You are the way you structurally talk: structural-temporal neighbourhoods of posts to characterize users in online forums

(Draft version: April 25, 2016)

Alberto Lumbreras
Julien Velcin
Laboratoire ERIC
Université de Lyon, France
alberto.lumbreras@univ-lyon2.fr
julien.velcin@univ-lyon2.fr

Bertrand Jouve
FRAMESPA/IMT
Université de Toulouse 2, France
jouve@univ-tlse2.fr

Marie Guégan
Technicolor, France
marie.guegan@technicolor.com

***Abstract*—Users of social networks are often characterised by extracting some relevant features from the graphs associated to that network. The structural dynamic of the conversation are usually forgotten due to a lack of proper tools. We present a purely graph-based method to cluster users in online forums.**

I. Introduction

The interactions between users in online forums are often modelled as complex networks. As such, they can be studied from many levels, each of which is represented by a graph where vertices and edges may represent different things. The most studied graph is a graph where vertices represent users and edges represent interactions between users (a post from one user replying to a post from other user). From the point of view of the community, some typically analysed properties are the degree distribution, the clustering coefficient, density, or diameter of the graph. From the point of view of the individual users, analyses are typically focused either on the centrality of users, on the community structure or in blockmodeling (groups of users that tend to interact with the same other groups) [1]. Another common graph is a tree graph where vertices represent posts (or e-mails) and and edges from one vertex to another indicates that the first is a reply to the second. Given the different discussion trees of a forum (a forest) the global analyses include the depth distribution, branching factors, or even the time between two posts [2]. representation include depth distribution of discussions, branching factors and so. Studying discussion trees from the point of view of the users means, since trees are discussions, studying the user at a conversational level. Unfortunately, there is a lack of structural tools to perform this analysis. In this paper, we propose to fill this gap through the concept of *structural-temporal neighbourhood.*

Our goal in this paper is two-fold: on the one hand, to illustrate how structural-temporal neighbourhoods can play the role of triads for conversation trees, in the sense that they show us the local structures or dynamics from which the bigger graph emerges. On the other hand, to show how structural-temporal neighbourhoods can be used for the detection of different types of conversationalists in online forums or any other type of online discussion that is representable by a tree structure (e.g.: e-mails).

The remaining of the paper is as follows. We first discuss about the convenience of the classic neighbourhood definition in dynamic graphs such as discussion trees. Then we introduce our two definitions of structural-temporal neighbourhoods. To illustrate the kind of neighbourhoods that we obtain in a real data, we analyse the neighbourhood census of a Reddit forum. Finally, we apply both neighbourhood definitions to detect clusters of users that tend to appear in the same neighbourhoods. We close the paper with some suggestions of future research.

II. Discussion trees

We represent a discussion thread by a tree graph $G = (V, E)$ where V is a set of n vertices representing the posts (also called messages or comments) and E is a set of $n-1$ directed edges that say which posts replied to which post. Vertices have two attributes namely time and author. If for two given vertices v_i, v_j there exists an edge $e = (v_i, v_j) \in E$ then v_j is called the *parent* of v_i, denoted as $p(v_i) = v_j$. The *root* of the tree is the only vertex with no parent, and corresponds to the post that starts the discussion. We say that two vertices v_i, v_j are *siblings* if and only if $p(v_i) = p(v_j)$. A vertex v_i is an ancestor of a vertex v_j and v_j is descendant of v_i if and only if v_i is in the path from v_j to the root. A *leaf* is a post with no replies. A *branch* is the path between a leaf and the root. Two vertices v_i and v_i are said to be *neighbours* if either $p(v_i) = v_j$ or $p(v_j) = v_i$ or, in other words, if their distance in the undirected version of G is one.

III. Structural neighbourhoods

Extending the classic definition of neighbourhood according to which two vertices are neighbours if the distance between them is one, we start by the following definition of *structural neighbourhood*:

Definition 1: Given a tree graph G, the *structural neighbourhood* of radius r of post i, denoted as $\mathcal{N}_i(r)$, is the induced graph composed of all the vertices that are at distance equal or less than r from post i.

In the context of discussion threads, this definition has two limitations. First, the decision on whether to include some post in the neighbourhood is only based on the structural distance, and therefore two posts that are at distance $d \leq r$ are considered neighbours of i regardless of the time when they were written. In conversations, time plays an important role, therefore this is an important limitation. Another consequence of looking only at the structure is that the number of possible neighbourhoods within a radius r is infinite. This poses a problem when trying to categorize conversations since many conversations, while structurally different, can be considered semantically equivalent.

IV. Structural-temporal neighbourhoods

As we have discussed, structural neighbourhood falls short in the analysis of conversational structures. Instead, we propose two new definitions of neighbourhood that take into account the order and time in which posts are written.

A. Order-based

Our first definition is based on the order in which posts are attached to the tree.

Definition 2: Given a ordered tree graph G, the *order-based neighbourhood* of radius r of vertex i, denoted as $\mathcal{N}_i^T(r,n)$ is the induced subgraph from its structural neighbourhood composed the n vertices that are closest to i in time and for which there exists a path to i in $\mathcal{N}_i^T(r,n)$.

An example of order-based neighbourhood is given in Figure 1 (left). This definition has two advantages over the *structural neighbourhood*. First, the temporal aspect of the conversation is better taken into account since the neighbourhood only includes posts that are not only near to i in the structure but also in time. Second, the size of the neighbourhood has an upper bound of $min(|\mathcal{N}_i(r))|, n)$ thus making the space of possible neighbourhood structures finite. The main limitation of this definition is that the obtained neighbourhoods might be very different with different choices of r and n, and we have no *a priori* criteria to choose the proper parameters other than making them small so that the neighbourhoods capture the local dynamic of the conversation around the post i.

B. Time-based

In a first attempt to take time into account, one might set fixed time-based boundaries for the neighbourhood and include only those posts whose timestamp t_j is at distance less than τ from the ego post i, $|t_j - t_i| < \tau$. However, the pace at which posts are added to the conversation may be very different between conversation threads (and also within a thread) and we have no *a priori* criteria for a proper choice of τ.

Rather than looking for a fixed time radius, we may look at changes in the pace how new posts are added to the thread. In statistical analysis, a *changepoint* in a sequence $x_1, ...x_n$ is a point that comes from a different probability distribution than its precedent values. If sequences are timestamps, they are monotonic increasing, therefore the changepoints will correspond to sudden pauses in the conversation. If applied either to the timestamps of posts in a branch (vertical sequence) or to the timestamps of the replies to the same post (horizontal sequence), we say that a sequence posts with timestamps $t_1, ..., t_n$ belong to the same (vertical or horizontal) *local dynamic* if there is no changepoint t_i in the sequence such that $1 < i \leq n$. For the detection of changepoints we use the PELT algorithm [3][1]. Now we can introduce our second definition of neighbourhood.

Definition 3: Given a tree graph G, the *time-based neighbourhood* of vertex i, denoted as $\mathcal{N}_i^T(r)$, is the maximal subgraph of the structural neighbourhood $\mathcal{N}_i(r)$ where all the vertices belong to the same vertical and horizontal local dynamic than i.

Note that we still have a radius parameter r to guarantee that the resulting structure remains local even if no changepoint is detected near the post i.

Algorithm 1 extracts time-based neighbourhoods of a given post according to the given definition. Since the breakpoints only depend on the tree and not on the particular post we analyse, we previously detect the horizontal and vertical breakpoints in the tree. In the case of multiple branches with some common posts, we consider that a common post is a vertical breakpoint if it is a vertical breakpoint in any of the branches. Once we have the breakpoints we can proceed with the algorithm. First, we extract the structural-neighbourhood. The time-based neighbourhood will be a subset of the later. Then, we look for horizontal and vertical breakpoints, which mark the frontiers of the time-based neighbourhood. There are four possible cases:

- A *vertical breakpoint in the ancestors*: If the breakpoint is in the path between the post and the root, then the breakpoint started the new local dynamic to which the ego post belongs. Thus, we remove the ancestors of the breakpoint, but not the breakpoint.
- A *vertical breakpoint in the descendants*: If the breakpoint is a descendant then it started a new different dynamic and therefore we must remove the descendants of the breakpoint and the breakpoint itself.
- An *horizontal breakpoint either in the older siblings or in the ancestors*: If the breakpoint is among the older siblings, then it started the horizontal dynamic to which the ego post belongs. Similarly, if the breakpoint is among the ancestors, then every older sibling of the breakpoint belongs to a previous local dynamic. In both cases, we remove the older siblings of the breakpoint but not the breakpoint.

[1] An implementation of this algorithm written by its own authors is available in the R library `changepoint`.

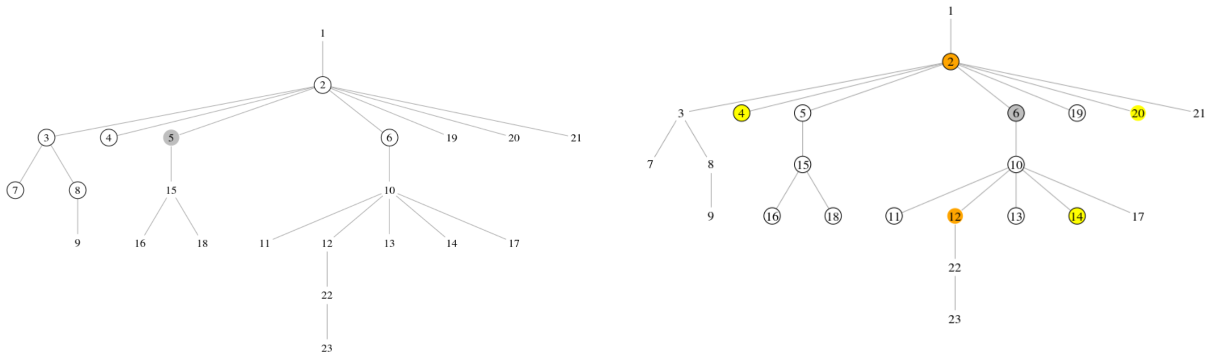

Fig. 1. Illustration of order-based (left) and time-based (right) neighbourhoods. Grey nodes represent the ego post. Nodes with circles are those in the neighbourhood. Numbers indicate the order of the post. The order-based neighbourhood has parameters $r = 3$, $n = 6$. In the time-based neighbourhood, horizontal changepoints (yellow) and vertical changepoints (orange) represent posts that are temporally far from their predecessors (siblings or parents) and therefore set the limits of the neighbourhood.

- An *horizontal breakpoint elsewhere*: In any other case, the horizontal breakpoint starts a different local dynamic and therefore we remove its younger siblings and the breakpoint.

Any time a vertex is remove we also remove its descendants. The final neighbourhood will be connected induced subgraph from the structural neighbourhood. If there are no changepoints in the structural neighbourhood, the time-base neighbourhood will be exactly the structural neighbourhood.

An example of time-based neighbourhood is given in Figure 1 (right).

Algorithm 1 Extraction of time-based neighbourhood

```
Input: Posts tree $g$, vertical breakpoints, horizontal breakpoints, ego post $ego$
Output: Subgraph of $g$ with all vertices in $V(g)$
  Compute structural neighbourhood $\mathcal{N}_i(r)$
  ancestors $\leftarrow$ ancestors(ego) in $\mathcal{N}_i(r)$
  older_siblings $\leftarrow$ older_siblings(ego) in $\mathcal{N}_i(r)$
  dump $\leftarrow \varnothing$
  for bp $\in$ vertical breakpoints do
    if bp $\in$ ancestors then
      dump $\leftarrow$ dump $\cup$ ancestors(bp)
    else
      dump $\leftarrow$ dump $\cup$ descendants(bp) $\cup$ bp
    end if
  end for
  for bp $\in$ horizontal breakpoints do
    if bp $\in$ (older_siblings $\cup$ ancestors) then
      dump $\leftarrow$ dump $\cup$ older_siblings(bp)
    else
      dump $\leftarrow$ dump $\cup$ younger_siblings(bp) $\cup$ bp
    end if
  end for
  $\mathcal{N}_i^{(t)}(r) \leftarrow$ delete(dump_posts $\cup$ descendants(dump_posts)) from $\mathcal{N}_i(r)$
```

V. Neighbourhood colouring and pruning

A. Colouring

Even if the structure contains some important information about the type of conversation, there is still some ambiguity left, and two similar neighbourhoods can represent very different types of conversation. We can easily reduce this ambiguity by assigning colors, or labels, to vertices, identifying some relevant property of the post.

In particular, we assign the following colours:

- *Red*: ego post.
- *Orange*: other posts written by the author of ego. It allows to identify re-entries (the same author participating more times in the discussion [4])
- *Yellow*: parent of ego post and other posts written by the same author. It allows to identify, for instance, debates between the ego author (red and orange) and the other author (yellow). We have observed this phenomena in our data, and it is often the cause of long chains.
- *White*: root post. Differentiating the root post from the rest has been proven by previous research on online discussions. For instance, some types of users seem to get more replies than others when they initiate a thread ([5], [6]). Also, in terms of preferential attachment, root posts use to get more replies than the non-roots [7], [8].
- *Black*: none of the above.

Since some posts might correspond to several colors, we perform a sequential assignment of colors given by `black, orange, red, yellow, white`.

We do not claim this choice of labels to be of universal. Indeed, other labellings might also give interesting results since they would look at conversations from new points of view (e.g.: a colour for leaf vertices, colours according to the length of the post, or colours to represent the sentiment of the post).

B. *Pruning*

At this point, we can still find structures whose only difference is that one of them has more leafs hanging from some of the nodes. Thus, we prune every neighbourhood by leaving a maximum of two consecutive siblings of the same color and where neither of them has any children. The reason of setting the limit to two is that it is the minimum necessary to distinguish between *zero*, *one* and *more than one* consecutive occurrences. In other words, we consider that the difference between one and zero replies to a post is relevant, but that five and six replies do not make any difference worth being represented.

Figure 2 shows some real examples, from our dataset, of time-based neighbourhoods after colouring and pruning.

VI. Application to Reddit forums

Our data consists of two different forums of Reddit, namely the Podemos dataset [2] and the Game of Thrones dataset[3]. The Podemos forum was conceived in March 2014 as a tool for internal democracy, and forum members used it to debate ideological and organizational principles that were later formalized in their first party congress hold in Madrid the October 18th and 19th 2014. Nowadays, its members use it mainly to share and discuss about political news. The Game of Thrones is a casual discussion forum about the Game of Thrones TV series. The Podemos dataset contains 75,000 posts corresponding to 4,262 threads written by 9,160 users between the April 25th and October 20th 2014. The Game of Thrones dataset contains 75,000 posts corresponding to 5,862 threads written by 18,907 users.

A. *Neighbourhood extraction*

As a previous step for all further analysis we detect the neighbourhood around every post in our datasets. Each neighbourhood structure (including colours) is given a unique integer number. Unfortunately, it is not practical to give a number to every theoretically possible structure since, even if we could enumerate all the possible trees up to a given size, a lot of of these trees would not be seen in the data. It appears then a much more practical approach to enumerate only those trees that are seen in the data. The consequence of this is that every dataset will have its own dictionary of trees.

We detect the order-based and time-based neighbourhood of every post as follows. A counter for every neighbourhood structure is created the first time it us detected. For every post, we extract its neighbourhood (order-based or time-based) colour and prune it as explained above (Section V). Then we check whether it is isomorphic to some of the already seen neighbourhoods. If this is the case, then we increment the counter for the neighbourhood previously seen. Otherwise we create a new entry for the current neighbourhood. The computational cost of this operation has a upper bound of $\mathcal{O}(n^2)$, n being the number of posts, for the case where each neighbourhood occurs only once. [4]

B. *Agreement between neighbourhood methods*

The detection of the time-based and order-based neighbourhood in the Game of Thrones dataset gave us a list of 2000 different (non-isomorphic) time-based neighbourhoods and 180 different (non-isomorphic) [5] order-based neighbourhoods, which means that the time-based is able to capture a much wider set of structures. Of course, this is not necessarily and advantage since these extra structures may be just spurious and only occur a very small number of times. Indeed, Figure 4 shows that around half of the time-based neighbourhoods occur only once, and that we only need 384 time-based to cover 95% of the census, while for the order-based, 53 structures are enough to cover the same percentage. Moreover, there is no much difference, in terms of frequency, between the 50 most frequent neighbours of each type. The reason why the frequency of the order-based neighbourhoods decay faster might be that the most frequent neighbourhood is much more frequent (upper-left red point in the left plot of figure 4) than the most frequent time-based neighbourhood.

The divergent number of neighbourhoods between order-based and times-based brings us to the following question: how differ the order-based and time-based neighbourhoods given to a same post? We re-labelled the order-based neighbourhoods and assigned them the label of their isomorphic time-based

[2] https://www.reddit.com/r/podemos

[3] https://www.reddit.com/r/gameofthrones

[4] TODO: It's actually much less, probably logarithmic, since the size dictionary grows incrementally

[5] TODO: Review numbers

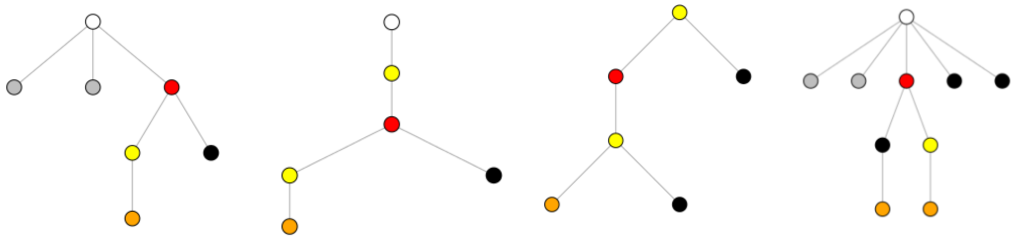

Fig. 2. Illustration of different real structures using different colors for the root (white), the ego (red) and other posts written by the same author (orange) the ego parent and other posts written by the same author (yellow), posts previous to the ego (grey) and posts posterior to the ego.

counterparts or the first unused number in the time-based index. This left us a set of 2183 trees. Figure 3 shows, for every post, its time-based neighbourhood and its order-based neighbourhood. Those in the diagonal (34%) see no difference between the time-based and order-based. Out of the diagonal, there are two rows indicating a high number of posts being assign the same order-based neighbourhood but very different time-based neighbourhoods. These order-based neighbourhoods are the number 1 and the number 50. These collapse of many time-based neighbourhoods into one is significant because it happens a lot among very frequent time-based neighbourhood. In the right plot of Figure 3 we can see other order-based neighbourhoods that collapse many of the most frequent time-based neighbourhoods, such as the 22, 17, 26, 29, 49, 56, 47 and 101.

C. Neighbourhood census

Given a set of local structures and a graph under study, a census is a counting of how many times does each structure appear in the graph. In the context of online discussion forums, triad census has been sometimes used in the graph of users interactions are often used (in these graphs, an edge exists from user u to user v if u wrote a post reply to some of the posts written by v in some thread.) [9], [6]. However, triads are poor objects when studying trees since there are only three possible structures with, at most, three nodes: a single vertex, a directed dyad (reply), star (two replies to the same post) and a chain (reply and reply to the reply). On the contrary, our neighbourhoods are potentially richer structures that capture the real structure of conversations.

In Figure 5 we compare the neighbourhood census in the Podemos and the Game of Thrones forums. Interestingly, two time-based neighbourhoods are more than three times likelier in the Game of Thrones. These neighbourhoods correspond to a discussion between two users (21) and to XXXXXX (25). The same comparison for the order-based triad is harder to interpret, since the outstanding neighbourhoods in the Game of Thrones correspond to XXXX (25) and ... (27)

This results lead us to think that the time-based neighbourhood is more convenient than the order-based. The main reason, we think, is that order seems to be a too dogmatic way to decide where a neighbourhood ends.

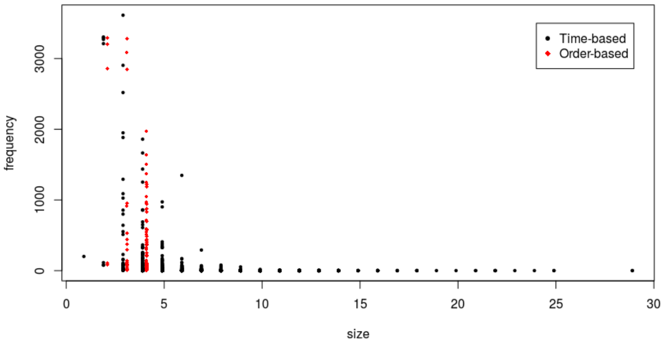

Fig. 6. Size of neighbourhoods and their frequencies.

D. Conversation-based clustering of users

In this section, we cluster users based on the neighbourhoods of their posts. Our intuition is that some users like participating in some kind of discussion rather than other. Certainly, most part of the information necessary to understand the nature of a discussion in in its textual content. However, due to the huge diversity of topics, vocabulary, and the difficulty of current algorithms to capture the language subtleties such as humour, irony, or context, we turn our attention towards the structure of the discussions, which can also contain some information. We work under the hypothesis that the structural-neighbourhoods in which a user post in embedded reflect the kind of conversation in that part of the thread.

We limit our study to a set of 100 *active users* who wrote more than 100 posts. For those users, we create an initial feature matrix $U \times N$ where U is the number of users and N is the number of neighbourhoods in the census, and where the the position (u, n) is a counter of the number of times that a post written by user u has a neighbourhood isomorphic to n. We drop those feature columns that are zero for every user (these are neighbourhoods seen only around posts of users with low-level activity). To make the feature vector of a user independent on the number of posts, we transform the counts into percentages. And since some features have much higher percentages than others for most users, we scale and normalize the matrix so that every feature has mean 0 and variance 1. To avoid non-significant scores, we remove also

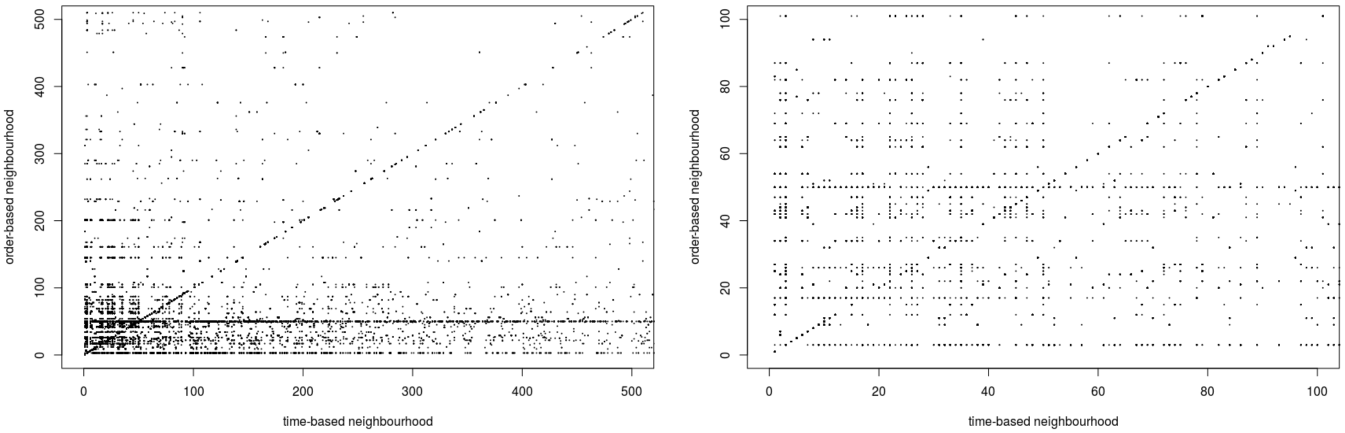

Fig. 3. Neighbourhood of posts from the point of view of order-based and time-based. Neighbourhoods with the same label in the order-based and the time-based axes are isomorphic. The points in the diagonal correspond to posts to whom the two methods assign exactly the same neighbourhood. Horizontal rows of points correspond to posts that are assigned many different time-based neighbourhoods but the same single order-based neighbourhood.

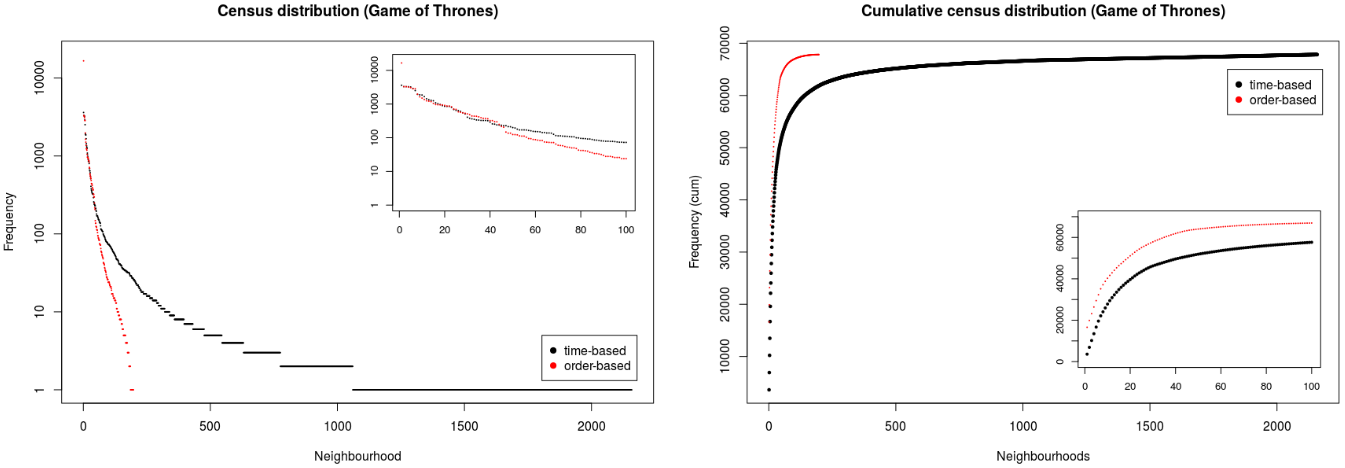

Fig. 4. Comparison of census distributions with order-based and time-based neighbourhoods.

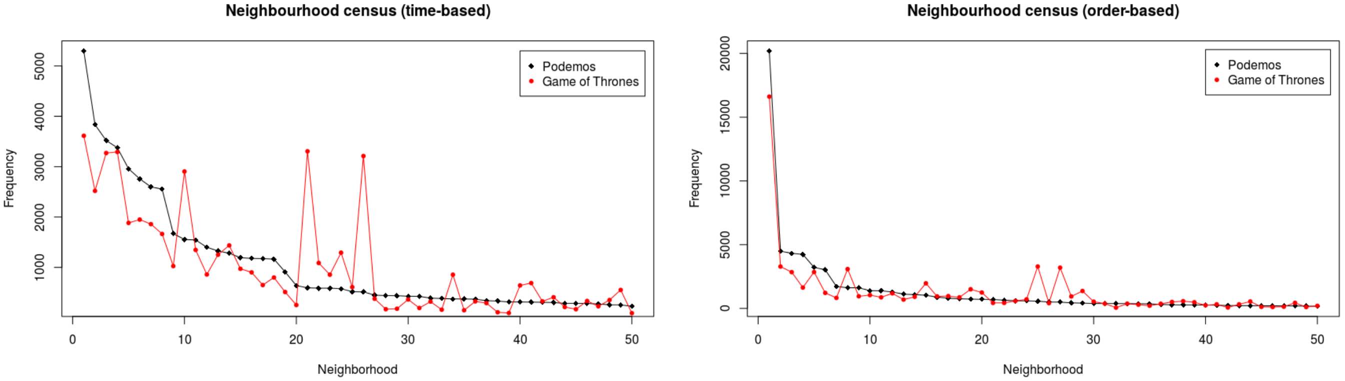

Fig. 5. Comparison of census for two different forums.

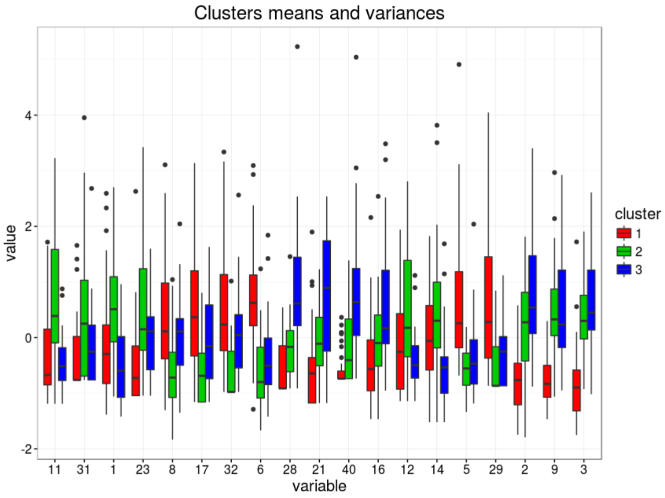

Fig. 7. Whiskers plots in the most relevant features (time-based)

the feature column corresponding to neighbourhoods that have a frequency less than 50 among the active users.

We use k-means to find the clusters, though one can use any other clustering method. Since the plot over the Within-Cluster Sum of Squares for $k = 1, ...20$ did not show any clear elbow we chose $k = 3$ clusters so that clusters are easier to interpret. We did the clustering over a feature matrix with order-based neighbourhoods and another feature matrix with time-based neighbourhoods. Figure 8 and shows the PCA projections of the users coloured by their assigned cluster.

VII. Conclusions

We presented a method to characterise conversations of users in online threads. Due to the tree nature of online threads, traditional patters such as triads are not able to capture much of relevant dynamics of a conversation. Our defined order-based and temporal-based neighbourhoods are able to capture a very rich variety of structures. We used this neighbourhoods to characterise users in terms of the structure of the conversations they participate in and showed that, indeed, there are different types of structural conversationalists

The concept of structural-temporal neighbourhood opens the door to some interesting paths of research. One might wonder whether other pruning are more pertinent than the proposed here. Also, even after pruning some neighbourhoods suggest the same type of conversation, so a manual merge might be convenient. Maybe sociologists can help to merge the found neighbourhoods in a meaningful way.

References

[1] A. McCallum, X. Wang, and A. Corrada-Emmanuel, "Topic and role discovery in social networks with experiments on enron and academic email," *Journal of Artificial Intelligence Research*, vol. 30, no. 1, pp. 249–272, 2007. [Online]. Available: http://www.aaai.org/Papers/JAIR/Vol30/JAIR-3007.pdf

[2] R. Bhatt and K. Barman, "Global Dynamics of Online Group Conversations," in *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[3] R. Killick, P. Fearnhead, and I. Eckely, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, pp. 1590–1598, 2012.

[4] L. Backstrom, J. Kleinberg, L. Lee, and C. Danescu-Niculescu-Mizil, "Characterizing and Curating Conversation Threads: Expansion, Focus, Volume, Re-entry," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, 2013, pp. 13–22.

[5] I. Himelboim, E. Gleave, and M. Smith, "Discussion catalysts in online political discussions: Content importers and conversation starters," *Journal of Computer-Mediated Communication*, vol. 14, no. 4, pp. 771–789, jul 2009.

[6] A. Lumbreras, J. Lanagan, J. Velcin, and B. Jouve, "Analyse des rôles dans les communautés virtuelles : définitions et premières expérimentations sur IMDb," in *Modèles et Analyses Réseau : Approches Mathématiques et Informatiques (MARAMI)*, 2013, pp. 1–12.

[7] V. Gómez, H. J. Kappen, and A. Kaltenbrunner, "Modeling the structure and evolution of discussion cascades," in *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, 2010, pp. 181–190.

[8] V. Gómez, H. J. Kappen, N. Litvak, and A. Kaltenbrunner, "A likelihood-based framework for the analysis of discussion threads," *World Wide Web*, p. 31, apr 2012.

[9] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, "Knowledge sharing and yahoo answers," in *Proceeding of the 17th international conference on World Wide Web - WWW '08*. New York, New York, USA: ACM Press, 2008, p. 665. [Online]. Available: http://dl.acm.org/citation.cfm?id=1367587 http://portal.acm.org/citation.cfm?doid=1367497.1367587

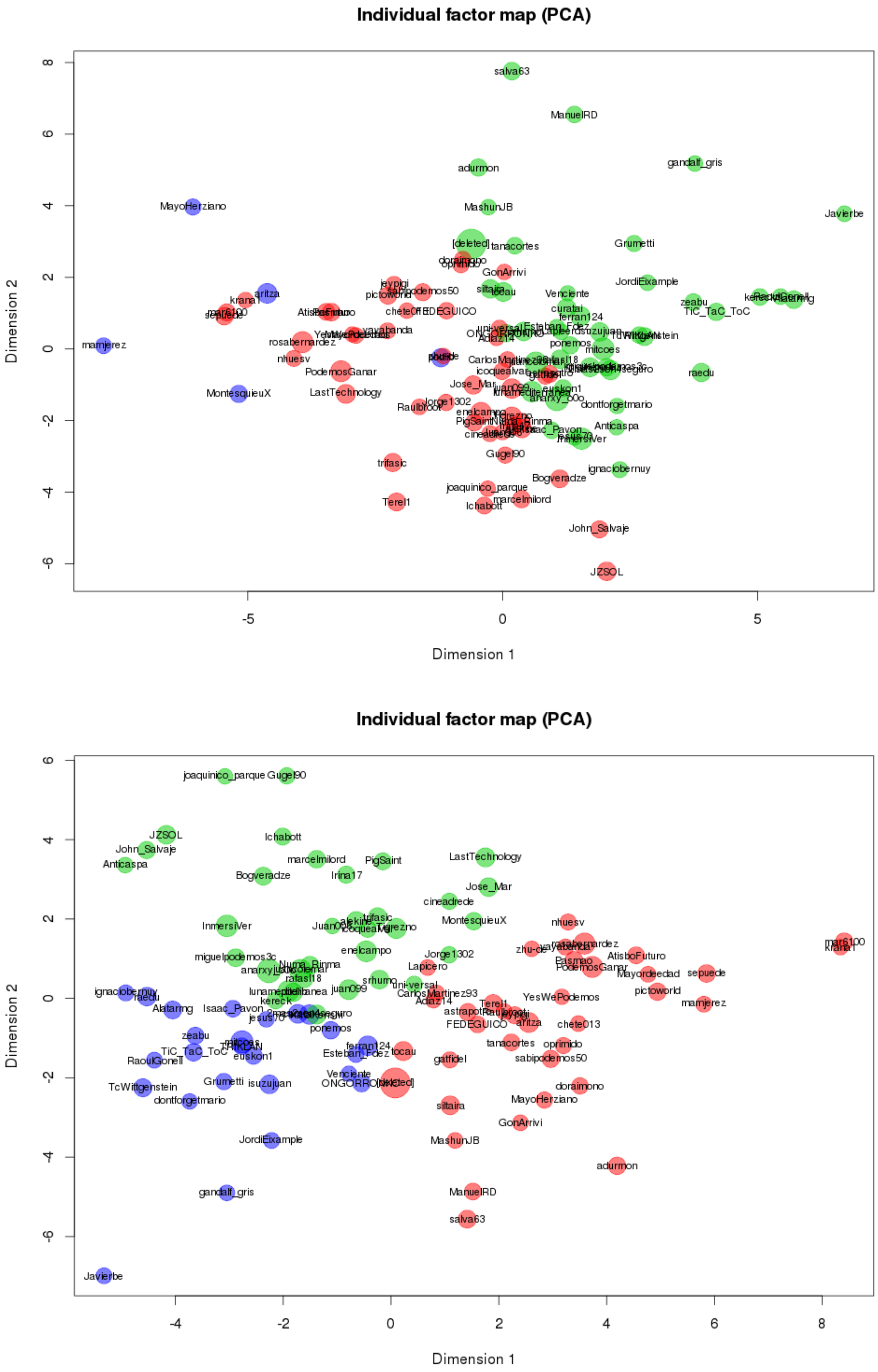

Fig. 8. PCA projections of the order-based (above) and time-based (below) neighbourhood features

APPENDIX

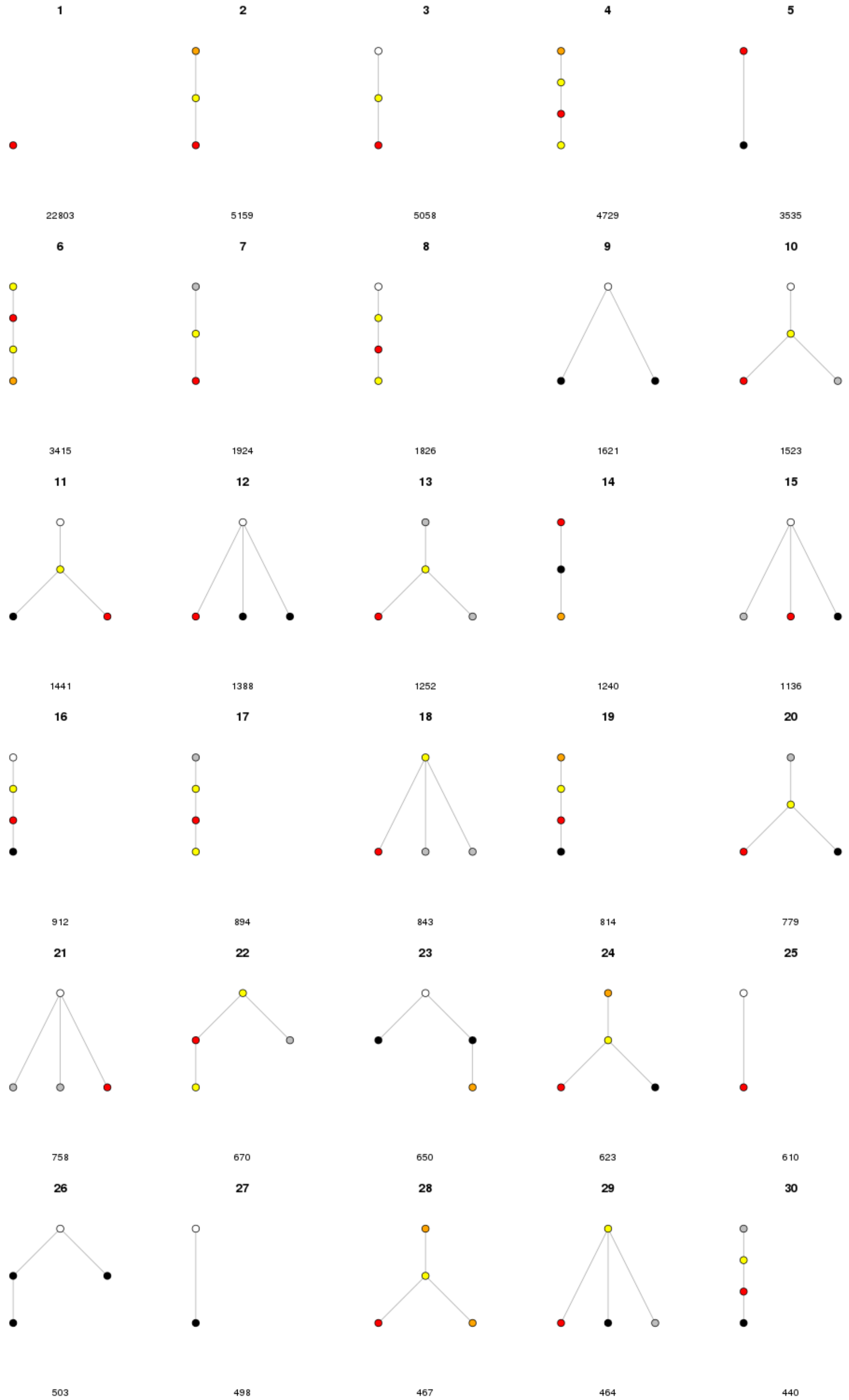

Fig. 9. Most frequent order-based neighbourhoods with $r = 2$ and $n = 4$ (Podemos)

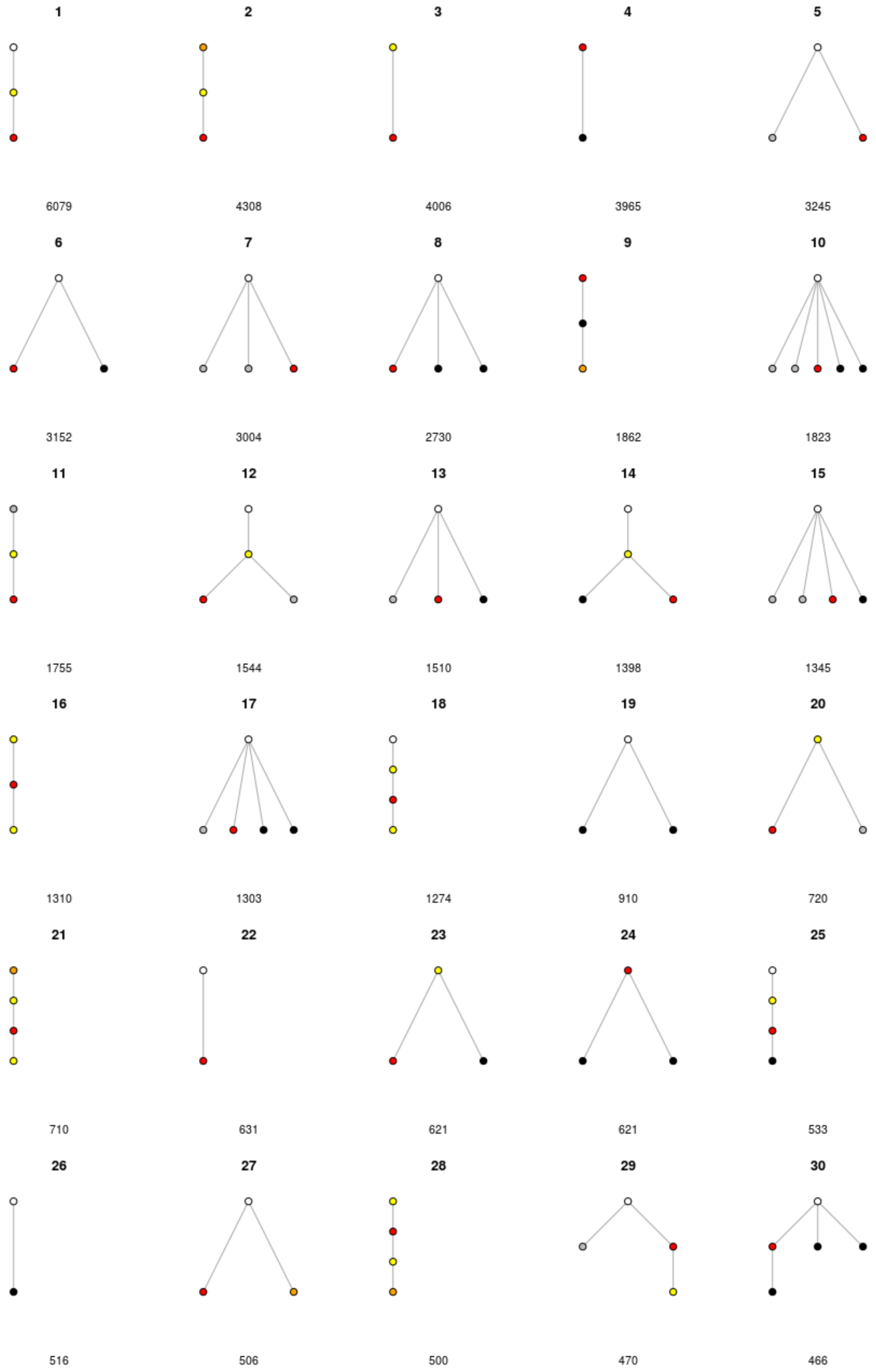

Fig. 10. Most frequent time-based neighbourhoods (Podemos)

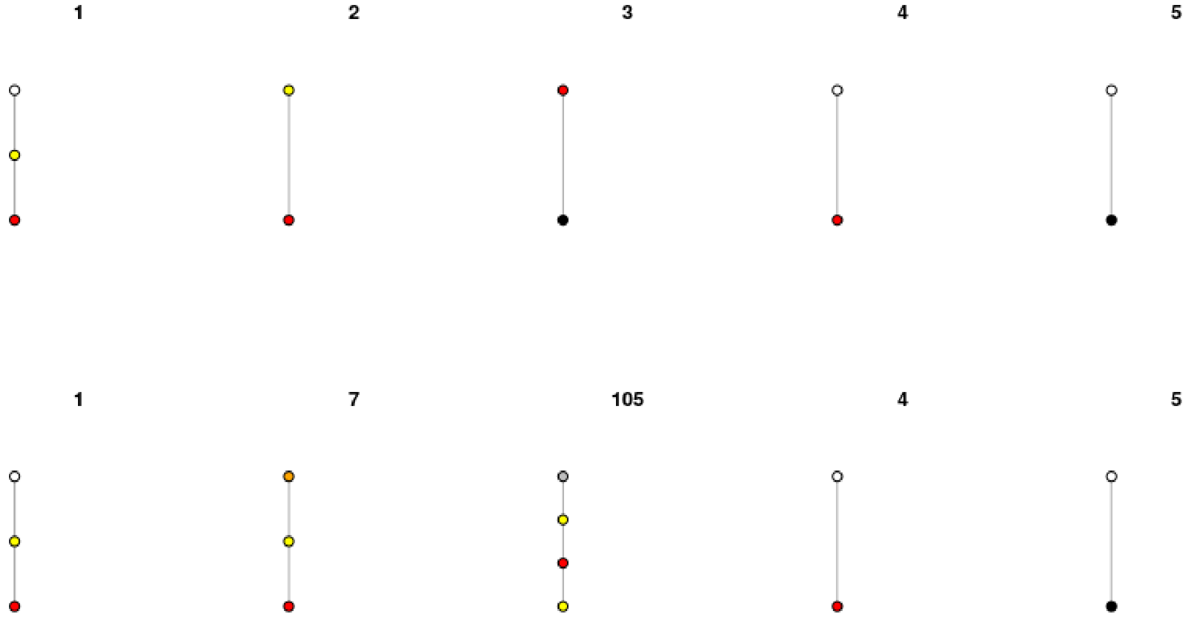

Fig. 11. Most popular time-based neighbourhoods (up) and how they are mostly detected in order-based neighbourhood (down).