,[b,e'])$ to $S$ and set isMax to False
23:     if isMax then
24:         add $(X,[b,e])$ to $R$
25: return $R$
```

Figure 2: Algorithm : Maximal Δ-cliques of a link stream

2.5.3 Entropy

The use of entropy to describe the fitness of a network was first introduced in the field of biology. However, “in technological networks, variation can be understood as innovation and selection pressures arise as the result of competition for new users. One may postulate that the resilience of such networks will be the determining factor in deciding the outcome of such competition.” (Robustness and network evolution—an entropic principle Lloyd Demetrius, Thomas Manke).[LD04]

The entropy can be used as a criterion to describe the evolution of the network and determine its robustness. Actually, it can be seen as the capacity to remain functional in face of perturbation. In this light, we can say that this concept of robustness is crucial to a community manager since it tells the degree of resilience of the community against perturbation such as the withdrawal of a member of the appearance of new members. The algorithms for community identification are closely related to the family of algorithms for clustering. The goal of clustering is to discover groups of similar objects within the data. Each cluster (i.e. group), contains objects that are similar to one another within the same cluster, and dissimilar to the objects in other clusters. In fact, There are multiple methods for clustering that can be used, such as: hierarchical clustering, partitioning, graph clustering methods, modularity based approach and block models [PBS15].

Many entropies were tested in many work papers on different datsets in order to find the accurate and the most describing entropy for each case. We will develop some of these entropies and show the importance of each one.

2.5.4 Shannon entropy

As mentionned in -Computer Information Systems and Industrial Management- by Khalid Saeed and Václav Snášel [KS14], the Shannon entropy was first introduced in the information theory, in 1948, as a measure of the uncertainty associated to a random variable. Actually, this entropy increase proportionally with the randomization of the associated variable. "The greater certainty of the variable, the smaller the entropy"[KS14]. For a probability distribution $p(X = x_i)$ of a discrete random variable X, the Shannon entropy is defined as:

$$H_s(X) = \textstyle\sum_{i=1}^{n} p(x_i) \log_a \frac{1}{p(x_i)}$$

X is the feature that can take values $x_1 ... x_n$ and $p(x_i)$ is the probability mass function of outcome x_i. "Depending on the base of the logarithm, different units can be used: bits ($a = 2$), nats ($a = e$) or hurtleys ($a = 10$)"[PBS15]. The Shannon entropy is also used for network anomaly detection where the probabilities $p(x_i)$ refer to the occurance of x_i in different time windows. A potential application of this entropy is anomaly detection in the linkstream model where the time dimension is taken into consideration. This way we could find out what could be wrong with an on line community.

2.5.5 Parameterized Entropy

As mentionned in the article -An Entropy-Based Network Anomaly Detection Method- by Przemysław Berezinski, Bartosz Jasiul and Marcin Szpyrka) [PBS15]: "the Shannon entropy assumes a tradeoff between contributions from the main mass of the distribution and the tail. To control this tradeoff, two parameterized Shannon entropy generalizations were proposed, by Renyi (1970s) and Tsallis (late 1980s) respectively." (..) "If the parameter denoted as α (or q) has a positive value, it exposes the main mass (the concentration of events that occur often), if the value is negative – it refers to the tail (the dispersion caused by seldom events). Both parameterized entropies (Renyi and Tsallis) derive from the Kolmogorov-Nagumo generalization of an average."

2.5.6 Gurevich Entropy

Gurevich entropy is a measure of the entropy of a graph. Therefore, it is easily used in the study of social networks. This theory is essential in the study of networks because it offers a better understanding of the nature of links in the graph. In fact, it takes into consideration the different paths of different lengths.
Let's consider a graph G and P it's incidence matrix Pi,j is the number of paths between I and j, and thus Pni,j represents the number of paths, with a length of n between I and j. The entropy of Gurevich is defined as follows: [BL07]

$$(1/n) * log(P_(i,j(n)) = \limsup(1/n * log((P_(i,j)^n) \qquad (1)$$

A low entropy is a sign of simple semantics of the graph and a high one is a sign of a higher complexity of the semantic. For instance, a hub like graph has a low Gurevich entropy.
The semantic of a graph can be interpreted in the case of social networks by the interactions that can be done. In fact, in the case of hub like networks, the number of interactions is limited (the hub send messages to the other nodes) and when the possibilities of interactions get higher the entropy gets higher too. This shows that the richer paths there are between different nodes (different way to reach a node) the higher is the entropy. Thus, Gurevich entropy is an indicator of the richness of the interactions within the network.
The limitation of this entropy is that it only encodes simple interaction without differentiation

of the modes of interaction. In other words, in a complex network, there are different types of interaction like the “like”, “share”, “subscribe”, ...
The Gurevich entropy considers all interactions likewise.

2.5.7 Transfer Entropy

The growth of online networks has led to an important exchange in information within the network and the concept of transfer entropy (TE) [Sat14]
was introduced to express the influence of the member of the community on other members. In fact, this use of the entropy is used to determine the influencer in the community and to describe and predict the tendencies of information spreading. To do so, the activity of the members is modeled as time series containing different patterns. The transfer entropy is measured on a couple of members, to determine the influence the activity of a member X on another member Y.
The activity of a member of the community can be encoded with an alphabet (in this example =(0,1) with 1 is the presence of an activity). And X and Y are considered as two discrete time processes.

$$TE_x - y = H(Y_n + 1|Y_n^k) - H(Y_n + 1|Y_n^k, X_n^k) \tag{2}$$

Y_n^k is a vector of the history of the activity of Y , $Y_n^k = [Y_n, Y_(n-1), \ldots, Y_(n-k+1)]$, we define X the same way.//

The transfer entropy measures the influence of the activity of X on the activity of Y. In fact, the more X Influences Y the lower is $H(Y_(n+1)|Y_n^k, X_n^l)$, and thus the higher TE we obtain.

This measure provides a way to determine the influential members of an online community. However, this metric can be difficult to implement. First, the length of the history of X and Y should be chosen carefully and can vary depending on the community we're dealing with. Secondly, the sampling of the activity of X and Y can have a huge impact on the values obtained. Indeed, when sampling we tend to lose some valuable information. Moreover, the frequency of sampling may influence the accuracy of the TE, when using a low frequency, it is easier to see some similarities in activity between two members.

Furthermore, in the definition above, we used a binary alphabet to describe the activities of the community members, using a richer alphabet may influence the accuracy of the TE, as some users tend to follow other users in some activities and not in others. Thus, it may be interesting to measure a transfer entropy of each activity and then to measure the influence of some activities on other activities, so that we can determine whether some actions trigger other activities.

Finally, this measure must be calculated on each couple of nodes and the choice of the right parameters is not so trivial. This can cost in runtime and complexity, and considering the fact that the goal is to offer a real-time description of the dynamic within the community, the use of TE may be hard to implement to determine the influential users and the patterns of behavior within the community.

2.5.8 Degree Distribution based Entropy

Most networks can be described by a degree distribution P_k. However, we will study the remaining degree: "the number of edges leaving the vertex other than the one we arrived along" [SV04]. This new distribution $q(k)$ is obtained from:

$$q(k) = \frac{(k+1)P_k}{< k >} \tag{3}$$

where $< k >= \sum_k kP_k$.

Actually, by using the previous distribution q, an entropy measure can be defined as follow:

$$H(q) = -\sum_{k=1}^{N} q(k) \log(q(k)) \tag{4}$$

As we mentioned earlier, the entropy of a network gives a measure of uncertainty. Within this context, this new definition of the entropy (the Degree Distribution based Entropy) gives a measure of the divesity of the linkstream and provides as well an average measure of network's heterogeneity.
As a result, this entropy helps measuring the amount of correlation between nodes in a graph and the average diversity associated to the $q(k)$ distribution.

3 Role identification

One of the main indicators of the health of the community is the number of members. However, this metric is just a simple indicator and does not tell much about the real dynamic of the and the behavioral parameters within the community. The role identification is crucially important in networks, it allows the community manager to follow the influential members and determine their tendencies of interaction. It also allows community manager to link behavior patterns with the health of the community and the contribution of the users to the flow of information. Thus identifying classes or clusters of likewise users is important in order to identify their influence and their behavioral pattern. To do so there are different ways to identify roles within a community some are based on the mere statistic and the structure of a member of the community others involve the use of the other ratios and indicators.
In general clustering users into users group can be done by using features covering the structure of the neighborhood of a certain user, the user's popularity, the user's tendency to initiate a discussion and his persistence in the information flow[SA04]. It is important to note that these feature depend mainly on the nature of the network and the types of interactions used within the community.

- In-degree Ratio: the number of users that reply to a certain user vi brought to the total number of users. This indicator represents the proportion of members who reply to vi and the concentration of responses.

- Thread Initiation Ratio:Proportion of threads that have been started by a certain user vi. This indicator allows to catch the user who are more active than other and who are more likely to move the community and to create information and value within the network. This class of users contains the members who are more influential than other users and who tend to generate ideas and content.

Guimera and Amaral Other methods use other metrics and set a threshold to separate the different users cluster. One of these methods is presented by Guimera and Amaral, in -Cartography of complex networks: modules and universal roles-, who developed a process to identify the roles: Guimera and Amaral start by defining for each node two metrics and then defining thresholds to separate different groups of users. Each group represents a role within the network. The two metrics are: Within-module degree it represents the node's connectivity with the community. It is based on the z-score concept. It is defined for a certain function f(u), where u is a node of the network: [GA05]

$$z(u) = \frac{f(u) - \mu(f)}{\sigma(f)}$$

μ is the mean of f in the community and σ is the standard deviation of f in the community. Hence, in the Guimera and Amaral module the within module degree is the z-score of the of the degree (ie in this precise case f represents the degree of each node). The second measure is the "partition coefficient", introduced also in -Cartography of complex networks: modules and universal roles- by Roger Guimerà and Luís Nunes Amaral, it characterizes the extra-connectivity of a node with other communities, it is defined as: [GA05]

$$P(u) = 1 - \sum_i (\frac{d_i(u)}{d(u)})^2$$

Where $d_i(u)$ represents the degree of u with the nodes of the community i. This metric determines the diversity of the links he node u have in general. Basically, a coefficient of 1 means that the node has diverse connections with all the communities and that these connections are well distributed.

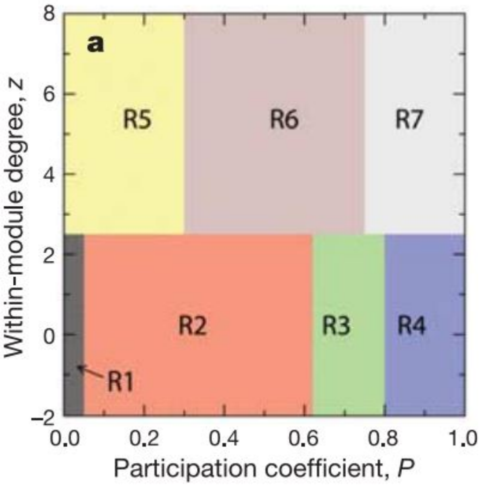


Figure 3: Partition of the space into 7 different roles. (from [GA05])

Role	z	P
non-hub, ultra-peripheral nodes	z < 2.5	P < 0.05
non-hub, peripheral nodes	z < 2.5	0.05< P < 0.62
non-hub, connector nodes	z < 2.5	0.62 < P < 0.80
non-hub,kinless nodes	z < 2.5	P> 0.80
provincial hubs	z > 2.5	P< 0.30
connector hubs	z > 2.5	0.30 < P < 0.75
kinless hubs	z > 2.5	P> 0.75

Table 1: Role defenition. (from [ND13])

On the other hand, a coefficient of 0 indicates that the node is only linked to one community. Using these metrics, Guimera and Amaral define seven roles linked to the positioning of the two measures (see figure 3). The thresholds are defined in table 1.

This method offers an understanding of the roles within a network, however, it has some limitations. First, this method can only be applied to a non-directed network. Furthermore the threshold used to identify each role are fixed in an empirical way and are not guaranteed to be the best values. In order to fix these problems, Dugué and al have brought some modifications to the approach of Guimera and Alamara.

Oriented graphs Generally, to extend the concept of non-directed interactions to directed ones, we use the in-interactions and the out-interactions. Thus, we can redefine the metrics of the Guimera and Alamara method for the in, and out interaction. It is important to note that the method of community detection must be adapted to the directed network context. In other words, we have to use a method for community detection in directed networks. Moreover, using this directed network means that the threshold defined by Guimera and Alamara are no longer valid and thus, Dugué and Al suggest a non-supervised method to determine the roles. [GA05]

Non supervised role identification The limits of the Guimera and Alamara method is the presence of fixed roles regardless of the network. In addition, only the measure of partition is normalized, the limits set for the measures of z seem to have no logical explanation. In order to solve this problem, the automatic non-supervised classification method is used. [ND13]

4 Event and anomaly detection

Anomaly detection is the analysis of large quantities of data to identify items, events or observations which do not conform to an expected pattern. Anomaly detection is applicable in a variety of domains, e.g., fraud detection, fault detection, system health monitoring [ML15]. An entropy-based approach is suitable to detect anomalies in networks. There are two phases in this approach: training and detection. In the training phase profile of legitimate data is built and model for classification is prepared. In the detection phase, current observation are compared with the model. Initially, during the training phase, a dynamic profile is built using min and max entropy values within a sliding time window. In the detection phase, the observed entropy is compared with the min and max values stored in the profile according to the following rule: [PBS15]

$$r_\alpha(x_i) = \frac{H\alpha(x_i) - k * \min_\alpha}{k * (\max_\alpha - \min_\alpha)}$$

With the use of this rule, anomaly threshold is defined. Values $r_\alpha(x_i) < 0$ or $r_\alpha(x_i) > 1$ indicate abnormal concentration or dispersion. These abnormal dispersion or concentration for different feature distributions are characteristic for anomalies.
Detection is based on the relative value of entropy with respect to the distance between min and max. Coefficient k in the formula determines a margin for min and max boundaries and may be used for tuning purposes. A high value of k, e.g., $k = 2$, limits the number of false alarms (alarms where no anomaly has taken placed) while a low value ($k = 1$) increases the detection rate (the percentage of anomalies correctly detected). Some other approaches to thresholding based on standard deviation $-mean \pm 2sdev$, median absolute deviation $-median \pm 2mad$ has been also taken into consideration but empirical results proved that proposed rule is the best choice.

5 Pattern detection in a dialogue

5.1 Coding the triplet

By coding our triplets of interaction with a letter for each member, we can have a long serie of letters.

These three triplets $T_1{=}(t_1, u, v)$ $T_1{=}(t_2, w, u)$ $T_1{=}(t_3, v, x)$
will gives us a sequence of letters :
s = UVWUVX
which are the ids of each member in a row. We propose to study the Smith-waterman algorithm in order to find eventually some regularity in these interactions.

5.2 The Smith-Waterman Algorithm

The first use of this algorithm is to detect some regularity and local alignment in two DNA sequences.[Wik16] The two schemes following are taken from this source
We can apply this algorithm to our sequences of letters formed from the linkstream.[Ale14]

Assuming we have to character strings as s_A and s_B. There are two steps in this algorithm :

- The calculation of the matrix of the local alignment score. Each coefficient of the matrix T[i][j] gives the local alignment score of the i first characters of s_A, and the j first characters of s_B. This calculation is done iteratively with the shortest local alignment score. At the end, the coefficient in the low-right corner gives the global alignment score between s_A and s_B

- The construction of the alignment from the matrix T. Starting from the low-right corner, we ride up the matrix to define by which path, we have found the optimal score. This path correspond to the optimal local alignment.

There are three case :

- The best alignment ends with a connection between $s_A[i]$ and $s_B[j]$. In this case, we have $T[i][j1] = T[i-1][j-1] + Distance(s_A[i], s_B[j])$
- The best alignment ends with a connection between $s_B[j]$ and a gap in the sequence s_A $T[i][j1] = T[i][j-1] + \Delta$
- The best alignment ends with a connection between $s_A[j]$ and a gap in the sequence s_B $T[i][j1] = T[i-1][j] + \Delta$

The recurrence formula is

$$T[i][j] = max \begin{cases} T[i][j-1] + \Delta \\ T[i-1][j-1] + Distance(s_A[i], s_B[j]) \\ T[i-1][j] + \Delta \\ 0 \end{cases}$$

with Δ the penalty associated with a gap and $Distance(s_A[i], s_B[j])$ a distance we have to define in a similarity matrix. The value 0 permits to have only positive values in the matrix T.

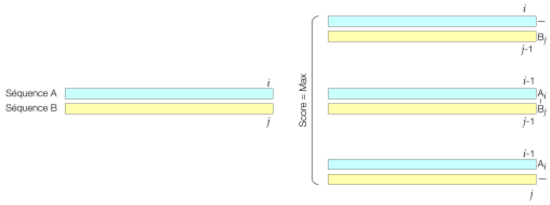
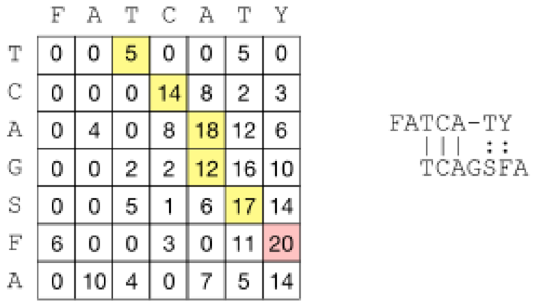


Figure 4: Illustration of the matrix calculation [Wik16]

Assuming the matrix T is constructed, we are looking for the local maximum in the matrix T, above a thresold value set up at the beginning. This values corresponds to the ends of the local alignment. Then we will ride up the matrix from this position and determine all the edition options taken by decrypting the path.



	F	A	T	C	A	T	Y
T	0	0	5	0	0	5	0
C	0	0	0	14	8	2	3
A	0	4	0	8	18	12	6
G	0	0	2	2	12	16	10
S	0	0	5	1	6	17	14
F	6	0	0	3	0	11	20
A	0	10	4	0	7	5	14

```
FATCA-TY
  |||  ::
  TCAGSFA
```

Figure 5: Illustration of the matrix ride up [Wik16]

6 Implementation

6.1 Entropy

Each of the entropy definitions described above has its way to characterize a Linkstream by the different measures that can be calculated and interpret based on the way they are computed. Nevertheless, not any of these Entropy definitions can be implemented practically or tested on a dataset in the real world due to either its difficulty to comprehend by our project team or the lack of a clear vision on the utility that can add to the analysis process of the online communities dynamics. This could be, however, a potential way to explore in the future researches in order to understand more the way the entropy works.

Our objective was always to find a way we can help a community manager to better understand an online community and to analyze its dynamics so he/she can react the right way toward any event or change among this online society.

The art state gave us a vision on the different types of entropy and the way they can be implemented first in a Static Graph. However, we have chosen to work with the Degree Distribution based Entropy, and leave the others for further researches, and the transition from a static graph to a linkstream was one of the reasons for this decision. In fact, we believe that the Degree Distribution based Entropy is more likely to be interpretable in the case of a linkstream modeling.

Thereupon, we basically worked on the implementation of this entropy in Python and on finding a suitable interpretation of this measure through different datasets (Linkstreams).

Actually, our work is based on the work of Patipol CHIAMMUNCHIT who succeed to implement the linkstream modelisation in python. His work helped us to manipulate a linkstream object instead of a simple numpy matrix and to use some other measures related to this type of objects.

Moreover, we also have used in the development process of our code other libraries like pandas and numpy for managing the datasets and manipulating the data frames, math for providing access to the mathematical functions, and matplotlib for generating plots, histograms, scatter plots, ...etc.

6.2 Results

As the entropy is defined as a measure for the linkstream modeling, it should be used to characterize a linkstream from another and to give a better understanding of the structure of an on line community and the events that occurs over time. That's why we implemented the entropy this way so a community manager could manage different communities and compare their characteristics or to compare them with a reference case. For instance, we have tested, experimentally, the variation of a linkstream entropy over many parameters which are considered as vital, like the population size. In fact, the figure (Fig.6) below shows the effect of changing population size on the entropy measure in a way that the entropy decreases after the population size exceed the number of links (no clusters were considered) which was fixed for this example on 30 links.

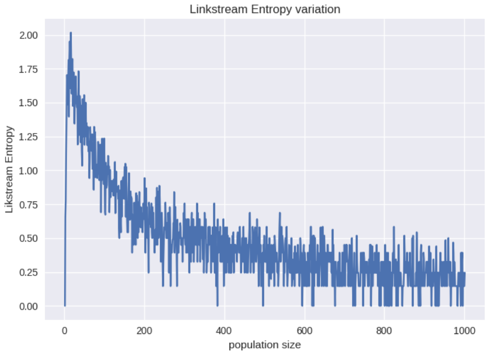


Figure 6: Linkstream Entropy variation against population size

The entropy in this case gives an idea on how much the community is "active".
Similarly, the entropy increases with the number of links, when the population size is fixed (50 nodes in this example, Fig.7):

However, the mean delta-entropy doesn't behave the same way :

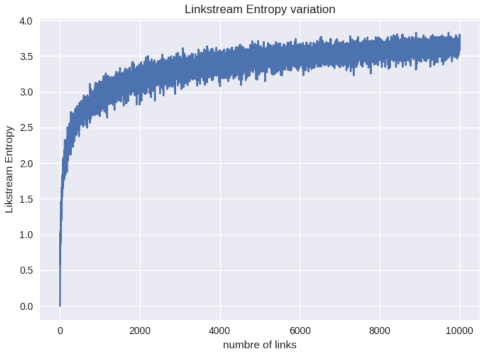


Figure 7: Linkstream Entropy variation against number of links

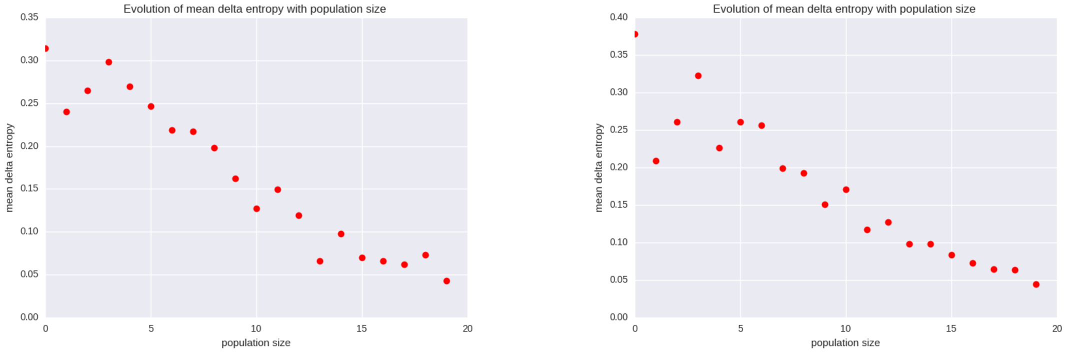


Figure 8: Mean-delta Entropy variation: Number of total links (1) static, (2) dynamic

Clustering Factor: After several test changing clusters over and over, we have concluded to a theory of the distribution of mean delta-entropy. In fact, it changes with the number of clusters in the community. Actually, as the figures below show, in case of a community with no clusters the mean delta-entropy changes with a logarithmic scale with delta. However, this evolution is considerably perturbed when the community has 2 or more clusters. This result could be an indicator, for the community manager, of the existence or not of clusters within an on line community.

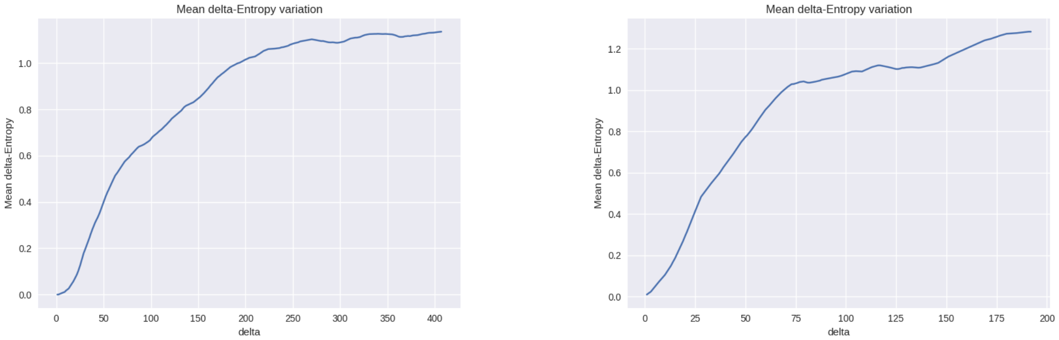


Figure 9: Mean-delta Entropy variation: (1) 0 cluster, (2) 2 clusters

Same tests were run on the entropy of the whole linkstream, but we could not achieve any improvements. Nevertheless, we were curious to see why the mean delta-entropy respond to clustering better than the whole linkstream entropy. Thus, we have done some tests to observe the difference :

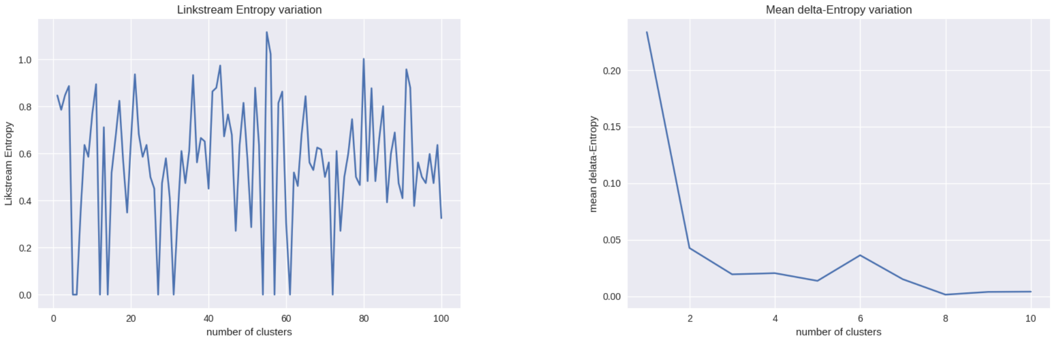


Figure 10: (1) Entropy variation, (2) Mean-delta Entropy variation

Indeed, the entropy and the mean delta-entropy behave differently in a way we didn't success to figure it out.

6.3 Perspective and further work

Since we have invested the majority of our project time on figuring out what entropy should we implement and the rest of our time on the implementation, we didn't test the entropy enough so we can come out in conclusions already. Actually, even our choice of entropy is still questionable, since the other entropies are as interesting as the one we implemented. In fact, implementing other entropies and comparing their results to ours could be a way to explore. However, our implementation is ready to be tested and explored in addition to the several tests we have made so far. Also, we have worked so far with only small data generated by the functions we have created. We believe that it could be interesting to test the entropy on large datasets and large time lines and to evaluate the performance of our code.

7 Conclusions

As we have exhibits in this overview of methods and concepts that help understanding online communities' dynamics, many solutions have been proposed. Indeed, these methods could help a community manager through his daily work to better manage an online community and to understand its dynamics in order to achieve a high level of QoS.

Due to the absence of a precise definition of what a good community is, many studies reaching highly divergent results have been equally successful. All these works are interesting, and often give complementary visions.

However, we tried in this paper to regroup the most interesting works of studying the dynamics of a community and tried to project the results on a Linkstream model. Three aspects were treated mainly, for a community modeled as a Linkstream, in order to follow up its evolution and the different events that could happen. First, we treated some KPIs that help describing online communities and give an overview of its components. Second, we mentioned some methods of detecting events and anomalies based on an entropic approach and some methods of identifying roles within a community modeled as a Linkstream. Finally, we explored some methods of patterns detection.

Nevertheless, there are many interesting theoretical and applied questions that remain open to further study and also a lot of works that could be beneficial to continue this research.

References

[Ale14] Zacharie Ales. Extraction et partitionnement pour la recherche de régularités : application à l'analyse de dialogues. mathématiques générales [math.gm]. insa de rouen. 2014.

[BL07] Jean-Louis Ermine Benoit LeBlanc. A shannon's entropy of knowledge. 2007.

[GA05] Roger Guimerà and Luís A Nunes Amaral. Cartography of complex networks: modules and universal roles. 2005.

[KS14] Václav Snášel Khalid Saeed. Computer information systems and industrial management: 13th ifip tc 82. 2014.

[LD04] Thomas Manke Lloyd Demetrius. Robustness and network evolution—an entropic principle. 2004.

[ML15] Clemence Magnien Matthieu Latapy, Assia Hamzaoui. Detecting events in the dynamics of ego-centered measurements of the internet topology. 2015.

[ND13] Anthony Perez Nicolas Dugu´e, Vincent Labatut. Identification de rôles communautaires dans des réseaux orientés appliquée à twitter. 2013.

[NGL] Raphaël Fournier-S'niehotta Qinna Wang Noé Gaumont, Tiphaine Viard and Matthieu Latapy. Analysis of the temporal and structural features of threads in a mailing-list.

[PBS15] Bartosz Jasiul Przemysław Berezinski and Marcin Szpyrka. An entropy-based network anomaly detection method. 2015.

[SA04] Harith Alani Sofia Angeletou, Matthew Rowe. Modelling and analysis of user behaviour in online communities. 2004.

[Sat14] Arun Sathanur. An activity-based information-theoretic annotation of social graphs. 2014.

[SV04] Ricard V. Solé and Sergi Valverde. Information theory of complex networks: on evolution and architectural constraints. 2004.

[TV16] Clemence Magnien Tiphaine Viard, Matthieu Latapy. Computing maximal cliques in link streams. 2016.

[Wik16] Wikipedia. Algorithme de smith-walterman. 2016.