Role detection in online forums based on growth models for trees

Alberto Lumbreras, Bertrand Jouve, Julien Velcin, Marie Guégan

Social Network Analysis and Mining manuscript No.
(will be inserted by the editor)

Role detection in online forums based on growth models for trees

Alberto Lumbreras · Bertrand Jouve · Julien Velcin · Marie Guégan

Received: date / Accepted: date

Abstract Some structural characteristics of online discussions have been successfully modeled in the recent years. When parameters of these models are properly estimated, the models are able to generate synthetic discussions that are structurally similar to the real discussions. A common aspect of these models is that they consider that all users behave according to the same model. In this paper, we combine a growth-model with an Expectation-Maximization algorithm that finds different parameters for different latent groups of users. We use this method to find the different *roles* that coexist in the community. Moreover, we analyze whether we can predict users behaviors based on their roles. Indeed, we show that predictions are improved for some of the roles when compared with a simple growth model.

Keywords Role detection · Clustering · Growth models · Social networks · discussion Forums

Alberto Lumbreras
CNRS, Institut de Recherche en Informatique de Toulouse - UMR 5505
2 rue Camichel, 31000 Toulouse
France
E-mail: alberto.lumbreras@irit.fr

Bertrand Jouve
FRAMESPA - UMR 5136, CNRS, Université de Toulouse
5 allée Antonio Machado, 31058 Toulouse, Cedex 9
France
IMT - UMR 5219, CNRS, Université de Toulouse
118 Route de Narbonne, 31062 Toulouse, Cedex 9
France
E-mail: jouve@univ-tlse2.fr

Julien Velcin
Laboratoire ERIC, Université de Lyon
5, avenue Pierre Mendés France, 69676 Bron
France
E-mail: julien.velcin@univ-lyon2.fr

Marie Guégan
Technicolor
975 Avenue des Champs Blancs
35576 Cesson-Sévigné
France
E-mail: marie.guegan@technicolor.com

1 Introduction

Social roles have been widely studied by sociologists, anthropologists, and psychologists. For them, a social role is a behavior that a community expects from an individual that holds some position in that community. A canonical ethnological study of roles in online forums was done in Golder (2003). Online roles have also been studied by computer scientists, who have put more emphasis on the detection of roles. In computer science, a role is usually regarded as a set of user-centered features or as the position that the individual holds in the social graph (Nolker and Zhou, 2005; Himelboim et al., 2009; Lui and Baldwin, 2010; Angeletou et al., 2011; White et al., 2012; Rowe et al., 2013; Buntain and Golbeck, 2014; Choobdar et al., 2017).

Previous works on online discussions are rather top-down or bottom-up approaches. Top-down approaches take an *a priori* definition of one or several roles and examine the community to find persons that match these patterns; an example of this are the methods to find trolls (Kumar et al., 2014), anti-social users (Cheng et al., 2015), influencers (Agarwal et al., 2008), celebrities (Forestier et al., 2012) or leaders (Goyal et al., 2008). On the other hand, bottom-up approaches look for (*a priori* unknown) behavioural patterns among users to obtain a descriptive definition of roles; as a canonical example see, for instance, Chan et al. (2010). In this paper, we propose a rather bottom-up approach.

Our approach is to think of a role not only as a descriptive but also as a predictive aspect of user behavior. Indeed, an interesting characteristic of roles in sociology is that, once we know the role of an individual, we can predict, to some extent, how the individual will behave in a given situation. Imagine that we observe distinct behaviors of individuals in a given population: eating, sleeping, exercising, firefighting and rescuing cats from trees. If we count how many times each person has engaged in each behavior, we might cluster them and find groups of people that behave in a similar way. We might find, for instance, the group of firefighters. However, this cluster is purely descriptive: if we see the firefighter in a given context such as a fire or next to a cat on top of a tree, the cluster will not be able to predict which action the firefighter will choose, either saving the cat or extinguishing the fire. Alternatively, if a predictive model is provided it would be able to predict the action of the firefighter in this context given the past behaviors of the other firefighters in similar situations.

We conceptualize a behavioral function as a probability distribution over the space of all possible behaviors in a given context. We assume that there exists a finite repertoire of behavioral functions and that all the observed behaviors of a user are drawn from one of these functions. We say that two users have the same role if they tend to share the same behavioral function.

In this paper, we set three main goals: (a) proposing a behavioral function for discussion threads, (b) finding groups of users with similar behavioral functions, and (c) testing whether these behavioral functions have predictive power —if they can predict the behavior of a user in a new context.

We will use random graph models (Kolaczyk, 2009) as the basis for our behavioral functions. In particular, we will focus our attention on growth models. Growth models are random generators of graphs that try to mimic the growing mechanism of a network through stochastic processes governed by a set of parameters. Formally, a growth model defines a probability distribution that quantifies the probability of an existing vertex i of being chosen as the parent for a new vertex x_t:

$$p(x_t \sim i | G_{t-1}; \boldsymbol{\theta}) \tag{1}$$

where G_{t-1} is the state of the graph before x_t is attached, and $\boldsymbol{\theta}$ are the parameters of the model. The specification of this probability distribution depends on what we think is a reasonable assumption about the growth process. These models may be seen as behavioral functions since they model the way users choose a post to reply. The repertoire of possible behaviors is then a

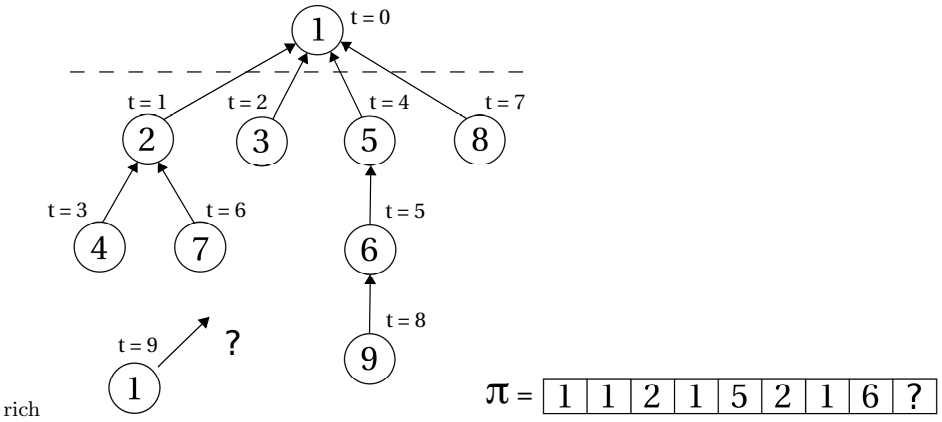


Fig. 1 An example of a thread and its parents vector π. π starts at t=1 because the root post has no parent.

set of parameters $\theta_1, ..., \theta_K$, and the above probability will therefore depend on the θ associated to the author of the post x_t —the author's role.

The structure of this paper is as follows. In Section 2, we present the different growth models that can be applied to tree graphs. In Section 3, we address our goal (a) by an adaptation of one of the models to allow that posts written by different users have different growth parameters $\boldsymbol{\theta}$. The idea is very simple and consists in estimating, by Expectation-Maximization, clusters of users with their own parameters. In Section 4, we address our goals (b) and (c) by finding clusters of users and their parameters in a Reddit dataset, and by testing whether our model is a better predictor in a test set.

2 Network Growth models

Random graph models are stochastic generators of graphs. They may be used to try to reproduce the properties of some real-world network. A good random graph reproduces many relevant properties with few assumptions and a small number of parameters. In that case, the proposed growth mechanism of the random model might be a reasonable approximation of the growth laws under which the real-world graphs evolve (Kolaczyk, 2009).

Following Gómez et al. (2012), we represent a discussion tree at time step t as a vector of parents $\boldsymbol{\pi}_{1:t-1} = (\pi_1, ..., \pi_{t-1})$ where π_n is the parent number of the post written at the time-step n. Note that the shape of the tree at any instant t can be completely recovered from this representation (see Figure 1). With this notation, Equation 1 can be re-expressed as:

$$p(\pi_t = i | \boldsymbol{\pi}_{1:t-1}, \boldsymbol{\theta}) \tag{2}$$

Our growing graph is therefore a tree that starts its growing process with a first vertex (root post) written at $t = 0$ that triggers a conversation. The parent of the next post, written at $t = 1$, will always be the root ($\boldsymbol{\pi}_1 = 1$). Then, at each time-step t a post is added to the tree creating a new vertex (a reply) to an older post i ($\boldsymbol{\pi}_t = i$). One might hypothesize that users tend to reply to popular posts (*preferential attachment*) or that they prefer recent posts, or well-written posts, or that all posts have indeed the same probability of being replied to. Two growth models for discussion trees have been proposed in Kumar et al. (2010) and Gómez et al. (2012). In Kumar et al. (2010), the probability of replying to a post depends on the number of replies and its

Authors	$p(\pi_t = k \vert \boldsymbol{\pi}_{1:t-1}) \propto$	Parameters
Barabási and Albert (1999)	$d_{k,t}^{\alpha}$	degree
Gómez et al. (2010)	$(\beta_k d_{k,t})^{\alpha_k}$	degree, root
Kumar et al. (2010)	$\alpha d_{k,t} + \tau^{t-k}$	degree, recency
Gómez et al. (2012)	$\beta_k + \alpha d_{k,t} + \tau^{t-k}$	degree, recency, root

Table 1 Growth models for online discussions

(a) $\alpha = 0$ **(b)** $\alpha = 1$ **(c)** $\alpha = 1.8$

Fig. 2 Barabasi-Albert graphs with one edge created at every step.

recency. In Gómez et al. (2012), the probability depends on the number of replies, its recency and whether a post is the root.

The remaining of this section is as follows. First, we recall the Preferential Attachment model of Barabási and Albert (1999), and three other growth models for discussion threads (Kumar et al., 2010; Gómez et al., 2010, 2012). Then we present our model, which finds K sets of parameters for K types of user and is based on Gómez et al. (2012).

In the following sections, we describe the growth models that have been proposed to explain the growth of online conversations. A summary is shown in Table 1.

2.1 Barabasi-Albert (1999)

The *preferential attachment* model proposed by Barabási and Albert (1999) is one of the best known growth models. The Barabasi-Albert model builds a graph by sequentially adding its vertices. Once a new vertex t is added to the graph it decides whether to create an edge to an existing vertex i with probability

$$p(\pi_t = i \vert \boldsymbol{\pi}_{1:t-1}) = \frac{d_{i,t}^{\alpha}}{\Omega_t}; \qquad \Omega_t = \sum_{j=1}^{t} d_{j,t}^{\alpha} \tag{3}$$

where $d_{i,t}$ is the degree of the vertex i before vertex t is added. The particular cases of $\alpha = 1$, $0 \leq \alpha < 1$ and $\alpha < 0$ are known as *linear*, *sublinear* and *anti* preferential attachment. For $\alpha > 0$, the model reproduces a rich-get-richer phenomena controlled by α. Figure 2 shows examples of Barabasi-Albert graphs generated with different α. Graphs generated by the Barabasi-Albert model reproduce some interesting properties of the real networks such as a power-law distribution of the vertices degrees.

2.2 Kumar et al. (2010)

In Kumar et al. (2010), the authors propose a model that combines both *preferential-attachment* and *recency*. The higher the degree of a post and the later it was published, the easier for this post to attract the incoming replies. Besides, at every time step, a decision is made to stop the thread or to add a new post. Every new post chooses its parent according to:

$$p(\pi_t = i|\boldsymbol{\pi}_{1:t-1}) = \frac{\alpha d_{i,t} + \tau^{t-i}}{\Omega_t} \qquad \text{for } \alpha \geq 0 \quad \text{and } \tau \in (0,1) \tag{4}$$

and the probability of stopping the thread is:

$$p(\pi_t = \emptyset|\boldsymbol{\pi}_{1:t-1}) = \frac{\delta}{\Omega_t} \tag{5}$$

The authors report that when the alternative function $d_{i,t}\tau^{t-i}$ is used, the recency factor prevents the preferential attachment factor from generating heavy-tailed degree distributions. The normalization factor is:

$$\Omega_t = \delta + \sum_{j=1}^{t} \alpha d_{j,t} + \tau^{t-j+1} = \delta + 2\alpha(t-1) + \frac{\tau(\tau^t - 1)}{\tau - 1} \tag{6}$$

where $2(t-1)$ is the sum of degrees (in-degrees and out-degrees) in a tree of size t and the third term is the result of a geometric series.

The authors also propose an improvement of the model to account for the identity of post authors. For a new post v replying to a post u, its author $a(v)$ can be either $a(u)$ (a self-reply), another author $a(w)$ that has already participated in the chain from u to the root, or some other new author belonging to the set of authors A that have not participated in the chain:

$$a(v) = \begin{cases} a(w) & \text{with probability } \gamma \\ a(u) & \text{with probability } \epsilon \\ a \in A & \text{with probability } 1 - \gamma - \epsilon \end{cases} \tag{7}$$

The Maximum Likelihood Estimators of the parameters $\alpha, \tau, \gamma, \epsilon$ are found by a grid search. The authors show that this model properly reproduces the relationship between size and depth of the trees, the degree distribution at different depths, and the number of unique authors as a function of the thread size in Usenet forums.

2.3 Gómez et al. (2010)

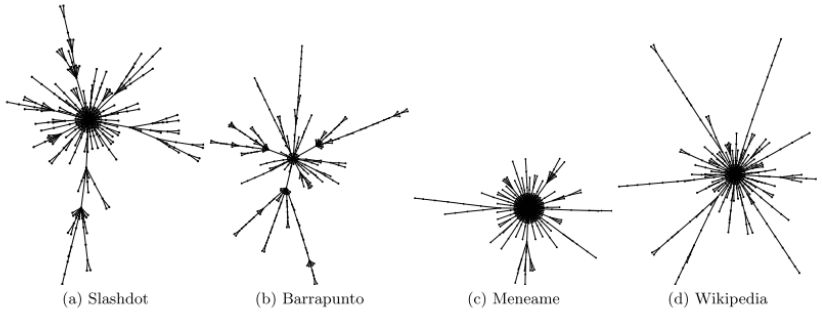
In Gómez et al. (2010), the authors combine *preferential-attachment* with a *bias towards the root*. The probability of choosing an existing parent k is

$$p(\pi_t = i|\boldsymbol{\pi}_{1:t-1}) = \frac{(\beta_i d_{i,t})^{\alpha_i}}{\Omega_t} \tag{8}$$

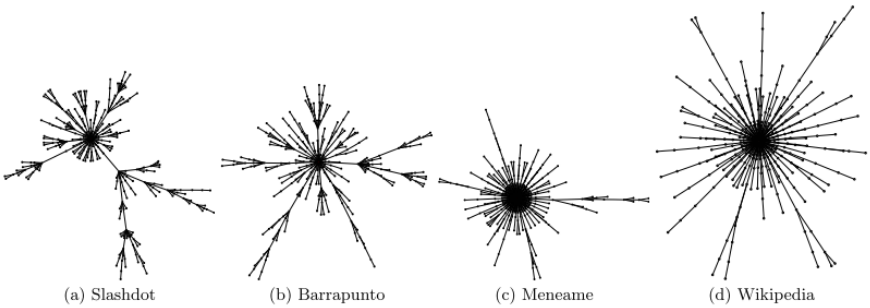
where

$$\alpha_i = \begin{cases} \alpha_1 & \text{for } i = 1 \\ \alpha_c & \text{for } i \in \{2, ..., t\} \end{cases}$$

$$\beta_i = \begin{cases} \beta & \text{for } i = 1 \\ 1 & \text{for } i \in \{2, ..., t\} \end{cases} \tag{9}$$



(a) real



(b) synthetic

Fig. 3 Random graphs for discussion threads (Gómez et al., 2010).

Note that α_i is the preferential attachment exponent, and that if $\alpha_1 = \alpha_c$ and $\beta = 1$ we recover the Barabasi-Albert model of preferential attachment. The normalization factor is:

$$\Omega_t = \sum_{l=1}^{t} (\beta_l d_{l,t})^{\alpha_l} \tag{10}$$

The Maximum Likelihood Estimators of the parameters α_1, α_c and β are found using the Nelder-Mead algorithm to minimize the negative log-likelihood (Nelder et al., 1965). Figure 3 shows some trees generated with their estimated parameters for four different datasets.

2.4 Gómez et al. (2012)

In Gómez et al. (2012), the authors combine *preferential-attachment*, a *bias towards the root*, and *novelty*. Unlike in their former model in Gómez et al. (2010), here they sum these factors instead of multiplying them:

$$p(\pi_t = i | \boldsymbol{\pi}_{1:t-1}) = \frac{\beta_i + \alpha d_{i,t} + \tau^{t-i}}{\Omega_t} \qquad \text{for } \alpha, \beta \geq 0, \quad \tau \in (0, 1) \tag{11}$$

where

$$\beta_i = \begin{cases} \beta & \text{for } i = 1 \\ 0 & \text{for } i \in \{2, ..., t\} \end{cases} \tag{12}$$

The normalization factor resembles the one of Kumar et al. (2010). The differences are the bias towards the root β (only counted once since there is only one root) and the fact that Gómez et al. (2012) gives an out-degree one to the root —which has no practical impact since it acts as an *offset* to the β term:

$$\Omega_t = \beta + 2\alpha(t-1) + \frac{\tau(\tau^t - 1)}{\tau - 1} \tag{13}$$

As in Gómez et al. (2010), Maximum Likelihood Estimators are found by Nelder-Mead optimization. Although the log-likelihood is now non-convex, the authors reported that, for large enough data, the problem seems to approach convexity and the optimization algorithm tends to give the same optimum for different initializations.

3 A new role-based network growth model

The models presented above consider that the probability of choosing a parent is irrespective of the user who writes the post. In other words, they consider that the model parameters are shared by all the users. However, it seems reasonable to think that different users may behave according to different parameters. Some users, for instance, might tend to reply to the root and avoid conversations deeper in the tree. Others might tend to ignore old posts. Others might be especially attracted by popular posts. Formally, we assume that there are K latent types of users and that users of type k behave according to their own group parameters $\boldsymbol{\theta}_k$, for $1 \leq k \leq K$. We think of these parameters as the ones that control the different user roles. Thus, we will say that users with similar parameters (similar behavioral functions) share the same role. In this section, we present a new model, built upon Gómez et al. (2012), that finds different parameters for different groups of users.

3.1 Formalization

We use the same parameters than Gómez et al. (2012): α controls the tendency of users to reply to popular posts, β controls the bias to the root and τ controls how much users penalize old posts.

For any given post n, let d_{p_n} denote the degree of its parent p_n just before n is attached; let r_{p_n} be 1 if p_n is the root, and 0 otherwise. Let l_{p_n} be the number of time-steps elapsed between p_n and n ($l_{p_n} \geq 1$). Let $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be the set of posts of all the conversations, and let $\mathbf{x}_i = \{t_i, d_i, r_i, l_i\}$ be the set of features associated to a post. Let us assume that there are K different types—or roles—of users who behave following different parameters $\boldsymbol{\theta}_1$,...,$\boldsymbol{\theta}_K$ where $\boldsymbol{\theta}_k = \{\alpha_k, \beta_k, \tau_k\}$. Let $\mathbf{z}_u = (z_{u1}, ..., z_{uK})$ be the membership vector of user u where $z_{uk} = 1$ if u belongs to cluster —or role— k and 0 otherwise. To lighten the notation, we use unbolded z_u to denote the position of the active cluster, that is the value k such that $z_{uk} = 1$. Let N_u be the set of posts written by u. The log-likelihood of the whole dataset can be expressed as:

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{u=1}^{U} \sum_{n \in N_u} \ln \left(\alpha_{z_u} d_{p_n} + \beta_{z_u} r_{p_n} + \tau_{z_u}^{l_{p_n}} \right) - \ln \Omega_n \quad \text{for } \alpha_{z_u}, \beta_{z_u} \geq 0, \tau_{z_u} \in (0, 1) \tag{14}$$

where Ω_n is a normalization factor that guarantees that the probabilities of all possible choices sum up to 1. Let t be the time-step when the post n is written and let M denote the set of posts that have been added to the thread before the post n. The normalization factor associated to the post n written at time t (and therefore with t candidate parents) is:

$$\ln \Omega_n = \ln \left\{ \sum_{m \in M} \alpha_{z_n} d_m + \beta_{z_n} r_m + \tau_{z_n}^{l_m} \right\} \tag{15}$$

$$= \ln \left\{ \alpha_{z_n} \sum_{m \in M} d_m + \beta_{z_n} \sum_{m \in M} r_m + \tau_{z_n} \sum_{m \in M} \tau_{z_n}^{l_m - 1} \right\} \tag{16}$$

$$= \ln \left\{ \alpha_{z_n} (2t - 1) + \beta_{z_n} + \frac{\tau_{z_n} (\tau_{z_n}^t - 1)}{\tau_{z_n} - 1} \right\} \tag{17}$$

where the term $(2t-1)$ is the sum of degrees in a tree of size t if the root vertex is considered to have an out-degree 1 (as we do), and $\frac{\tau^t - 1}{\tau - 1}$ is the result of a geometric series. Note that this sum only depends on the time-step t and the model parameters, and not on the particular structure of the thread. [1]

3.2 Expectation-Maximization for the role-based growth model

We want to estimate the parameters of each role $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$ and the latent role of every user $z_1, ..., z_U$. Let $\mathbf{Z}$ be the matrix of membership vectors. If there was one group of $\boldsymbol{\theta}$, there would be no $\mathbf{Z}$—or it would be an array of ones—and we could proceed as in Gómez et al. (2012), and find the Maximum Likelihood Estimators for the parameters of the only cluster. However, if there are different groups then the optimization of the parameter will depend on the group since the parameters will be optimized taking into consideration who belongs to that group. This is a classic scenario that can be solved by Expectation-Maximization (EM). Let us start by expressing the log-likelihood of our model in terms of our latent variables $\mathbf{Z}$:

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \tag{18}$$

Unfortunately we cannot analytically maximize the parameters $\boldsymbol{\theta}$ because of the sum inside the logarithm. We make a trick consisting on multiplying and dividing the joint probability by an arbitrary probability distribution over $\mathbf{Z}$ in order to transform the term inside the logarithm into an expected value:

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \overbrace{\sum_{\mathbf{Z}} q(\mathbf{Z}) \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})}}^{\mathbb{E}_{\mathbf{Z}}[\frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})}]} \tag{19}$$

Thanks to this trick, we can use Jensen's inequality to get the sum outside the logarithm. We know, by Jensen's inequality, that the logarithm of an expected value is always greater than or

[1] In practice, $\mathbf{X}$ may be represented as a matrix of feature vectors $\mathbf{x}_i$ that makes the computing of the log-likelihood easy to vectorize in some programming languages.

equal to the expected value of the logarithm [2]. Therefore:

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \overbrace{\sum_{\mathbf{Z}} q(\mathbf{Z}) \frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})}}^{\mathbb{E}_{\mathbf{Z}}[\frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})}]} \geq \overbrace{\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})}}^{\mathbb{E}_{\mathbf{Z}}[\ln \frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})}]} \tag{20}$$

which is a lower bound of the log-likelihood $\ln p(\mathbf{X}|\boldsymbol{\theta})$. The equality holds if the function is a constant. In our case, when $p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})/q(\mathbf{Z}) = c$, or $p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})/c = q(\mathbf{Z})$. Since $q(\mathbf{Z})$ is a probability distribution, its integral must be 1. Thus, $q(\mathbf{Z})$ that maximizes the above expression is:

$$q(\mathbf{Z}) = \frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})}{\int_{\mathbf{Z}} p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})} = p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}) \tag{21}$$

which is the posterior distribution of $\mathbf{Z}$ given the observed data and the parameters. Replacing $q(\mathbf{Z})$ by the posterior in Equation 20 we obtain:

$$\underbrace{\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}) \overbrace{(\ln p(\mathbf{Z}|\boldsymbol{w}) + \ln p(\mathbf{X}|\mathbf{Z},\boldsymbol{\theta}))}^{\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})}}_{\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})]} - \overbrace{\sum_{\boldsymbol{Z}} p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}) \ln p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta})}^{\mathcal{H}(\mathbf{Z})} \tag{22}$$

where $\boldsymbol{w}$ are the a priori probabilities assigned to each cluster, and $\mathcal{H}(\mathbf{Z})$ is the entropy of the posterior.

For the maximization of the log-likelihood we can ignore the entropy term and we can do an iterative optimization over parameters $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$ and the class assignments $\boldsymbol{z}_1, ..., \boldsymbol{z}_U$ until a lower bound of the likelihood converges. That is, we maximize this term:

$$\underbrace{\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}) \overbrace{(\ln p(\mathbf{Z}|\boldsymbol{w}) + \ln p(\mathbf{X}|\mathbf{Z},\boldsymbol{\theta}))}^{\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})}}_{\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})]} \tag{23}$$

At each iteration, we update the posterior $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta})$ using the θ of the last iteration (E-step) and then we re-compute the parameters $\boldsymbol{\theta}, \boldsymbol{w}$ that maximize the whole term using the updated posterior (M-step). We repeat the expectation and maximization steps until the improvement in the log-likelihood is lower than some threshold.

We now provide the exact equations for the expectation and maximization steps of our model. Let $\mathbf{X}_u$ be the submatrix of $\mathbf{X}$ formed by all the posts written by user u. Let $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_U\}$ be the indicators matrix where $\mathbf{z}_i = \{z_{i1}, ..., z_{iK}\}$. Let $z_{ik} = 1$ if user i belongs to group k and $z_{ik} = 0$ otherwise.

M-step— For the M-step, the expectation of the complete log-likelihood is:

$$\mathbb{E}\left[\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})\right] = \mathbb{E}\left[\sum_{u=1}^{U}\sum_{k=1}^{K} z_{uk}\left\{\ln w_k + \ln p(\mathbf{X}_u|\boldsymbol{\theta}_k)\right\}\right] \tag{24}$$

$$= \sum_{k=1}^{K}\sum_{u=1}^{U} \mathbb{E}\left[z_{uk}\right]\left\{\ln w_k + \ln p(\mathbf{X}_u|\boldsymbol{\theta}_k)\right\} \tag{25}$$

[2] In general, $f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$ where f is a concave function.

where, for a given cluster k, each $\mathbf{X}_u$ proportionally contributes to $\mathtt{E}[z_{uk}]$. We note that the parameters of each cluster can be optimized separately as:

$$\arg\max_{\boldsymbol{\theta}_k} \sum_{u=1}^{U} \mathbb{E}\left[z_{uk}\right] \left\{\ln w_k + \ln p(\mathbf{X}_u|\boldsymbol{\theta}_k)\right\} \tag{26}$$

and for the w parameter:

$$\boldsymbol{w}_k = \frac{1}{U} \sum_{u=1}^{U} \mathbb{E}\left[z_{uk}\right] \tag{27}$$

E-step— In the E-step, we update the posterior:

$$p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) = \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{\sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})} = \frac{\prod_{u=1}^{U} \prod_{k=1}^{K} p(\mathbf{X}_u|\boldsymbol{\theta}_k)^{z_{uk}}}{\sum_{\mathbf{Z}} \prod_{u=1}^{U} \prod_{k=1}^{K} p(\mathbf{X}_u|\boldsymbol{\theta}_k)^{z_{uk}}} \tag{28}$$

which can be easily factorized by users, and then we can obtain the expected value for each z_{uk}:

$$\mathbb{E}[z_{uk}] = \sum_{z_{uk}} z_{uk} \frac{w_k p(\mathbf{X}_u|\boldsymbol{\theta}_k)}{\sum_{k=1}^{K} w_k p(\mathbf{X}_u|\boldsymbol{\theta}_k)} = \frac{w_k p(\mathbf{X}_u|\boldsymbol{\theta}_k)}{\sum_{k=1}^{K} w_k p(\mathbf{X}_u|\boldsymbol{\theta}_k)} \tag{29}$$

where the likelihood $p(\mathbf{X}_u|\boldsymbol{\theta}_k)$ can be also factorized:

$$p(\mathbf{X}_u|\boldsymbol{\theta}_k) = \prod_{n \in N_u} p(\mathbf{x}_n|\boldsymbol{\theta}_k) \tag{30}$$

The E-step is done with Equation 29 while the M-step is done with Equation 25. Because Equation 25 cannot be analytically maximized due to the form of our likelihood, we use the Nelder-Mead optimization as in Gómez et al. (2012).

4 Experiments

Ideally, we would like our model to be descriptive and predictive. That is, we would like it to give clusters of users and parameters for each cluster, and to use these parameters to predict a user behavior in new threads. If the model can make predictions, it would confirm that meaningful roles exist and that the behavior of a group of users is consistent, not just circumstantial or mere noise.

In this section, we infer the parameters for our model and find clusters of users in the `podemos` and `gameofthrones` datasets (Section 4.2) from the Reddit website. Then we benchmark our model against Gómez et al. (2012) (henceforth *gomez* and *lumbreras*) by executing two different tasks. First, we test whether our model can generate synthetic threads that are more realistic than those generated by *gomez* (Section 4.3). Lastly, we test whether our model can make better predictions of post replies in a test set (Section 4.4).

Forum	Threads	Posts	Users	Posts/user
gameofthrones	156,937	3,326,169	278,748	11.9
podemos	88,815	1,368,457	30,032	45.56

Table 2 Datasets used in this paper. All posts (comments and root posts) made between 2013 and 2016 in two Reddit subforums.

4.1 Dataset

Reddit (`http://www.reddit.com`) is a giant forum of forums, called *subreddits*. Subreddits cover all kinds of topics, and new subreddits are continuously created. Since July 2015, a dataset with all Reddit content from 2007 is available for download and updated on a monthly basis (`http://files.pushshift.io/reddit/comments/`). This is, by far, the best publicly available dataset regarding quality and quantity. Some Reddit data has been analyzed, for instance, in Wang et al. (2012). We chose two subforums from which we downloaded all comments between 2013 and 2016 (Table 2):

- `podemos` (`http://www.reddit.com/r/podemos`): a forum for supporters of the Spanish party Podemos. It 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 held in Madrid on October 18th and 19th, 2014. Nowadays, its members use it mainly to share and discuss political news.
- `gameofthrones` (`http://www.reddit.com/r/gameofthrones`): a forum for discussions about the Game of Thrones TV series. Every new season is broadcasted in April, once a week, and every season has 11 episodes.

In order to test the predictive power of our model in a reasonable time on a standard laptop (2.8GHz CPU) and with our R implementation, we divide each user's posts into *training* (50%), *validation* (25%) and *test* (25%). We used the training set of posts to estimate the parameters of *gomez* (α, β, τ) and *lumbreras* ($\boldsymbol{w}$, $\alpha_k, \beta_k, \tau_k$ for each cluster and $p(z_u = k|\mathbf{X}_u, \boldsymbol{\theta}_k)$ for each user). We used the validation set to select the final number of clusters in *lumbreras*, and finally we used the test set to compare the results of the two models. We note that different runs of the same experiment, with different posts in each of the sets, gave similar results (slightly different parameters but the same clusters).

Users with only one or two posts will be assigned to some of the clusters (and its parameters) even if one or two posts is clearly not enough information to infer anything about the user. Thus, we selected the 1,000 users with more posts in the forum to guarantee a high enough number of observations per user. This left us with users that contributed at least with around 1,000 posts. The Automoderator users (programs provided by the moderator to execute automatic tasks such as deleting spams or warning users about the norms of the forum) have been removed.

4.2 Inference

To infer the cluster of each user $\mathbf{z}_1, ...\mathbf{z}_U$ and the cluster parameters $\boldsymbol{\theta}_1, ...\boldsymbol{\theta}_K$ we randomly initialize the parameters α, β, τ for each cluster and run the Expectation and Maximization steps defined in Section 3.2. Different runs with different initial parameters did not show relevant differences in the final results, specially in the clusters with more outlying parameters. Models with a high number of cluster did show more sensibility to initialization, but these models were never selected due to their complexity –number of parameters.

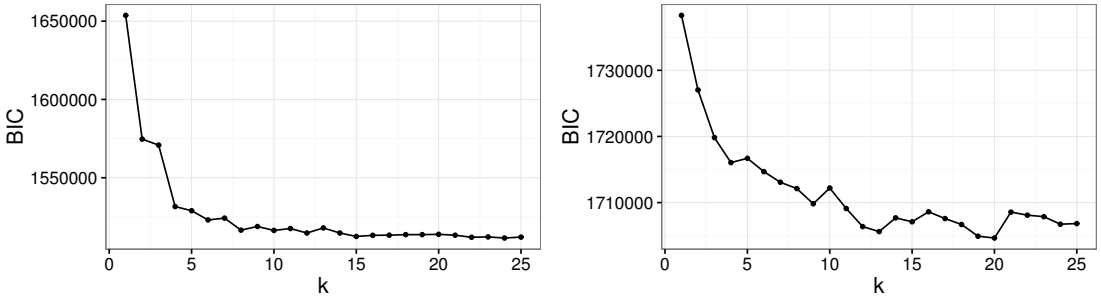


Fig. 4 BIC values in `podemos` and `gameofthrones`

Regarding the M-step, the non-convexity of the log-likelihood might make it necessary to try different re-starts to reduce the odds of being trapped in a local maximum. However, as in Gómez et al. (2012), we noticed that the Nelder-Mead optimization in our Reddit datasets only gives different maxima when the number of observations is small. Several runs showed that the EM barely improves the lower bound of the likelihood after around 15 iterations. Thus, we automatically stopped our algorithm after 20 iterations.

Number of clusters. To decide the number of clusters, we computed the Bayesian Information Criteria (BIC) for multiple models, from one cluster (equivalent to *gomez* model) to more than 50. The BIC is a measure of the likelihood penalized by the complexity of the model. In particular, it is defined as

$$BIC = -2\mathcal{L} + n_p \log n \tag{31}$$

where $\mathcal{L}$ is the log-likelihood, n_p is the number of parameters (for K clusters with three parameters in each cluster, $n_p = 3K + K - 1$) and n is the number of observations. The best model candidates are considered those that minimize the BIC. Since there is some randomness in the BIC score due to different parameter initialization, the score shows oscillations when the differences between two consecutive BIC values are no longer significant. We chose the minimum relative to these oscillations. We computed the BIC in the validation set and we found a minimum BIC at $K = 8$ in `podemos` and $K = 12$ in `gameofthrones` (indeed, with $K = 13$ the algorithm leaves one cluster empty). In Figure 4 we show the BIC curves for `podemos` and `gameofthrones`. We also computed the Akaike Information Criteria (AIC), defined as $AIC = -2\mathcal{L} + 2n_p$ and obtained similar curves. We also computed, for each user, the uncertainty of its classification as $(1 - max(z_{i1}, ...z_{iK}))$ (Bensmail et al., 1997). We obtained a mean uncertainty of 0.03 and a median uncertainty of 0 for `podemos`, and a mean of 0.12 and a median of 0.03 in `gameofthrones`, which means that the model is very sure about the user memberships. The estimated parameters are shown in Table 3 and Table 4.

All clusters have different parameters than *gomez*, meaning that not all users have behaved similarly in our training set. For instance, members of cluster 8 in `podemos` show an extremely high tendency to reply to root posts (high β). Other extreme groups are cluster 8 of `gameofthrones`, made of users whose only predictive parameter is the degree — popularity of the posts —, cluster 5, where users tend to reply to the root posts –either because the thread is short or because they like replying to the root even in long threads–, or cluster 9 with all parameters at 0. Note that the closer the parameters are to zero, the more random is the behavior—degree, recency or root posts would not have any effect and all posts would have the same probability of being chosen as a parent.

cluster	α	β	τ	w	users
1	0	0.21	0.69	0.08	87
2	0.03	1.22	0.88	0.15	150
3	0	1.48	0.90	0.03	28
4	0.08	8.87	0.76	0.16	161
5	0.04	3.30	0.81	0.25	246
6	0.02	0.84	0.18	0.06	172
7	0.29	5.27	0.09	0.12	93
8	0.06	79.4	0.05	0.06	54
Gomez	0.00	3.58	0.93	-	

Table 3 Estimated paramaters for `podemos`

cluster	α	β	τ	w	users
1	0.05	3.37	0.95	0.15	148
2	0.03	0.63	0.96	0.15	145
3	0.06	6.72	0.89	0.11	104
4	0	0.11	0.42	0.01	7
5	0.01	57.8	0.97	0.03	30
6	0.02	0.78	0.84	0.11	114
7	0.1	2.67	0.78	0.09	93
8	5.8e+15	0	0	0.03	28
9	0	0	0	0.02	13
10	0.03	1	0.68	0.08	83
11	0.04	2.65	0.99	0.11	105
12	016	2.13	0.97	0.12	129
Gomez	0.06	2.64	0.93	-	

Table 4 Estimated paramaters for `gameofthrones`

4.3 Structural properties

After having estimated the parameters for *gomez* and *lumbreras*, we generated 10,000 synthetic threads with each model in order to see whether there are structural differences between the two models. We generated the threads as follows. We assume that we know the authors and the order in which they participate, but we do not know to whom they will reply within their posts —we need the authorship information to know which parameters we have to apply. Thus, for a randomly chosen thread in the dataset (with at least a post from the active users), we keep the sequence of authors of the posts chronologically sorted, and we remove the edges. That leaves us with a sorted sequence of posts with no tree structure. Then, we use the estimated parameters to generate a new set of edges keeping the real sequence of authors. Recall that, in *lumbreras*, the parameters applied to a post v depend on the cluster of its author $a(v)$. In other words, a post chooses its parent according to its parameters $\alpha_{z_{a(v)}}, \beta_{z_{a(v)}}, \tau_{z_{a(v)}}$ where $z_{a(v)}$ denotes the cluster of the author of v. If the author is not in our list of analysed users, we use the parameters estimated in *gomez*. Therefore, the only difference between the trees generated by *lumbreras* and *gomez* is in the posts written by the 1,000 most active users, which represent around 25% of the total number of posts in the threads where they participate.

Following (Gómez et al., 2012), we measured the following properties:

- Degree distribution: number of replies to a post
- Subtree size: number of descendants of a post
- Size versus depth: number of posts in the tree versus length of the longest chain

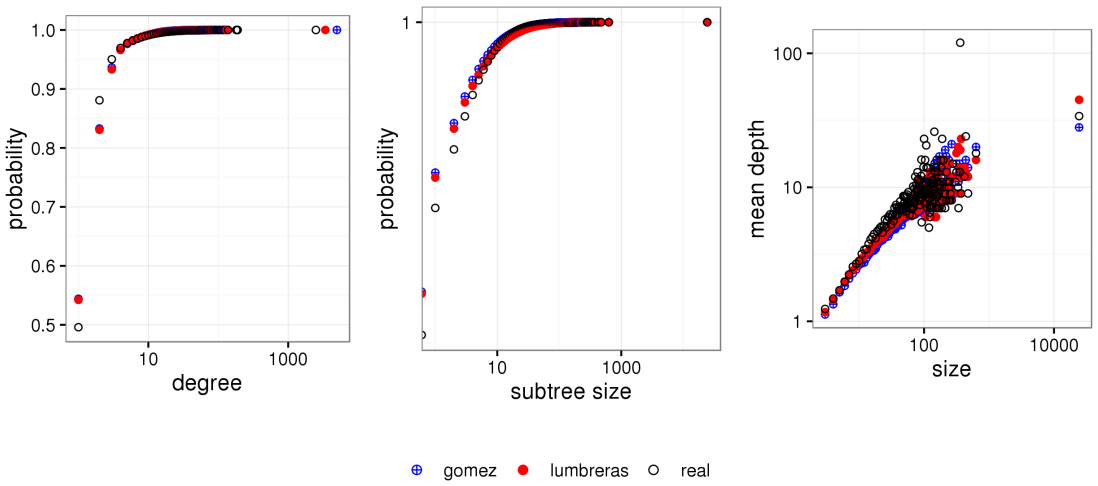


Fig. 5 Properties of synthetic trees and real trees in `podemos`

The results are shown in Figure 5. We observe that the ability to reproduce real structures is very similar in both models. Indeed, a Kolmogorov-Smirnov test did not reject the hypothesis that the cumulative distributions come from the same underlying distribution (p-values over 0.7 and 5% confidence). This is not entirely unsurprising since, as we said above, only posts written by the top 1,000 users have different parameters than the *gomez* model.

4.4 Link prediction

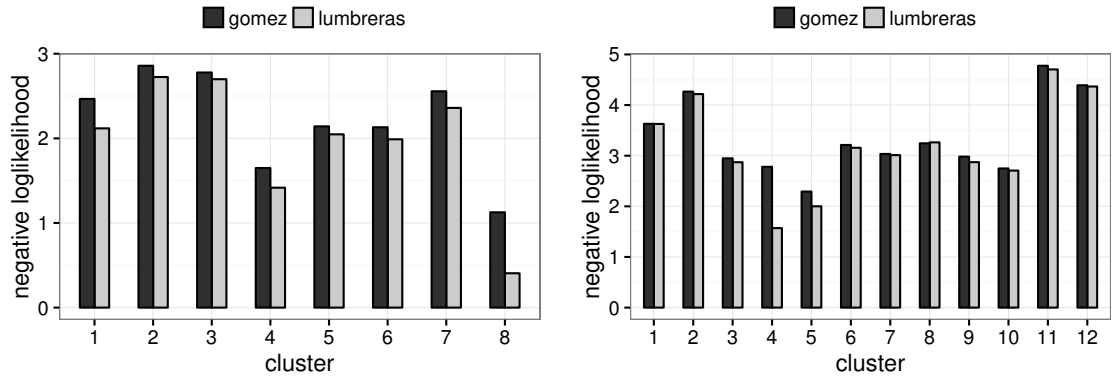
We finally analyzed whether our clusters —roles— have predictive power. If users behave, at some degree, according to role archetypes, we should be able to predict their behavior using the parameters associated to their estimated role. Otherwise, the clusters are only a good description of what happened.

We tested the predictive power of our clusters through a task of link prediction, proceeding as follows: for all the trees in our dataset, we removed the parent of those posts that had been labeled as *test* and we tried to predict their parents with *lumbreras* and *gomez*. We took three different metrics (the likelihood of the test observations, the percentage of hits and the ranking error) and compared the two models in each cluster. For a better understanding of the strengths and weaknesses of the models, we included two reference models: a model that always chooses the post with the highest degree (*barabasi*) and a model that always chooses the most recent post (*recency*).

Likelihood of test data. We compute the mean negative log-likelihoods of the choices given the model parameters and (for the *lumbreras* model) the posts authors. Given a post, a set of candidate parents and the parameters of the model, we know how to compute the likelihood of each possible parent choice. For *gomez*, the log-likelihood of a choice is computed using Equation 11. For *lumbreras*, we first get the most likely cluster of the author

$$z_i^{'} = \arg\max_k p(z_{ik}|\mathbf{X}, \boldsymbol{\theta}_k) \tag{32}$$

and then apply the parameters of the cluster $z_i^{'}$ to the same equation.



(a) `podemos` **(b)** `gameofthrones`

Fig. 6 Mean negative log-likelihoods per cluster in the test set (lower is better)

Figure 6 shows the mean negative log-likelihoods in each cluster. The negative log-likelihood is lower (better) for *lumbreras* in every cluster. We note that the groups of users with a very high β (cluster 8 in `podemos`), is especially predictable; the group of users in cluster 4 of `gameofthrones` (α zero and low β) is also very predictable in terms of the likelihood of their choices; cluster 5 (high β) is the second most predictable in `gameofthrones`. Even if these are also the clusters with better likelihood in *gomez* the improvement in our model is bigger.

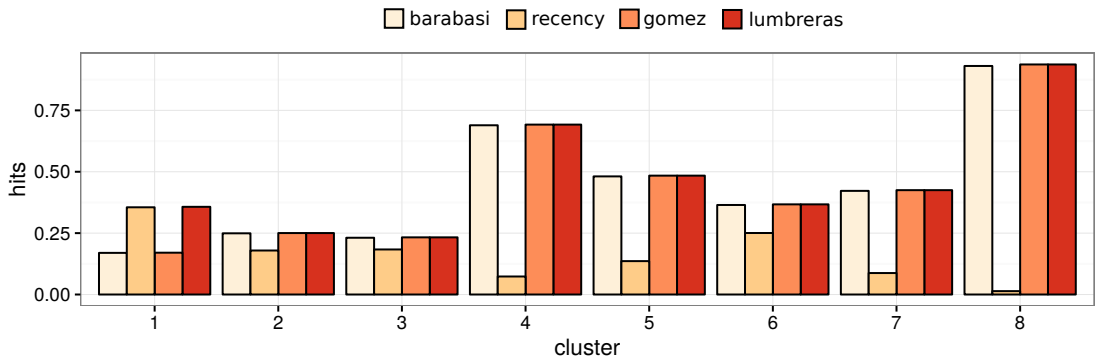
Hits. We define as a *hit* when the chosen parent was the most likely parent according to the model. Figure 7 shows the hits for *gomez*, *lumbreras* and the other two reference models. On the one hand, there is almost no difference between *gomez* and *barabasi*, which means that *gomez* usually assigns more likelihood to the post with more replies. This is surprising since *gomez* has very lower α in the two datasets. A possible explanation is that, since *gomez* has a high β and root posts also tend to have a higher degree, predicting the post with the highest degree often has the same result than predicting the root. Another remarkable result is that the *recency* model is always the worst model except for clusters with $\alpha = 0$ (cluster 1 in `podemos` and cluster 4 in `gameofthrones`), where *recency* outperforms *barabasi* and *gomez*. These are users for whom the degree and the root posts are not as important as for the others, and thus their behaviors are harder to predict by *barabasi* and *gomez*. Because *lumbreras* detected that these users have different behavior, it makes better predictions. Yet, these are the only clusters where *lumbreras* is clearly better than *barabasi* and *gomez*.

Normalized Ranking Error. Our *hits* metric only considers whether the model did a perfect prediction. Yet it is interesting to give some score, for instance, to *almost-perfect* predictions. If the chosen parent was given the second highest likelihood in a very long thread, we might give assign a near 1 score to the prediction. To formalize this idea, we choose to define a Normalized Ranking Error (NRE) as:

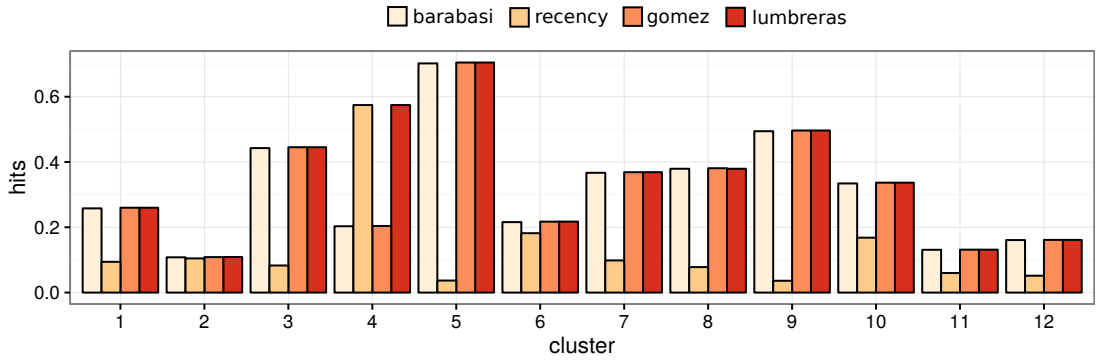
$$NRE = \frac{r-1}{l-1} \tag{33}$$

where r is the position of the chosen parent in the predicted ranking and l is the length of the thread, or the number of parents to choose from ($1 \leq r \leq l$).

While *hits* are low for almost every cluster, the ranking error shows a more optimistic picture (Figure 8): the medians for *gomez* and *lumbreras* are clearly better than those of the reference



(a) podemos



(b) gameofthrones

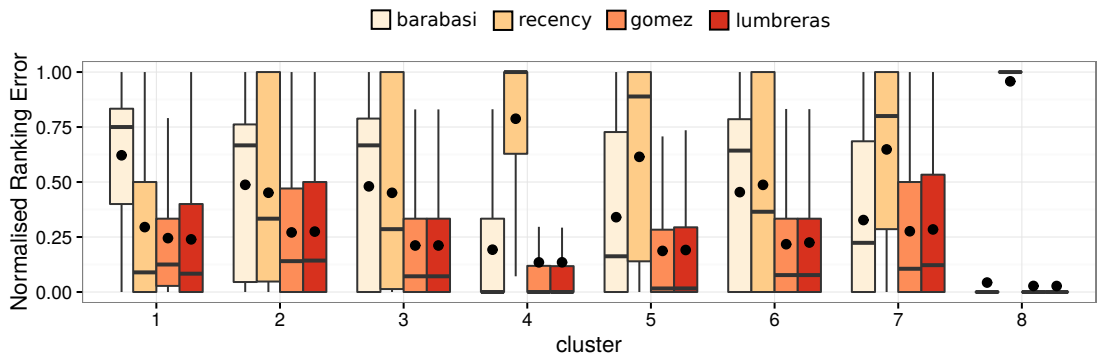
Fig. 7 Hits per cluster in the test set

models. Yet, although for cluster 1 in `podemos` the median score in *lumbreras* is slightly better than in *gomez*, there is barely any difference for the rest of clusters.

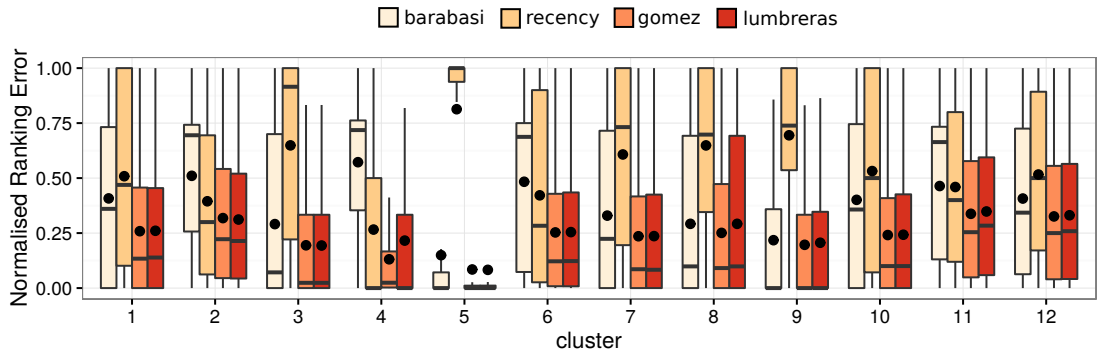
Finally, Figure 9 shows how thread sizes affect the accuracy of the models. We see that the longer the thread, the easier for *recency* to make better predictions and the harder for *barabasi*. In other words, the longer the thread, the less important the degree and the more important the recency—until over 50 posts where it stabilizes. *gomez* and *lumbreras* are almost equivalent for all sizes.

To summarize, we showed that our model, which infers groups of users with different behavioral parameters, gets better likelihoods than *gomez* when measured over unobserved behaviors. This supports the hypothesis that users behave, to some extent, following different behavioral functions. Moreover, for some groups of users with outlier behaviors, our model is able to make better predictions in terms of perfect *hits*. Yet, our (role-based) model does not make better predictions for most clusters. The increase in the likelihood is not enough to make better predictions in those clusters.

To summarize, we showed that our model, which infers groups of users with different behavioral parameters, gets better likelihoods than *gomez* when measured over unobserved behaviors. This supports the hypothesis that users follow different behavioral functions to some extent.



(a) podemos



(b) gameofthrones

Fig. 8 Normalized Ranking Error per cluster in the test set. Boxplots and means (black points).

However, this increase in the likelihood is not enough to make better predictions for all clusters, except for some groups of users with outlier behaviors, where our model is able to make better predictions in terms of perfect *hits*. We checked that the clusters with extreme behaviors do not correspond to some common roles such as trolls or spammers. However, our method might detect this kind of roles as long as their parameters are different from the rest.

5 Conclusions

We have conceptualized user roles as probability distributions over behaviors. In particular, we have studied replying behaviors: tendencies to reply to this or that post given the properties of each of the posts in the thread (number of replies, recency, root or not root)

These tendencies are formalized as probability distributions with three parameters. Our hypothesis has been that users can be divided into subgroups —clusters or roles— whose members behave according to the same parameters.

We set three goals: (a) proposing a behavioral function for discussion threads (b) finding groups of users with the same behavioral function (the same parameters) and (c) testing whether

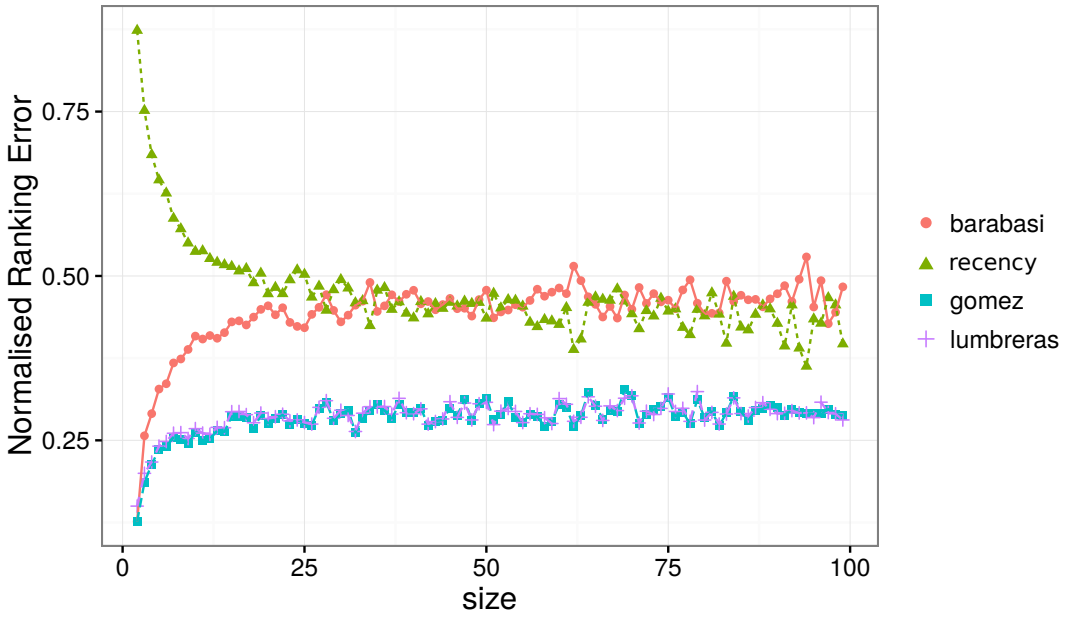


Fig. 9 Normalized Ranking Error by size of thread in `podemos`

these behavioral functions have predictive power —if they can predict the behavior of a user in a new context.

We have shown that, indeed, we can find different groups of users with different behavioral functions. That means that our model can be used, for instance, to better understand the dynamics of a community by inferring different groups of users that have contributed to these dynamics in different ways.

Regarding the predictive power, allowing different model parameters for different subgroups of users increases the likelihood of the model in unobserved behaviors (test set). In terms of practical predictions of *which post will be the next to be replied*, our role-based model is able to improve the predictions over special roles whose parameters are far from the other roles. This is the main interest of our model versus the other models that are not based on roles. Since they assign the same parameters to all users, they are not able to capture these special cases.

Regarding the roles with less extreme values, our model has some descriptive power but the predictive power is almost marginal. It might be that consistency in these behaviors is indeed weak —although not totally random— and that, in terms of signal, there is too much *noise*. Or it might also be that the tree growth models presented in this paper are only able to capture a small part of this behavioral signal.

An open question is whether the groups or roles detected by our method have some counterparts in traditional social role theory. Another interesting line of research is to analyze whether other growth models are able to capture more, or different roles, than the ones analyzed in this paper. The current model assumes that when a user chooses to reply to a post i over the set of posts in a thread, they consider the popularity i, its recency and whether i is the root of the thread. But it seems reasonable to assume that the choice also depends on who the author of i is. Some authors, for instance, might have the ability to write particularly interesting posts. Thus, we might consider that clusters are also associated to an *interestingness* factor and that users in the same cluster write posts with similar levels of *interestingness*. Also, the importance

of reciprocity has been recently shown in Aragón et al. (2017). Adding such new parameters in our model would be straightforward, since only the maximization step in the EM algorithm would be affected. Yet, there is a trade-off between expressiveness and complexity that might make the extension not worthy. Overall, we think that this approach of role-detection based on graph growth models provides a different and original approach to the study of online roles.

References

Agarwal, N., H. Liu, L. Tang, and P. S. Yu (2008, feb). Identifying the influential bloggers in a community. In *Proceedings of the international conference on Web search and web data mining - WSDM '08*, New York, New York, USA, pp. 207. ACM Press.

Angeletou, S., M. Rowe, and H. Alani (2011). Modelling and analysis of user behaviour in online communities. In *Proceedings of the 10th International Semantic Web Conference*, pp. 35–50.

Aragón, P., V. Gómez, and A. Kaltenbrunner (2017). To thread or not to thread: The impact of conversation threading on online discussion. In *11th International AAAI Conference on Web and Social Media*. The AAAI Press: The AAAI Press.

Barabási, A.-L. and R. Albert (1999). Emergence of scaling in random networks. *Science 286*(October), 509–512.

Bensmail, H., G. Celeux, A. Raftery, and C. Robert (1997). Inference in model-based cluster analysis. *Statistics and Computing 7*, 1–10.

Buntain, C. and J. Golbeck (2014). Identifying Social Roles in Reddit Using Network Structure. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, pp. 615–620.

Chan, J., C. Hayes, and E. Daly (2010). Decomposing discussion forums using common user roles. In *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*.

Cheng, J., C. Danescu-niculescu mizil, and J. Leskovec (2015). Antisocial Behavior in Online Discussion Communities. In *AAAI International Conference on Weblogs and Social Media*, pp. 61–70. AAAI Press.

Choobdar, S., P. Ribeiro, and F. Silva (2017). Evolutionary role mining in complex networks by ensemble clustering. In *Proceedings of the Symposium on Applied Computing*, SAC '17, New York, NY, USA, pp. 1053–1060. ACM.

Forestier, M., J. Velcin, A. Stavrianou, and D. Zighed (2012). Extracting celebrities from online discussions. In *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012*, pp. 322–326.

Golder, S. A. (2003). *A Typology of Social Roles in Usenet*. Ph. D. thesis, Harvard University.

Gómez, V., H. J. Kappen, and A. Kaltenbrunner (2010). Modeling the structure and evolution of discussion cascades. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pp. 181–190.

Gómez, V., H. J. Kappen, N. Litvak, and A. Kaltenbrunner (2012, apr). A likelihood-based framework for the analysis of discussion threads. *World Wide Web 16*(5-6), 645–675.

Goyal, A., F. Bonchi, and L. V. Lakshmanan (2008, oct). Discovering leaders from community actions. In *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*, New York, New York, USA, pp. 499. ACM Press.

Himelboim, I., E. Gleave, and M. Smith (2009, jul). Discussion catalysts in online political discussions: Content importers and conversation starters. *Journal of Computer-Mediated Communication 14*(4), 771–789.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data: Methods and Models*. New York: Springer.

Kumar, R., M. Mahdian, and M. McGlohon (2010). Dynamics of Conversations. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 553–562.

Kumar, S., F. Spezzano, and V. S. Subrahmanian (2014). Accurately detecting trolls in Slashdot Zoo via decluttering. In *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 188–195.

Lui, M. and T. Baldwin (2010). Classifying user forum participants: Separating the gurus from the hacks, and other tales of the internet. In *Proceedings of Australasian Language Technology Association Workshop*, pp. 49–57.

Nelder, J., R. Mead, B. J. a. Nelder, and R. Mead (1965). A simplex method for function minimization. *Computer Journal 7*(4), 308–313.

Nolker, R. D. and L. Zhou (2005). Social Computing and Weighting to Identify Member Roles in Online Communities. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pp. 87–93. Ieee.

Rowe, M., M. Fernandez, S. Angeletou, and H. Alani (2013). Community analysis through semantic rules and role composition derivation. *Web Semantics: Science, Services and Agents on the World Wide Web 18*(1), 31–47.

Wang, C., M. Ye, and B. a. Huberman (2012). From user comments to on-line conversations. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 244–252.

White, A., J. Chan, C. Hayes, and B. T. Murphy (2012). Mixed Membership Models for Exploring User Roles in Online Fora. In *Proceedings of the 6th annual international conference on weblogs and social media - ICWSM2012*, pp. 599–602.