



HAL
open science

Automatic role detection in online forums

Alberto Lumbreras

► **To cite this version:**

Alberto Lumbreras. Automatic role detection in online forums. Social and Information Networks [cs.SI]. Université de Lyon, 2016. English. NNT : 2016LYSE2111 . tel-01439342

HAL Id: tel-01439342

<https://theses.hal.science/tel-01439342>

Submitted on 18 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIVERSITÉ
LUMIÈRE
LYON 2

N° d'ordre NNT : 2016LYSE2111

THESE de DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de

L'UNIVERSITÉ LUMIÈRE LYON 2

École Doctorale : ED 512 Informatique et Mathématiques

Discipline Informatique

Soutenue publiquement le 7 novembre 2016, par :

Alberto LUMBRERAS

Automatic role detection in online forums

Devant le jury composé de :

Charles BOUVEYRON, Professeur des universités, Université Paris Descartes, Président

Pascale KUNTZ-COSPEREC, Professeure des universités, Université de Nantes, Rapporteur

Andreas KALTENBRUNNER, Directeur de Recherches, Eurecat-Technology Centre of Catalonia, Rapporteur

Marie GUEGAN, Expert, Examineur

Matthieu LATAPY, Directeur de Recherches, C.N.R.S., Examineur

Julien VELCIN, Maître de conférences HDR, Université Lyon 2, Directeur de thèse

Bertrand JOUVE, Directeur de Recherches, Université Toulouse 2, Co-Directeur de thèse

UNIVERSITÉ DE LYON
École doctorale InfoMaths (ED 512)

THÈSE

en vue d'obtenir le grade de

DOCTEUR

specialité

INFORMATIQUE

présentée par

Alberto LUMBRERAS CARRASCO

Automatic role detection in online forums

Jury:

Bertrand JOUVE	Directeur de Recherche CNRS FRAMESPA & Institut de Mathématiques de Toulouse	Co-Directeur
Julien VELCIN	Maître de conférences (HDR) Université Lyon 2	Co-Directeur
Marie GUÉGAN	Ingénieur Chercheur Technicolor R&D	Encadrante industrielle
Pascale KUNTZ-COSPEREC	Professeur des Universités Université de Nantes	Rapporteur
Andreas KALTENBRUNNER	Directeur Scientifique Eurecat - Technology Centre of Catalonia	Rapporteur
Charles BOUVEYRON	Professeur des Universités Université Paris Descartes	Examinateur
Matthieu LATAPY	Directeur de Recherche CNRS Laboratoire d'Informatique de Paris 6	Examinateur

Abstract

This thesis addresses the problem of detecting user roles in online discussion forums. A role may be defined as the set of behaviors characteristic of a person or a position. In discussion forums, behaviors are primarily observed through conversations. Hence, we focus our attention on how users discuss. We propose three methods to detect groups of users with similar conversational behaviors.

Our first method for the detection of roles is based on conversational structures. We apply different notions of neighborhood for posts in tree graphs (radius-based, order-based, and time-based) and compare the conversational patterns that they detect as well as the clusters of users with similar conversational patterns.

Our second method is based on stochastic models of growth for conversation threads. Building upon these models we propose a method to find groups of users that tend to reply to the same type of posts. We show that, while there are clusters of users with similar replying patterns, there is no strong evidence that these behaviors are predictive of future behaviors —except for some groups of users with extreme behaviors.

In our last method, we integrate the type of data used in the two previous methods (feature-based and behavioral or functional-based) and show that we can find clusters using fewer examples. The model exploits the idea that users with similar features have similar behaviors.

keywords: roles, role detection, social network analysis, forums, machine learning, clustering, Bayesian statistics, Dirichlet process, graphs.

Résumé

Nous traitons dans cette thèse le problème de la détection des rôles des utilisateurs sur des forums de discussion en ligne. On peut définir un rôle comme l'ensemble des comportements propres d'une personne ou d'une position. Sur les forums de discussion, les comportements sont surtout observés à travers des conversations. Pour autant, nous centrons notre attention sur la manière dont les utilisateurs dialoguent. Nous proposons trois méthodes pour détecter des groupes d'utilisateurs où les utilisateurs d'un même groupe dialoguent de façon similaire.

Notre première méthode se base sur les structures des conversations dans lesquelles les utilisateurs participent. Nous appliquons des notions de voisinage différentes (radius-based, order-based, and time-based) applicables aux commentaires qui sont représentés par des nœuds sur un arbre. Nous comparons les motifs de conversation qu'ils permettent de détecter ainsi que les groupes d'utilisateurs associés à des motifs similaires.

Notre deuxième méthode se base sur des modèles stochastiques de croissance appliqués aux fils de discussion. Nous proposons une méthode pour trouver des groupes d'utilisateurs qui ont tendance à répondre au même type de commentaire. Nous montrons que, bien qu'il y ait des groupes d'utilisateurs avec des motifs de réponse similaires, il n'y a pas d'évidence forte qui confirme que ces comportements présentent des propriétés prédictives quant aux comportements futurs –sauf pour quelques groupes avec des comportements extrêmes.

Avec notre troisième méthode nous intégrons les types de données utilisés dans les deux méthodes précédentes (feature-based et behavioral ou functional-based) et nous montrons que le modèle trouve des groupes en ayant besoin de moins d'observations. L'hypothèse du modèle est que les utilisateurs qui ont des caractéristiques similaires ont aussi des comportements similaires.

mots clé: rôles, détection de rôles, analyse des réseaux sociaux, forums, apprentissage automatique, clustering, statistique bayésienne, processus de Dirichlet, graphes.

Remerciements

Je tiens à remercier ceux avec qui j'ai interagi pendant ces ans et qui ont contribué à cette thèse d'une manière ou d'une autre, soit dans la partie *productive* (discussions, séminaires, révisions, logistique,...) soit dans la partie *reproductive* (bières, encouragements,...).

Tout d'abord, je tiens à remercier mon premier encadrant chez Technicolor, James Lanagan, qui m'a accompagné pendant les premiers mois jusqu'à son retour en Irlande, et à Marie Guégan, qui en a pris la relève. Marie est une experte à écouter attentivement et à détecter des erreurs de raisonnement. Cela a été un plaisir d'avoir confronté des idées avec elle sous cette sorte de danse dialectique thèse-antithèse-synthèse. Merci aussi à tous les copains de Technicolor qui ont fait que mon stage soit franchement agréable.

Je tiens bien entendu à exprimer toute ma gratitude à Julien Velcin pour son optimisme et ses orientations, ainsi que pour rassembler les membres du labo autour d'un verre à Les Berthom à chaque fois que j'allais à Lyon pour discuter avec lui. Et également à Bertrand Jouve pour ses conseils, sa capacité à garder la tête froide et sa bonne humeur.

A Marie, Julien et Bertrand, je leur remercie pour leur qualité humaine.

Je voudrais remercier aussi Andreas Kantelbrunner et Pascale Kuntz pour leur temps et leur énergie dédiée à réviser cette thèse, et aussi à Matthieu Latapy et Charles Bouveyron pour m'honorer avec leur participation au jury ainsi qu'au jury de mi-parcours.

Pour les derniers mois de la thèse j'ai déménagé à Toulouse afin de rédiger le manuscrit et pour connaître l'écosystème de recherche toulousain. Je tiens à remercier les gens de l'équipe SIG de l'IRIT et de l'équipe FRAMESPA de l'Université Jean Jaurès pour leur chaleureux accueil. Gràcies sobretot a Florence Sèdes pels seus consells i per la seva tendresa. Merci encore à Bertrand pour avoir fait possible mon déménagement dans cette ville dont il est difficile de ne pas tomber amoureux —au moins pour un républicain espagnol comme moi.

Je tiens à remercier aussi tous les amis et les camarades à Rennes qui m'ont fait oublier —et presque apprécier— la pluie bretonne et la distance entre Rennes et Barcelone. A Arnaud pour ses blagues difficiles à comprendre et son énorme humanité, et à Mario et Cristina, qui ont devenu quelque chose très similaire à une famille. Leur dévouement pour un monde plus juste est toute une inspiration.

No hay páginas suficientes en el mundo para agradecer y dedicar esta tesis a mis padres. Aunque en un principio les hubiera gustado que me quedase en mi puesto en Telefónica, me temo que un psicoanalista diagnosticaría que en el origen de este salto está su inculcación de valores de clase trabajadora, de apego a la cultura y de fuerza de voluntad.

Otro agradecimiento infinito a mi compañera en este y en otros maratones, Elisa, por los kilómetros arriba y abajo, por el apoyo emocional y logístico, por animarme a empezar a correr y por ayudarme a llegar. Y por envolverlo todo de alegría. Quizás habría acabado la tesis sin ella, pero no la habría acabado tan feliz.

A mon frère David

Contents

Contents	ix
1 Introduction	1
1.1 Role Theory	2
1.2 Online forums	4
1.3 Role detection in online communities	5
1.4 Industrial context of the thesis	13
1.5 Outline of the thesis	14
2 Analysis of datasets	17
2.1 Forum dynamics	19
2.2 User dynamics	23
2.3 Conversation dynamics	26
2.4 Summary	32
3 Role detection based on conversation structures	33
3.1 Discussion trees	34
3.2 Methodology	35
3.3 Radius-based neighborhoods	37
3.4 Order-based neighborhoods	47
3.5 Time-based neighborhoods	51
3.6 Comparative analysis	57
3.7 Summary	58
4 Role detection based on thread growth models	61
4.1 Network Growth models	63
4.2 A new role-based network growth model	67
4.3 Experiments	71
4.4 Summary	79
5 Role detection based on features and latent behavioral functions	81
5.1 Introduction. Why a dual-view model?	82
5.2 Related models	82
5.3 Model description	84
5.4 Application to role detection in online forums	88
5.5 Inference	91
5.6 Experiments	92
5.7 Summary	103

6 Contributions and perspectives	105
6.1 Contributions	105
6.2 Roles or not roles?	105
6.3 Perspectives	106
Bibliography	109
A Conditional distributions for the dual-view mixture model	119
A.1 Chinese Restaurant Process	119
A.2 Conditionals for the feature view	120
A.3 Conditionals for the behavior view	123
A.4 Sampling $\beta_0^{(a)}$	126
A.5 Sampling $\beta_0^{(f)}$	127
A.6 Sampling α	128

1 Introduction

The more the individual is concerned with the reality that is not available to perception, the more must he concentrate his attention on appearances.

Ervin Goffman

Contents

1.1	Role Theory	2
1.2	Online forums	4
1.2.1	Vocabulary	4
1.2.2	Forums as graphs	5
1.3	Role detection in online communities	5
1.3.1	Social sciences	6
1.3.2	Blockmodeling	7
1.3.3	Feature-based	9
1.3.4	Triads and motifs	10
1.3.5	Other methods	12
1.4	Industrial context of the thesis	13
1.5	Outline of the thesis	14

FROM the first newsgroups in the 1980s up to the present day, online forums have always been amongst the most popular ways of online communication. Even after the huge expansion of social networks, forums are still used by 15% of online users in the U.S. (Duggan, 2015). Modern forums cover a large variety of topics such as politics, health, technology or video-games, and many applications such as questions & answers (Q&A), sharing and discussing news (newsboards), seeking other students help in Massive Open Online Courses (MOOC) or discussing among supporters of a political cause. The current growth of forums like Quora, Reddit, or StackExchange suggests that this form of online communication is stronger than ever and that it still has an enormous potential ahead in terms of the number of users and application to novel areas.

As forums become more populated and users produce more content, they open the door to new challenges and opportunities in fields like Computer Science, Complex Systems and Sociology. Computer Scientists may develop tools to help users explore the content of the forums so that the experience is satisfactory and users do not abandon the community; areas like Machine Learning, Information Retrieval and Recommender Systems are starting to play an important role here. The framework of Complex Networks is excellent to analyze and understand how new dynamics emerge from *micro* to *macro* levels, or from the individual to the community. Some models have been proposed, for instance, to explain how the structure of a conversation grows over time (Kumar et al., 2010; Gómez et al., 2012), and it is astonishing to see how some mathematical models

of human dynamics have the simplicity of physical laws. Online forums are also places where communities emerge (Kollock, 1998). The net of interactions between users in the form of posts, or comments, ends up creating shared meanings and, in general, a common culture that defines and delimits the set of possible behaviors. As an example, the popular slogan *don't feed the troll* suggests users to ignore those playing the role of a *troll*. Nonetheless, the most fruitful approaches need to be interdisciplinary. Forums are big data, complex systems and human communities and, as such, an integrated view is likely to shed more fruitful results than the sum of the parts (McFarland et al., 2016; Tinati et al., 2014).

The central topic of this thesis is *online roles*. 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 occupies 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.

We think that one of the most interesting aspects 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. Roles are both descriptive and predictive categories of behavior. We know how to address the two problems separately: a descriptive categorization of users is an unsupervised learning task (clustering), while learning to predict users behaviors from a log of past behaviors is often a supervised learning task. However, integrating both tasks is not trivial.

The aim of this thesis is precisely to integrate both a *descriptive* and a *predictive* vision of online roles. By thinking of roles as archetypical behaviors, we will explore some ways to find clusters of users with similar behaviors.

The remaining of this chapter is as follows. In Section 1.1, we will introduce the notion of role in sociology. Before going any further into how roles are conceptualized in online contexts we introduce, in Section 1.2.2, some basic vocabulary of discussion forums and different formal representations of user interactions; we will apply our role detection methods on these representations. In Section 1.3 we will introduce the State of the Art of role analysis and detection in online forums. We close this chapter in Section 1.5 with a summary and an outline of the thesis.

1.1 Role Theory

Imagine a metro station. If we observed the social life for several days, including the behaviors and the interactions of the passengers, what would we find? A big amount of what happens there seems ephemeral and chaotic but, after a few days, we would notice some patterns in many aspects such as the way people choose their seats, their interpersonal distances or their conversations (Nash, 1975). Learning these patterns helps to understand the inner mechanisms of the metro community such as its rituals and its norms. Despite being the first time most of the metro passengers see each

other, and despite the absence of explicit norms for most of the observed behaviors, the emergence of these patterns is possible thanks to a mixture of biology, culture, and a network of inter-personal interactions that enable the propagation of norms, beliefs, and expectations between passengers who never shared the same wagon.

A very special pattern is what sociologists call a *role*. Following Ervin Goffman's metaphor of the social drama (Goffman, 1959), a role is a script that an individual (actor) plays in front of others (audience) because the others expect the individual to do so. In the metro, when an old lady stands beside a sitting teenager, everyone expects the teenager to offer his seat to the lady. This expectation does not depend on the particular teenager nor the particular lady, but on their social positions.

The idea that a role is attached to a social position has two interesting consequences: on the one hand, a role census of a community, and the description of behaviors attached to every role, represent a good summary of the community. On the other hand, knowing the role of an individual allows us to predict the way the individual is likely to behave in a given situation, even if we have no other information about the individual. Or to put it in Golder and Donath words (Golder and Donath, 2004):

Roles, in turn, are useful because, when they comprise sets of expectations, they allow us to generalize across people and have some a priori knowledge about entire categories of people, how to act toward them and what to expect in their actions toward us. More importantly, we categorize people precisely in order to understand them (Foucault, 2003), because having expectations is so vital to our understanding of others.

Although *role theory* has been around at least since the first half of the 20th century, there is neither a unified theory nor a single definition of *role*. In *Role Theory: Expectation, Identities, and Behaviors* (Biddle, 1979), the author complained:

And as a matter of fact, the plasticity of role concepts is one of the major reasons why the role orientation is popular. Since social scientists have rarely found it necessary to spell out exactly what they meant by role, social position, or expectation, these terms could be applied with impunity to almost any purpose. Role might be considered an identity, a set of characteristic behaviors, or a set of expectations; expectations might be descriptive, prescriptive, or evaluative; and so it goes. And given such a superficial level of analysis, it would be possible either to exalt or decry role notions with equal enthusiasm since each of us would be talking about somewhat different things.

Social scientists have confronted the issue of roles from different approaches which, in part, reflect the sociological debate. The *functionalist* approach considers that a society, as the human body, is a system where every part has a function (Linton, 1936; Parsons, 1951); functionalism sees a role as the behaviors that the community expects from an individual *because* he occupies a given position. The *symbolic interactionism* approach thinks of a society as the result of the interactions between individuals (Mead, 1934);

interactionism sees that roles are continually negotiated and individuals take, define or modify roles as a result of their interactions. The *structuralist* approach to roles models a society as a set of positions, filled by persons and relationships between positions [Nadel \(1957\)](#); individuals that fill the same positions are said to have the same role. Though it imposes a very restrained view of roles, it is easy to formalize mathematically; later we will see a popular formalization called *blockmodeling* introduced by [White et al. \(1976\)](#). See [Biddle \(1986\)](#) for a complete review and further discussion about the different approaches.

Biddle proposes a soft definition that captures the elements that are shared by most approaches ([Biddle, 1979](#)):

In current social science the term role has come to mean a behavioral repertoire characteristic of a person or a position; a set of standards, descriptions, norms, or concepts held for the behaviors of a person or social position; or (less often) a position itself.

1.2 Online forums

Before delving into how roles are analyzed and detected in online communities, let us first introduce some basic concepts regarding the vocabulary of forums and their graph representations.

1.2.1 Vocabulary

- A **forum** (also *online forum*, *internet forum* or *message board*) is an online discussion site where people can hold written conversations in the form of textual messages. Forums are made by *threads* and threads are made by *posts*. Forum sites usually sub-divide their content in big topics such as sports, politics, science, statistics or programming. Since the same site often hosts forums of different topics we may also use the word **subforums**.
- A **post** (also *comment* or *message*) is a text message written by a user to participate in a conversation thread. The **root** post is the first message of a thread. Usually, root posts are questions or links to some content (e.g., an online article) that might interest the community.
- A **thread** (also *discussion thread* or *conversation*) is a conversation around the topic started by the root post. A thread is made by posts where users reply to each other. In some contexts we will also use the term **conversation** to talk about specific parts of the thread instead of the whole thread.
- A **reply** is a post written in reply to another post. Every post, except the root, is a reply to some other previous post. The replied post is the **parent** of the post.

1.2.2 Forums as graphs

In this thesis, we study roles from a structural point of view. To this aim, we represent user activity in the form of graphs. Deciding what vertices and edges represent must be subjected to the type of analysis we want to make. We use three representations: in the *coparticipations graph*, edges represent participations in the same thread; in the *interactions graph*, representation edges represent replies between users; in the *tree graph*, vertices represent posts and edges represent replies from one post to another.

Coparticipations graph: A coparticipation graph is an undirected graph where vertices represent users, and where an edge between u and v , denoted as (u, v) , indicates that u and v have participated at least once in the same thread.

Interactions graph: An interactions graph (or replies graph) is a graph where vertices represent users and an edge between u and v , denoted as (u, v) , indicates that u has replied to a post of user v in some of the threads.

Discussion tree graph: A discussion tree graph is a graph that represents a single thread. Vertices correspond to posts and an edge between post u and v , denoted as (u, v) , indicates that u is a reply to v .

Note that we can—and we should—customize these graphs to our needs. For instance, we can impose a threshold for an edge to exist (a minimum number of coparticipations or interactions), or put weights on edges, or make the interactions graph undirected. Also, we can set stronger conditions for an edge to exist, such as the existence of replies in both directions or a minimum number of replies.

Figure 1.1 we show a real conversation of a forum represented both as a tree and as an interaction graph (although an interactions graph is not limited to one single thread). Unlike the interactions and the coparticipation graphs, which aggregate interactions over a period of time, a tree graphs represent the dynamics of a conversation in a thread without any loss of structural information. That makes them very appropriate to our task of finding roles based on how different users converse. For this reason, we will especially focus our attention in tree graphs.

Looking at the tree of Figure 1.1 one may think—and rightly so—that the textual content in each of the vertices would give much relevant information about a user’s behavior. Content and structure are not competitors but complementary perspectives. For example, a debate between two users may be detected either through textual analysis or through a cascade such as $C \leftarrow E \leftarrow C \leftarrow E$ or $C \leftarrow E \leftarrow C \leftarrow E$ in the figure.

1.3 Role detection in online communities

In some sense, an online forum is similar to the metro example that we introduced earlier. Imagine now an online forum. Just as it happens in the metro, forums are places with none or few explicit norms and where most of the participants have never seen each other. The collective behavior of the forum users also follows several patterns (Whittaker et al., 1998) and individuals also play several social roles.

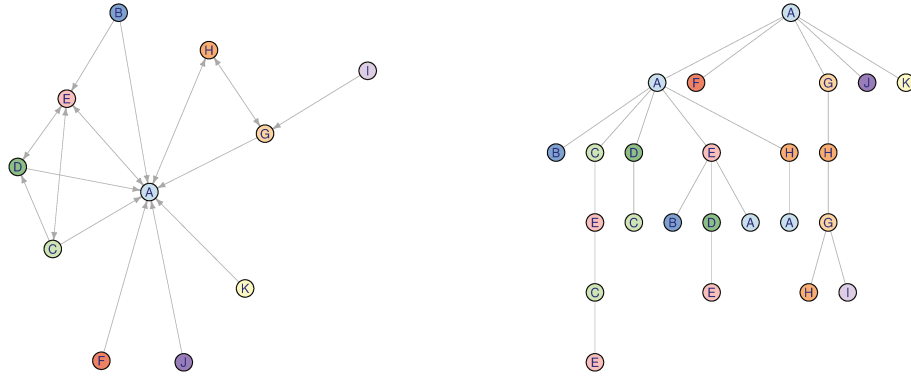


Figure 1.1 A thread represented as a graph of interactions and as a tree graph

To find these roles, we can follow either a top-down or a bottom-up approach. 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 summary, top-down approaches are techniques to find known roles, while bottom-up approaches are techniques to detect unknown roles.

In this section, we present the main approaches to the problem of detecting users roles in social networks and especially in forums.

1.3.1 Social sciences

Ethnological and sociological methodologies are arguably the most complete when it comes to identifying and describing the roles played by members of a community.

Ethnologists live within a community for some time, talk to its members, learn the cultural codes and norms of the community, and produce a report where all these are synthesized. In online communities, the most popular ethnological study has been that of Golder and Donath (2004), who studied the roles in Usenet. These roles are the celebrity, the lurker, the newbie, the troll, the ranter, and the flamer.

Leave et al. (2009) describe a sociological multi-level iterative strategy to identify roles in online communities. They suggest the combination of a general qualitative and quantitative analysis—where a first draft of the existing roles will come up—and a more specific analysis of the structural and behavioral patterns (e.g.: egonets) of each role to

refine the findings at the higher levels. The analysis goes on until the researcher finds a good correlation between a role category and a behavioral pattern.

It is important to choose the right starting point for social role research. Starting from simple behavioral regularities or distinctive social positions is a flawed approach because it is overly inductive: of the infinite patterns we can find, there is no reason to think that the ones that initially stand out will be of any social significance. Starting from abstract categories like 'altruist' commits the error of over-deduction: simply because we can label a set of activities as altruistic does not necessarily mean that those behaviors are motivated by altruism. The key is to connect higher levels and abstract categories to observable behavioral regularities and distinctive network positions in order to discover types of actors relevant to theoretical questions in social research. Gleave et al. (2009)

When the size of the community is significant, the researcher can only identify and describe the key roles by studying a small set of individuals. However, these methods are not able to label each member of the community.

1.3.2 Blockmodeling

The family of blockmodeling algorithms searches groups of users who share similar relationships to the other groups. Blockmodeling works over the *sociomatrix*, a $U \times U$ matrix where the cell (i, j) contains a value that reflects the relationship from user i to user j . The relationship may be, for instance, the level of affection from i to j , the number of telephone calls or the number of replies from i to j in a discussion forum.

The initial motivation of blockmodeling raised from sociology and the study of *structural roles* of individuals that hold the same *position* in their social networks. If our community is a family and relations are “is son of”, the positions of two siblings, though not exactly the same, are structurally equivalent since their relationships with the rest of the family members are exactly the same: they both are siblings of their parents, as well as nephews of their mothers’ brother, and so forth.

Historically, different notions of similarity have motivated different algorithms. *Structural equivalence* (Lorrain and White, 1971) considers that two users are equivalent if they are equally related to the same other users. *Regular equivalence* White and Reitz (1983) considers that two users are equivalent if they are equally related to the same other groups of equivalent users. Finally, *stochastic equivalence* (Holland et al., 1983) considers that two users are similar if they *tend* to be equally related to the other groups of equivalent users. Following this last definition, Nowicki and Snijders (2001) derived a general Bayesian model that has been the base of many Stochastic Blockmodeling adaptations to different kinds of data.

For the sake of illustration, let \mathbf{G} be a binary matrix that encodes the set of observed relationships between a group of U individuals (e.g.: $g_{ij} = 1$ for i likes j and 0 otherwise). Let us assume that each individual is member of one out of K social groups and let

z_1, \dots, z_U denote the latent, or unknown, membership of each individual. Let η_{ab} be the probability that a member of the group a likes a member of the group b . If we knew the memberships \mathbf{z} and the probabilities $\boldsymbol{\eta}$, we could compute the likelihood of the observed \mathbf{G} as:

$$p(\mathbf{G}|\boldsymbol{\eta}, \mathbf{z}) = \prod_{i=1}^U \prod_{j=1}^U (\eta_{z_i, z_j})^{g_{ij}} (1 - \eta_{z_i, z_j})^{\overline{g_{ij}}} \quad (1.1)$$

But we do not know the values of \mathbf{z} and $\boldsymbol{\eta}$. The *inverse*, that is inferring the *posterior* probabilities of \mathbf{z} and $\boldsymbol{\eta}$ from the observed data, can be done via Bayes formula:

$$\underbrace{p(\boldsymbol{\eta}, \mathbf{z}|\mathbf{G})}_{\text{posterior}} = \frac{\overbrace{\left(\prod_{i=1}^U \prod_{j=1}^U (\eta_{z_i, z_j})^{g_{ij}} (1 - \eta_{z_i, z_j})^{\overline{g_{ij}}} \right)}^{\text{likelihood}} \underbrace{p(\mathbf{z})p(\boldsymbol{\eta})}_{\text{prior}}}{\sum_{\mathbf{z}} \int_{\boldsymbol{\eta}} \underbrace{\left(\prod_{i=1}^U \prod_{j=1}^U (\eta_{z_i, z_j})^{g_{ij}} (1 - \eta_{z_i, z_j})^{\overline{g_{ij}}} \right)}_{\text{marginal likelihood}} p(\mathbf{z})p(\boldsymbol{\eta})} \quad (1.2)$$

where $p(\mathbf{z})$ and $p(\boldsymbol{\eta})$ are some *prior* probabilities that we assign to the latent variables. The priors are often chosen by analytical convenience so that the term in the denominator becomes integrable. This happens, for instance, when the prior is *conjugate* to the likelihood. Unfortunately, likelihood functions often lack a conjugate prior. In these cases, the inference is mostly made either by Monte Carlo sampling (usually Gibbs Sampling) or by Variational Inference. While the former is a technique to draw samples from the true posterior, the latter is a (computationally faster) technique to obtain an analytical approximation to the posterior. Because inference in blockmodels is very computationally demanding due to its relational nature, the popularity of the variational approaches has increased in the last years (Daudin et al., 2008; Latouche et al., 2012).

Stochastic Blockmodeling can be adapted to edges with all sorts of weight distributions such as Poisson (DuBois et al., 2013) or Gaussian. Other variations are also possible, such as allowing users to belong to many clusters (Mixed Membership Stochastic Blockmodels, Airoldi et al. (2008)) or to change their interaction patterns, and thus their roles, over time (Fu et al., 2009; Ho et al., 2011). An appealing contribution is that of non-parametric blockmodeling that infers the number of clusters using a Dirichlet Process prior over the cluster assignments (Kemp et al., 2004, 2006).

Although blockmodeling can be applied to many types of adjacency matrices, its application to forums has some particular limitations. Figure 1.2 shows the blockmodels in an undirected binary matrix of interactions between the 100 most active users in the forums (according to the model of Daudin et al. (2008) available in the R package `mixer`). While blockmodels uncover interesting structures in interaction or coparticipation matrices, such as blocks of users that avoid each other, they have important limitations in the context of online forums. First, the white blocks detected, might represent a group of

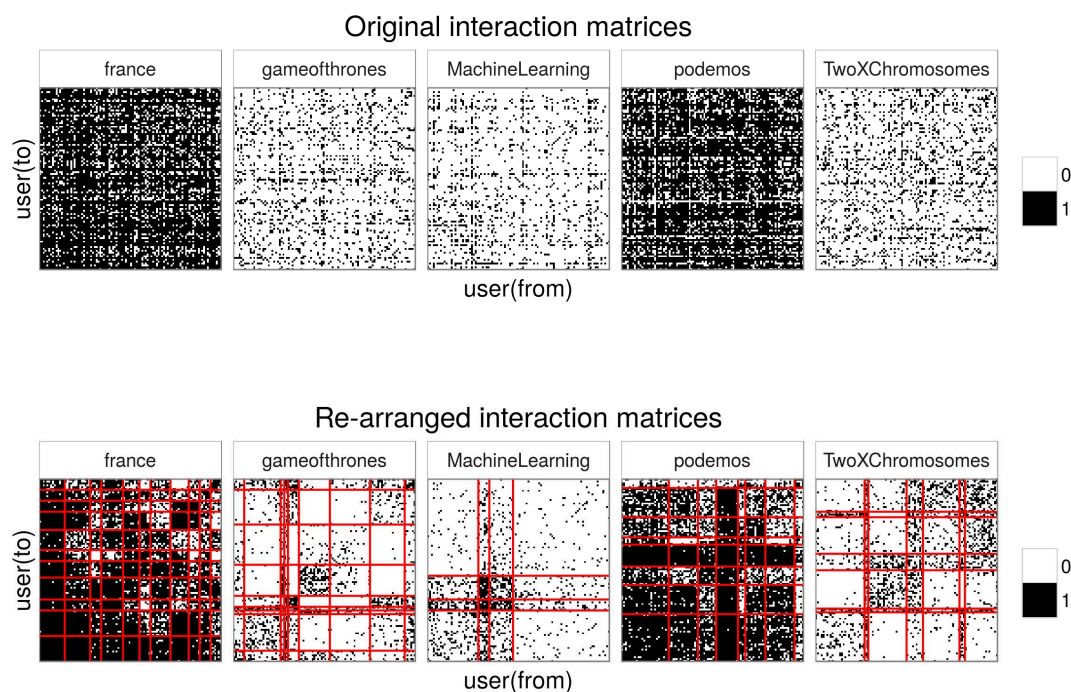


Figure 1.2 Blockmodeling over the interactions matrix of the 100 most active users

users that arrived some time later and thus did not have time to interact with the other users—to solve this, we would need a dynamic or longitudinal blockmodeling where nodes can appear and disappear. Second, even if a blockmodel is applied to analyze the block structure of a single thread, the positions of a user in two different threads are hardly comparable, and it is difficult to say anything about the role of a user beyond a particular conversation.

1.3.3 Feature-based

Most role detection in discussion forums are based on describing each user through a set of features and then classifying users by clustering or by some rules based on thresholds to find groups of users with similar features. The resulting clusters are interpreted as roles.

[Himelboim et al. \(2009\)](#) search *catalists*—users that start discussions and attract many participants and posts—since they are the ones that set the topics that the community debates about. They classify a user as a catalyst if they have a high score in three metrics: number of posts attracted by their threads (reply share), the number of users in their threads (replier share), and ratio of threads that get comments from at

least two users.

Welser et al. (2007) and Gleave et al. (2009) selected users with recognized roles and found that the egonet of a user is very attached to its role. Based on this idea, Buntain and Golbeck (2014) select some features from the egonet and find clusters over them. The features are undirected network density, low degree distribution, the ratio of neighbors with low degree, the proportion of intense ties, clustering coefficients, and triangle density.

Chan et al. (2010) and White et al. (2012) select a set of user features and perform a clustering over them. They analyze the resulting clusters and manually merge some of them if they represent the same type of user; then each cluster is defined as a role. The set of features is the indegree and outdegree exponents of their egonets, the ratio of neighbors that replied to each other, the mean and standard deviation of posts per thread, their in-degree, the number of posts with at least one reply, the ratio of initiated threads. Using these same features, Angeletou et al. (2011) build a rule-based system with adaptive thresholds instead of a clustering.

Rowe et al. (2013) extend their work on Angeletou et al. (2011) and propose five features: the proportion of users that the user has replied to, the proportion of threads created by the user, the proportion of threads started by the user, and the average points per post awarded to the user.

In the context of TV-series forums, Anokhin et al. (2012) use some basic features and adapt some others from the citation analysis literature such as the h-index (Hirsch, 2005) and the g-index (Egghe, 2006). They also use a cross-topic entropy to measure how focused is a user in a TV-show.

Forestier et al. (2012) look for celebrities based on the ethnological study of Golder and Donath (2004). They build some meta-features based on the number of posts the in-degree and out-degree of the user in the reply network, and the number of threads.

One of the advantages of these methods is that the results are easy to interpret since each role is clearly attached to a range of values in each feature. Besides, a different choice of features will give us a set of roles based on different criteria, which may be useful if the researcher is especially interested in one specific aspect of user behavior.

Table 1.1 summarizes the main features used in the literature.

1.3.4 Triads and motifs

Another approach to detecting user roles is looking at the triads where they appear in the interaction graph (or some other representation). A triad is a graph consisting of three vertices. There are 16 possible triads, from the empty triad with no edges to the fully connected triad (Figure 1.3). Triads have been largely used in social network analysis since they represent their most basic building blocks (Wasserman and Faust, 1994). When a triad is statistically over-represented—it appears significantly more frequently than expected—it is called a motif (Milo et al., 2002). Motifs have been used in online forums to analyze patterns over the Q&A graph (Adamic et al., 2008).

To test whether a triad is over-represented, we need a reference null model. The null model can be built by randomizing the original graph while preserving some of its

Feature	References
Threads started	
Threads started	Rowe et al. (2013)
Threads started with replies	Himmelboim et al. (2009); Chan et al. (2010)
Posts in threads	Himmelboim et al. (2009)
Users in threads	Himmelboim et al. (2009)
Posts written	
Posts with reply	Chan et al. (2010)
Votes per post	Rowe et al. (2013)
Activity levels	
Posts	Forestier et al. (2012); Rowe et al. (2013)
Mean Posts/thread	Chan et al. (2010)
Std. dev. post/thread	Chan et al. (2010)
Focus Dispersion	Rowe et al. (2013)
Avg. post-to-thread tf-idf	Nolker and Zhou (2005)
Avg. poster-to-poster tf-idf	Nolker and Zhou (2005)
Threads with reciprocal replies	Chan et al. (2010); White et al. (2012)
Egonet	
In-degree distribution exponent	Chan et al. (2010); Forestier et al. (2012)
Out-degree distribution exponent	Chan et al. (2010)
Reciprocal neighbours	Chan et al. (2010); White et al. (2012)
Indegree	Chan et al. (2010); White et al. (2012)
Outdegree	Forestier et al. (2012); Rowe et al. (2013)
Degree	White et al. (2012); Forestier et al. (2012)
Neighbours with low degree	Rowe et al. (2013)
Weighted indegree	Nolker and Zhou (2005); Buntain and Golbeck (2014)
Weighted outdegree	Buntain and Golbeck (2014)
Density	White et al. (2012)
Neighbours with intense ties	White et al. (2012)
Clustering coefficient	Buntain and Golbeck (2014)
Triangle density	Buntain and Golbeck (2014)
Replies graph	
Betweenness	Nolker and Zhou (2005)
Closeness	Nolker and Zhou (2005)
Language	
Content-based features	Lui and Baldwin (2010)

Table 1.1 Features for role detection

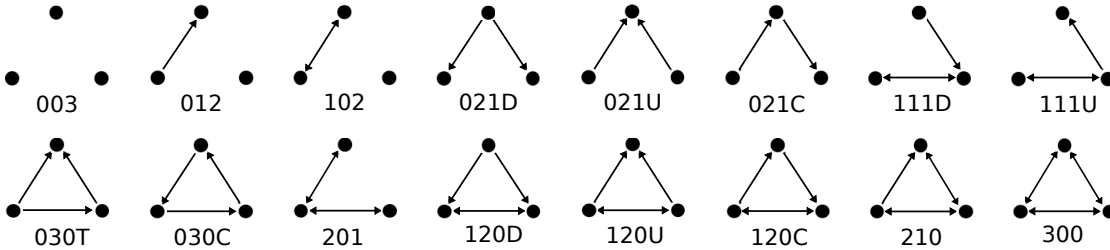


Figure 1.3 List of all possible triads and their labels in MAN notation (Holland and Leinhardt, 1970). This notation records the number of mutual (M), asymmetric, (A) and null (N) dyads in each triad, along with further indication of the direction of ties.

properties such as the degree distribution. Given a set of randomized versions of the graph, we can then easily compute the z -score and p -value of the triad in the real graph to decide whether its frequency is statistically significant or if it can be explained by the reference model.

A good null model should reproduce known properties of the graph so that the motifs that are detected are those that cannot be explained by our current knowledge. Some null models are, for instance, the rewiring of the edges that preserve the degree distribution (Wernicke and Rasche, 2006) or random graphs that keep the same blockmodel structure (Birmelé, 2012).

It is important to note that conversation trees are acyclic graphs, and therefore not all randomisation strategies can be applied. A valid strategy would be to re-build the tree sequentially: for every new vertex, choose a parent vertex uniformly from the vertices that are already in the tree. We may respect the degree distribution (e.g., Dorat et al. (2007)) or give a probability to each parent according to some thread growth model such as Gómez et al. (2012).

A major impediment for the application of triads to conversations is that there are only three possible triads in a tree (012, 021U, and 021C, see Figure 1.3) and therefore a triad analysis in a tree is less verbose than a triad analysis in an interaction graph. The obvious solution is to convert the tree into an interactions graph and do the triad analysis there. However, we would be losing the conversation structure; a chain in the interaction graph (triad 021C) does not necessarily correspond to a chain in the tree graph (see Figure 1.1) In Chapter 3, we propose some extensions for a richer analysis of tree-shaped conversations.

1.3.5 Other methods

There are other approaches that fall off the above categories. Some authors, for instance, have proposed different visualizations to spot interesting roles. Viegas and Smith (2004) designed a visualization tool to plot some aspects of authors posting behavior, and used this tool to visually identify some roles. Fisher et al. (2006) proposed using egonets

of second degree and their degree distribution. An analysis of different visualization methods and their utility to find roles can be found in [Welser et al. \(2007\)](#).

Of course, the concept of role is not limited to discussion forums. They have also been studied in Q&A forums ([Furtado et al., 2013](#)), a feature-based approach that uses features like the number of questions, the number of answers, and the scores to their questions and answers given by the community. [Maia et al. \(2008\)](#) build a social graph for the case of YouTube where edges mean subscriptions, and use features from the egonet (e.g.: in-degree, out-degree, reciprocity, and clustering coefficient) and activity features (e.g.: number of uploads, the number of watched videos) to cluster users. [Labatut et al. \(2014\)](#) define social capitalists as users trying to increase their number of followers and interactions by any means possible and identify different types of social capitalists clustering users over structural features. [Zhang et al. \(2007\)](#) classify users of a Java forum in terms of expertise with an adapted PageRank algorithm. [Welser et al. \(2011\)](#) apply their methodology presented in [Gleave et al. \(2009\)](#) to find roles in Wikipedia.

Although language is certainly an important indicator of user roles, it has been barely used for their automatic detection. In [Golder and Donath \(2004\)](#), they describe a flamer as someone who tries to trigger long debates (flame wars). For instance, a flamer in Usenet wrote this:

```
Look moms (and dads), you little babies ARE NOT cute. Sorry to
break the news, they are UGLY. So stop bringing them into the office
so that fat old bags (who couldn't if they tried) can get up off
their fat arses and say things "oh isn't that cute". YUCK.
```

Detecting this is a flamer requires knowing a cultural context where every one says babies are cute, something that is out beyond the current state of the art of Natural Language Processing. One might instead apply sentiment analysis over posts to detect the positiveness or negativeness of a user and add this feature to the role analysis. Topic analysis was used by [McCallum et al. \(2007\)](#), who mixed topic analysis and blockmodeling over the e-mails between Enron employees to detect their role (position) in the company, exploiting the fact people in the same position write about the same things to people in the same other positions. It is not clear, however, how this assumption would map into online forums where the community is already, by definition, a topic-based community.

Other authors have analyzed community-based roles in social graphs, such as bridges or hubs ([Scripps et al., 2007](#); [Chou and Suzuki, 2010](#); [Gliwa et al., 2013](#); [Henderson et al., 2012](#)) and their evolution over time ([Rossi et al., 2013](#)).

1.4 Industrial context of the thesis

This is an industrial thesis developed in Technicolor R&D France under the CIFRE program (Conventions Industrielles de Formation par la Recherche) founded by the Association Nationale de la Recherche et de la Technologie (ANRT) of the Ministère de l'Enseignement supérieur et de la Recherche.

Technicolor is a company specialized in the media and entertainment sector. Its technologies and services range from film post-production (e.g.: visual effects, color enhancement,...) to Set-Top Boxes. Like many companies in the technological sector, Technicolor is paying a growing attention to the end-user services and, in particular, to user profiling and content personalization. Forums where people talk about movies or series are of particular interest for the company. On the one hand, they are a good source of information to measure the audience reaction to a movie. On the other hand, they can be a service themselves that add value to the final content: be it a simple fan forum or forums where users are also active participants of a *transmedia* narrative.

In this context, we have filed a patent based on the method presented in Chapter 5.

1.5 Outline of the thesis

The current methods of role detection are based on analyzing the interactions graph from different points of view (blockmodels, features or triads). In this thesis, we descend to the conversational level, represented by trees of posts, to find roles based on the different ways how people structurally talk.

The content and contributions of each of the following chapters is as follows. ¹ Chapter 2 is a general description of our dataset that will allow contextualizing the results of the subsequent chapters. A major conclusion of this chapter is that our forums are not big groups of users where everyone knows everyone—which might suggest, for instance, a blockmodeling approach—but rather a majority of occasional users and a minority of active users that creates most of the content.

In Chapter 3 we present our first method for the detection of roles based on conversational structures. We apply different notions of neighborhood for posts in tree graphs (*radius-based*, *order-based*, and *time-based*) and show that our *order-based* neighborhoods, similarly to triads in other types graphs, are able to capture most of the relevant conversational structures with no need of complex patterns. We use these neighborhoods to detect groups of users that tend to participate in the same type of conversation.

In Chapter 4 we present our second method based on stochastic models of growth for conversation threads. Building upon these generative models we propose a method to find groups of users that tend to reply to the same type of posts. We show that, while we are able to find clusters of users based on their past behaviors, there is no evidence that these behaviors are predictive of future behaviors. This finding calls into question our concept of *role* as a consistent behavior of a user, at least concerning the type of conversational behavior that we consider here.

Note that, in Chapter 3, we cluster users based on feature vectors that are directly built from observed behaviors and, in Chapter 4, we assume an underlying behavioral function with latent parameters and we cluster users based on their estimated parameters. In Chapter 5 we present a third method that integrates the type of data used in the two previous chapters (feature-based and behavioral or functional-based) and that

¹Chapter 4 and Chapter 5 also include a State of Art specific to the chapter. We avoided presenting them in the current chapter in order to give them in their proper contexts.

can find clusters using fewer examples. The model exploits the idea that users with similar features have similar behaviors. Not only the method integrates inputs of different nature, adding more consistency to the clusters, but it tackles the problem of clustering users when most users have only participated a few times. We use synthetic data to show the properties of this model while leaving its application to real forums for our future research.

We conclude in Chapter 6 with a summary of the results and a discussion on the perspectives opened by this thesis.

2 Analysis of datasets

—Don't be scared ... it is me. Love you and miss you.
—Wow ... this is so cool!

First AOL Instant Message, Jan. 6, 1993

Contents

2.1	Forum dynamics	19
2.1.1	Posting activity	19
2.1.2	Newbies versus oldies	20
2.2	User dynamics	23
2.2.1	Posting	23
2.2.2	Lifespans	23
2.2.3	Communities	24
2.3	Conversation dynamics	26
2.3.1	Thread size	26
2.3.2	Thread duration	28
2.3.3	Motifs of interaction in threads	28
2.4	Summary	32

THERE are plenty of available datasets for social network analysis. However, forum datasets are mostly conceived for natural language processing, and most of them lack the metadata indicating which posts replied to which posts. In this thesis, we have explored several datasets:

- **IMDb** (<http://imdb.com>): the IMDb dataset had been internally crawled by Technicolor. It contains the discussion threads about the top 100 rated and the bottom 100 rated films in 2013. Unfortunately, lots of posts on the site have been deleted under unknown criteria. In the dataset, these posts have neither text nor parent, leaving most trees disconnected and thus difficult to work with. We reported some initial research and a discussion around it in [Lumbreras et al. \(2013\)](#).
- **Boards.ie** (<http://www.boards.ie>): *boards.ie* is an Irish forum organized around different subjects such as sports or politics. It was released by the owner company at the ICWSM 2012 conference and contains ten years of data. There is no metadata indicating which post replied to which posts, and replies must be inferred from text quotations and user mentions in the posts. This is a challenge in itself because, among other reasons, the quotation and naming conventions in the forum are too open and changing through the years. Research with this dataset includes [Kan et al. \(2013\)](#); [Angeletou et al. \(2011\)](#); [Chan et al. \(2010\)](#).

Forum	Threads	Posts	Users	Posts/user
TwoXChromosomes	120,231	3,690,732	335,740	10.9
gameofthrones	156,937	3,326,169	278,748	11.9
podemos	88,815	1,368,457	30,032	45.56
france	40,967	1,083,957	23,323	46.47
MachineLearning	11,509	114,407	14,572	7.8

Table 2.1 Datasets used in this thesis. All comments made between 2013 and 2016 in five Reddit subforums.

- **Slashdot** (<http://www.slashdot.com>): Slashdot is a technology-news forum where moderators publish news posts for users to comment on them. In this dataset the tree structure is explicit, but the first post is missing. This forum has been widely analyzed in Gómez et al. (2008); Kaltenbrunner et al. (2007); Gómez et al. (2010, 2012).
- **Reddit** (<http://www.reddit.com>): Reddit 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. 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 finally chose the Reddit dataset because it is the only one with complete trees and years of data. We chose five subforums from which we downloaded all comments between 2013 and 2016 (Table 2.1):

- **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 10 episodes.
- **france** (<http://www.reddit.com/r/france>): a francophone generalist forum where users can talk about any subject although France-related topics (culture, politics,...) are especially frequent.
- **TwoXChromosomes** (<http://www.reddit.com/r/TwoXChromosomes>): a forum for women where users talk about any women-related topic and very oriented to advice-seeking.

- **MachineLearning** (<http://www.reddit.com/r/MachineLearning>): a forum where users asks questions and share news and articles about Machine Learning.

In the following sections, we analyze these subforums at three different levels—forum level, user level conversation (thread) level—computing appropriate measures for each level.

2.1 Forum dynamics

The analysis at forum level will give us an overview of the amount of activity in every forum, their periodicity (if any) or peaks that might happen due to internal or external factors. We will also analyze at which paces new users join the forum and how the activity is distributed between older and newer users.

2.1.1 Posting activity

How do the activity levels evolve over time? We first look at the number of comments posted every day in each forum (Figure 2.1). Interestingly, they show different patterns on their activity growth, their periodicities, and their peaks.

Growth. Some forums like **france** and **MachineLearning** show a smooth increase in the average number of posts per day. **gameofthrones** is stable during the inter-season hiatus, but increases its activity levels every time a new season begins, probably thanks to an increasing popularity of the TV series. **podemos** is the forum with the most drastic initial transition, certainly coupled with its popularity and the expectations that the party raised in Spain. Its current decrease of the mean activity levels might signal either a disaffection or just that the former levels of activity were not sustainable in the long term.

Periodicity. While most forums only have weekly periods with activity decreasing during the weekends, **gameofthrones** has two different periods: the inter-season period, where no new episodes are broadcasted and the levels of activity do not change significantly, and the intra-season periods, where the broadcasting of every episode generates a new peak. As a curiosity, we noted the height of a peak is often proportional to the positive reception of the episode. For instance, the most popular episode of the third season is, according to the ratings in IMDb, the ninth —often known as the *Red Wedding*. It was broadcasted on June 2, 2013 and corresponds, indeed, to the highest peak of that season.

Spontaneous peaks. Forums can have spontaneous peaks often related to an external event. In **france**, the two main peaks correspond to the two terrorist attacks in 2015. In **podemos**, the highest peak corresponds to the foundational congress of the party. In **MachineLearning**, the first peak (May 2014) was due to an AMA session (*Ask me anything*) with Yann LeCun and the second one (April 2015) corresponds to an AMA with Andrew Ng.

Moreover, a visual analysis of forum thread lengths from Figure 2.2 shows that peaks tend to be generated by longer threads. This is because when there is some external

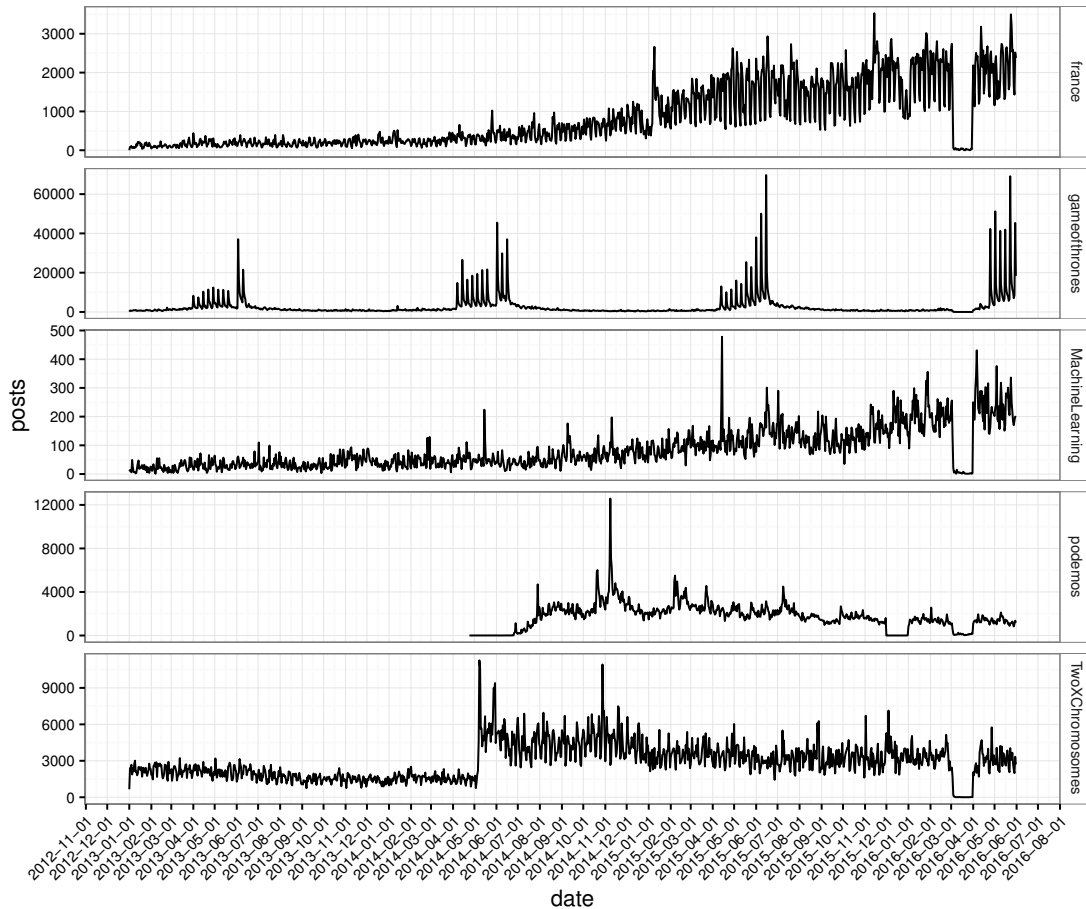


Figure 2.1 Posts by day (gaps in figures are missing data).

event relevant to the community, the discussion around it is centralized in a few long threads.

2.1.2 Newbies versus oldies

How many users join the community every month? We can think of a forum as a sort of biological community where individuals are born, live and die. From the point of view of the researcher who wants to extract types of behaviors, the longer individuals live, the easiest for any algorithm it will be to detect some patterns. Put the other way around, if all individuals posted only once, it would be harder to learn from them and especially to make a significant classification of individuals in roles. Besides, even if we ignore the minimal quantitative conditions for a community to exist, it seems obvious that it needs more than one participation per individual. In order to analyze how stable the communities are in terms of population, let us first define *birth* and *age*:

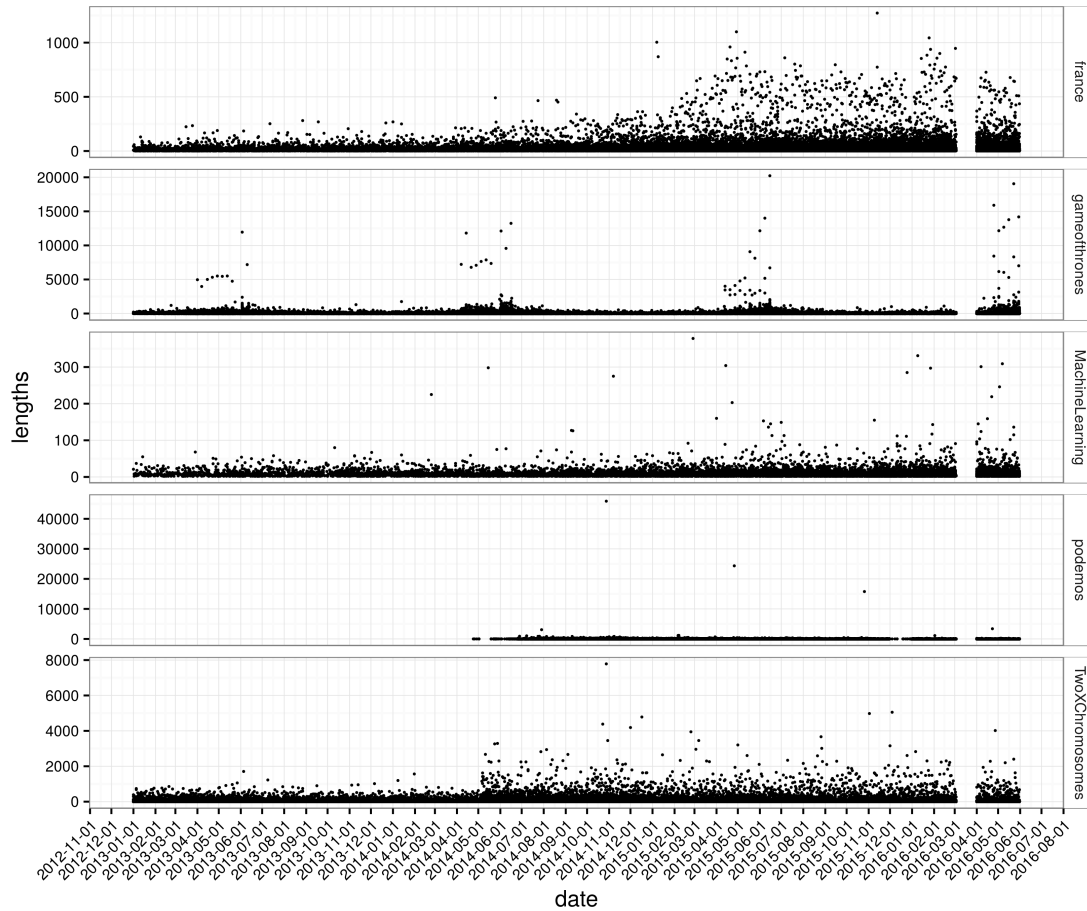


Figure 2.2 Thread lengths by day

- *Birth* (or join): a birth is the event where a user participates for the very first time in a forum. A high rate of births is an indicator of the interest of the forum towards non-users.
- *Age*: the *age* of a user at time t is the time between t and its date of birth.

For a given period, we call *newbies*, or *new users*, those users that are born in that period and *oldies*, or *old users*, those that were already born before. Figure 2.3 shows the number of users that are born every month. Interestingly, peaks in births are strongly related to peaks in the total number of posts—we will see later that newborns are not the ones who cause the peak in posts.

Who is responsible for most of the content? Despite being a minority, old users are responsible, at least, for 75% of the posts every month (Figure 2.4). *podemos* and *france* are the most extreme cases, where old users account for almost the totality of posts.

2. ANALYSIS OF DATASETS

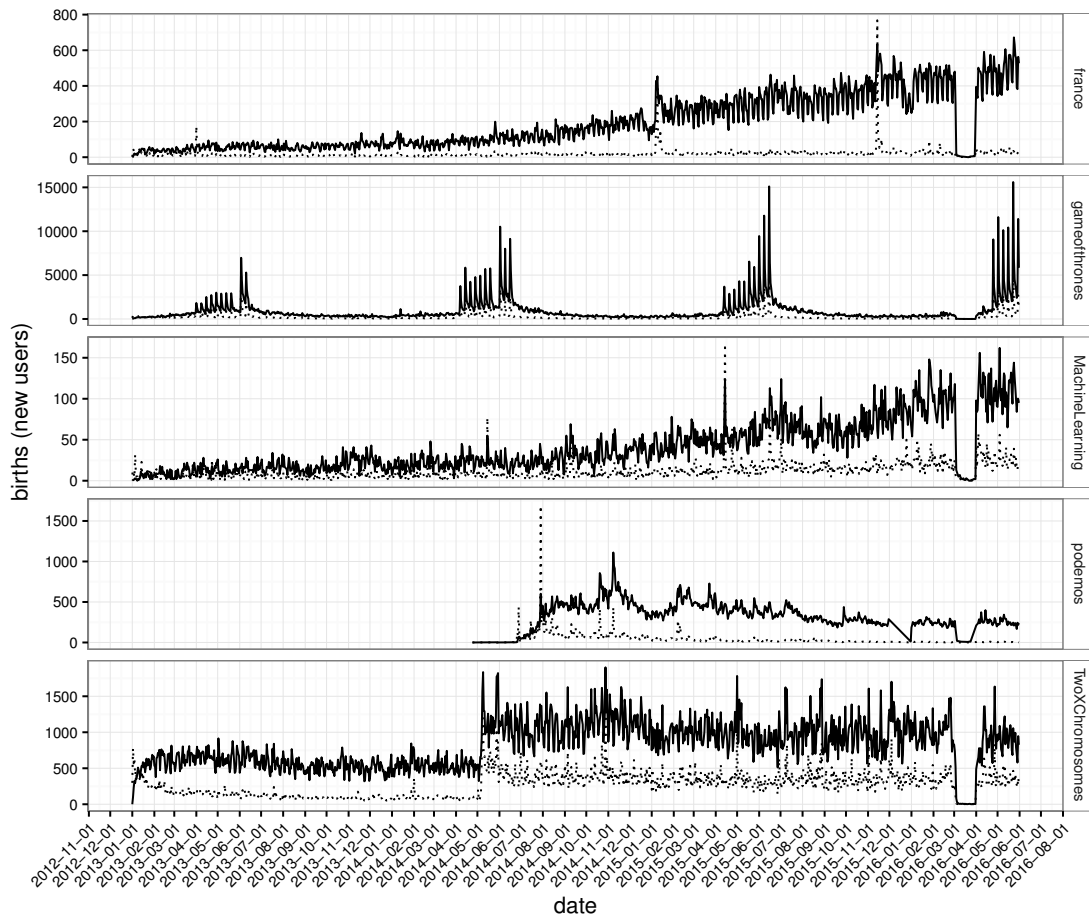


Figure 2.3 Births by day. Dotted lines are users that had already written before (*oldies*). Birth peaks are very related to peaks in number of posts.

The age distribution of posters shows a different pattern from forum to forum. Figure 2.5 shows the age distribution of the users who post each month. In *podemos*, the mean age increases almost linearly with time, meaning that the population is getting older without many new users joining the community. The population of *gameofthrones* gets older during the season break, but every time a new season begins, with the activity peaks, they enjoy a *baby-boom* that drastically decreases the median age. *france* is very stable; after the two terrorist attacks, a lot of new users participated in the discussion and then abandoned the forum, decreasing the mean age only during that month. *machine learning* and *TwoXChromosomes* enjoy the participations of older members (outliers in the plot) while new participants keep joining the forum, keeping the mean age almost constant.

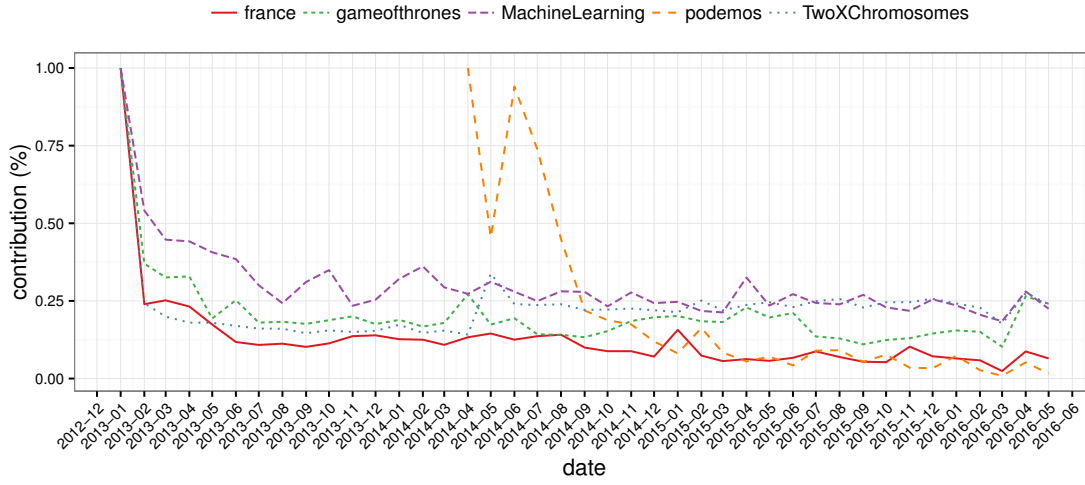


Figure 2.4 Percentage of posts written by new users each month.

Forum	# posts > 100	# posts > 1000
france	1252	180
gameofthrones	4378	68
MachineLearning	117	4
podemos	1527	224
TwoXChromosomes	3624	139

Table 2.2 Number of users with more than 100 and 1000 posts.

2.2 User dynamics

2.2.1 Posting

How big are inequalities in posting activity? As commonly observed in social networks, a minority of users creates most of the content. The number of posts per user are much closer to Log-normal (centered at 1 post) than to Power Law distributions, and in some cases, like **france**, they might be better fitted by slightly different distributions (Figure 2.6). Very few users have written more than 100 posts (Table 2.2). Though this is not surprising, this will make it very difficult to learn behavioral patterns from most users.

2.2.2 Lifespans

How long do users stay in the forums? We define as *lifespan* the number of days between the first and the last post of the user. Figure 2.7 shows that most users are

2. ANALYSIS OF DATASETS

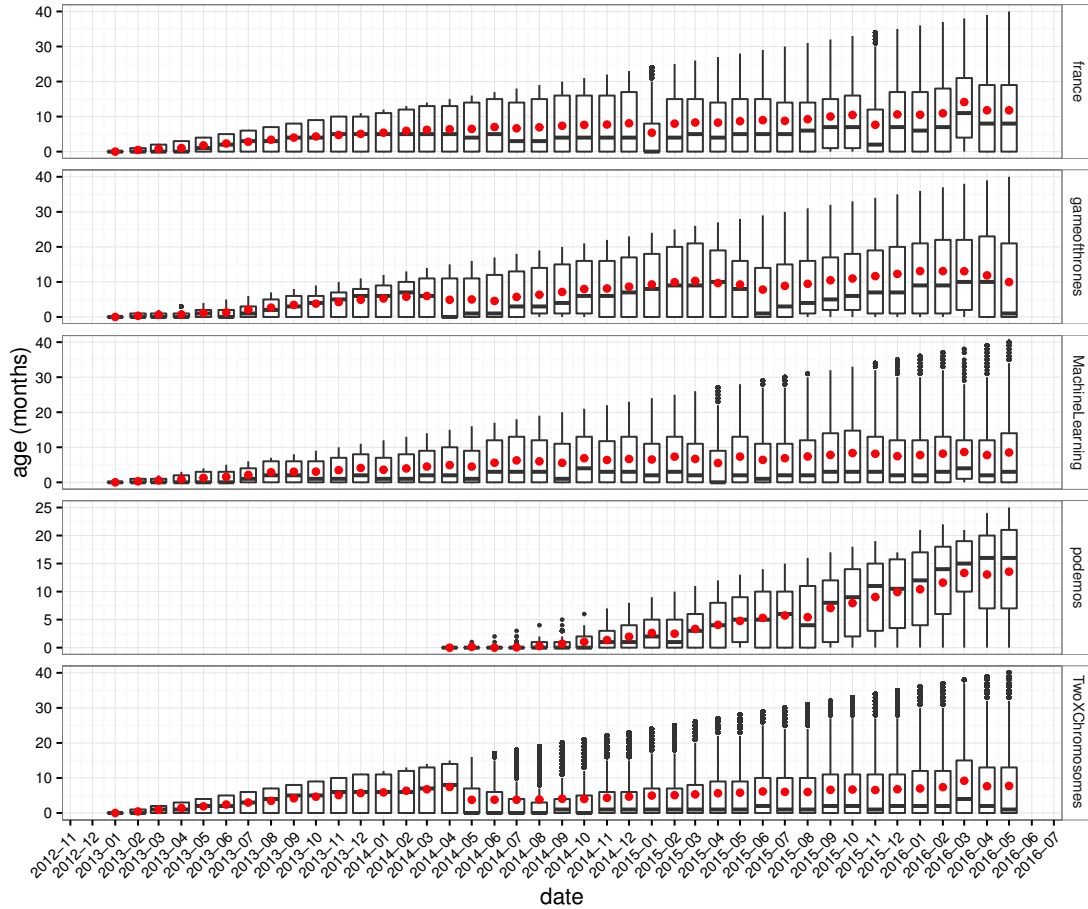


Figure 2.5 Distribution of users age by month (boxplots). Means marked by red circles.

casual participants, posting once and then abandoning the forum forever (they may also become silent readers, which is known as *lurking*). Only around 25% of users live more than a year. Lifespans have no important differences between forums, except for `gameofthrones`, where we can observe the fact that every year a new cohort joins the forum.

2.2.3 Communities

How big are user communities? Given a graph, a community is a set of vertices sharing with a high density of links between them and with sparser connectivity with the other communities. We consider communities in the coparticipation graph, where a link between two users indicates at least one coparticipation in the same thread. In the extreme case where the graph is made of dense communities with few edges between communities (high *modularity*), it would mean that the forum is actually a collection

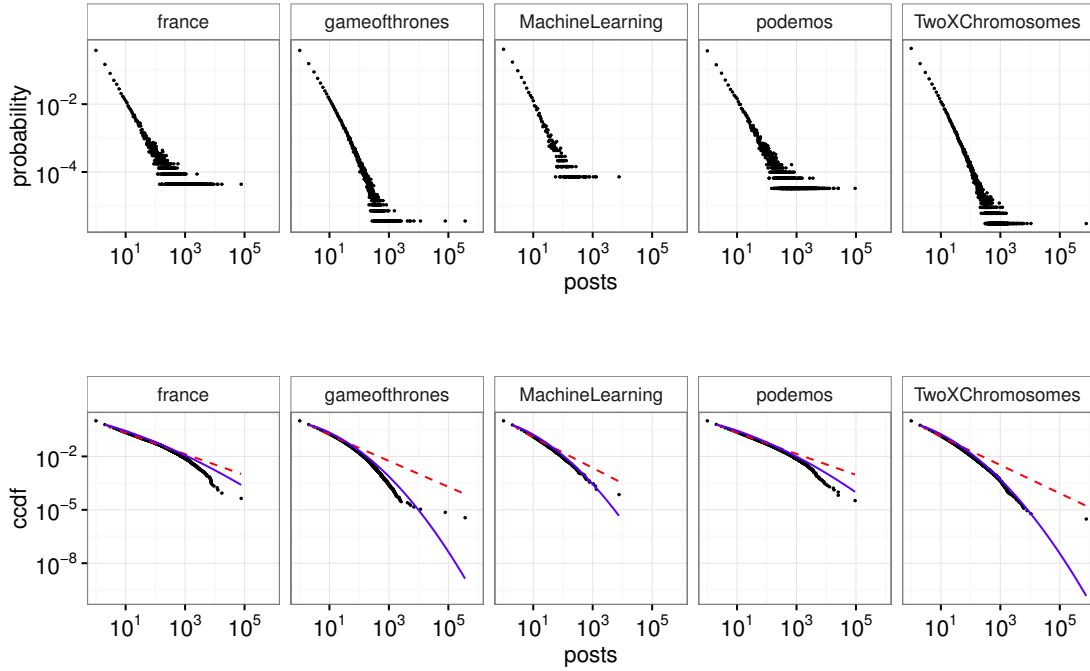


Figure 2.6 User activity. (*Top*) Distribution of number of posts. (*Bottom*) Complementary Cumulative Distribution with MLE fits of Power Law (dashed) and Log-normal (solid) distributions.

of subforums with their own community of users. The modularity of a partition is a scalar value between -1 and 1 that measures the density of links inside communities as compared to links between communities.

It seems reasonable to think that the denser a community is, the easier behaviors and—any cultural artifact such as norms, writing style and so on—are propagated between its members.

We build a coparticipation graph for each month and detect its communities with the Louvain algorithm (Blondel et al., 2008). Figure 2.8(a) shows the relative sizes of the top five communities. In *france* and *podemos*, the five biggest communities always contain nearly all the users. On the other hand, the five biggest communities in *MachineLearning* usually contain just a bit more than a half of the users. The modularity (Figure 2.8(b)) shows, indeed, that *MachineLearning* is the most modular forum. This suggests that *MachineLearning* has more subtopics and that users tend to concentrate on one or another topic according to their personal (maybe professional) profile and their interests.

Lastly, Figure 2.8(c) shows the community sizes in the aggregated coparticipations graph—a graph where we include all the coparticipations regardless of their month. The largest 10 communities for all three forums have more than 1000 members. In *Ma-*

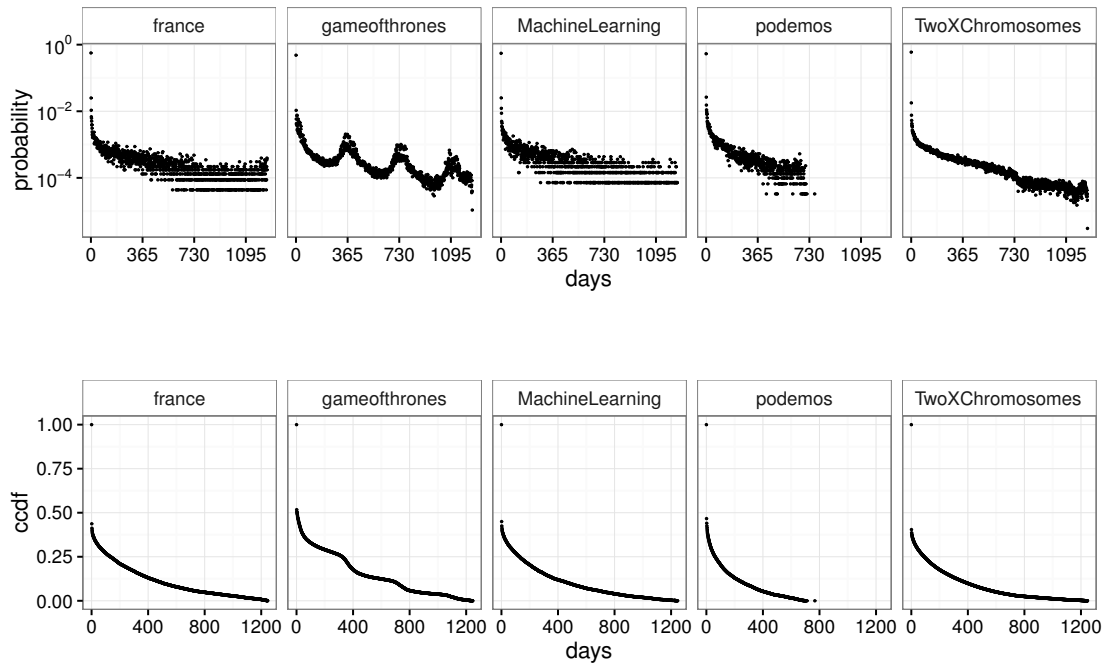


Figure 2.7 User lifespans (time between first and last post). (*Top*) Distribution of lifespans. (*Bottom*) Complementary Cumulative Distribution.

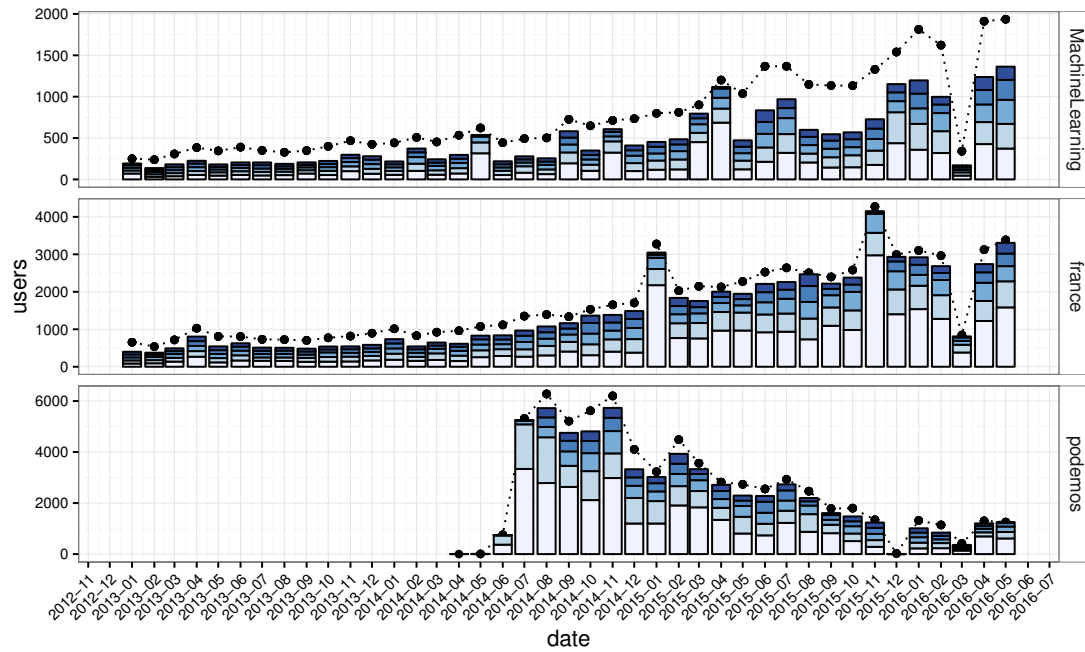
chineLearning, the size of the largest communities is actually more equally distributed.

2.3 Conversation dynamics

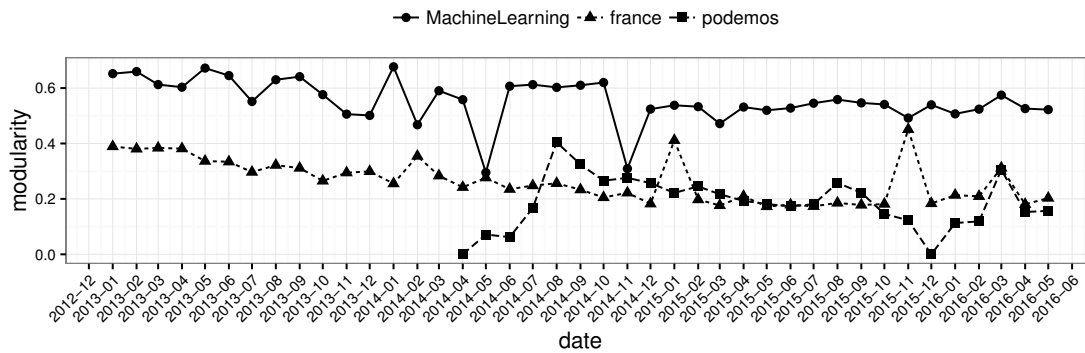
2.3.1 Thread size

How large are threads? The distribution of thread sizes may be affected by factors such as the type of forum and the user interface. Nevertheless, most forums share common traits regarding thread sizes such as a high probability of small threads and a long tail of long threads. In [Gómez et al. \(2012\)](#), the authors report different distributions between Wikipedia and some news-oriented forums such as Slashdot. While Slashdot showed a distribution with some scale, Wikipedia has a scale-free distribution. Even if Reddit is more similar to a news-oriented forum like Slashdot, the length distributions of analyzed Reddit forums are more similar to Wikipedia. Yet, there distributions are more similar to Log-normals (Figure 2.9).

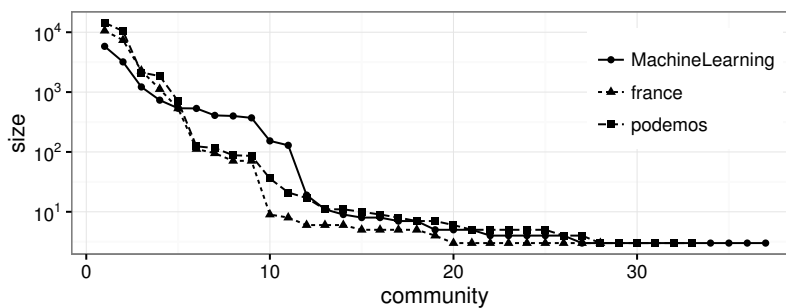
2.3. Conversation dynamics



(a) Size of coparticipation communities. The five biggest communities are shown in the bars. The points indicate the total number of users. Spaces between bars and points correspond to users in smaller communities.



(b) Evolution of modularity over time.



(c) Size of coparticipation communities when all participations are aggregated to the same graph regardless of their time.

Figure 2.8 Coparticipation communities for three of the forums.

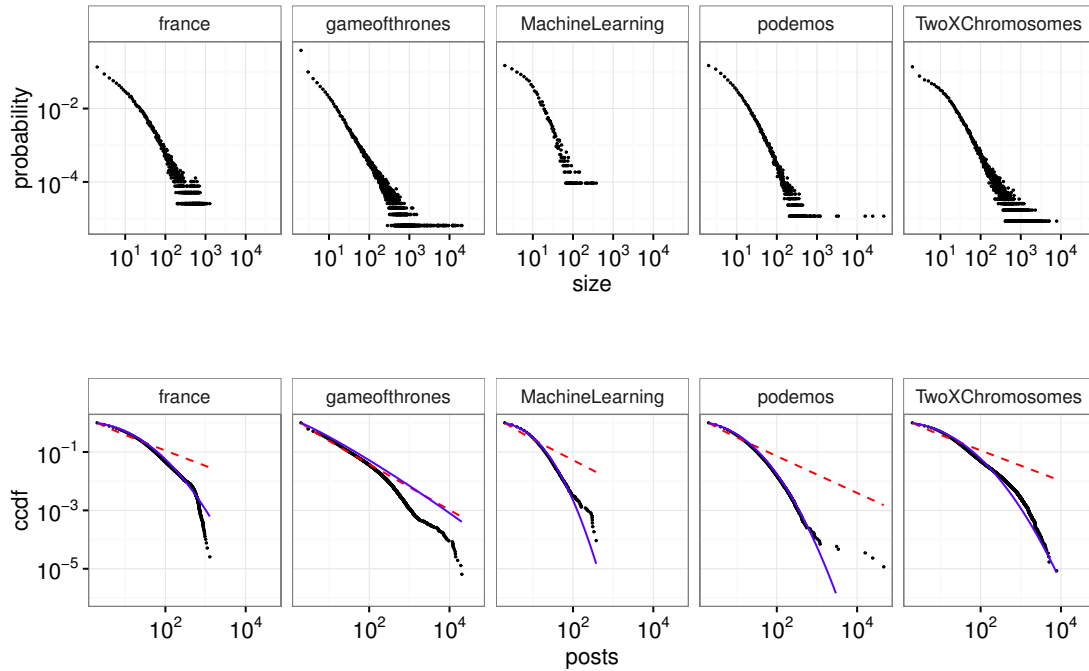


Figure 2.9 (*Top*) Threads size distributions. (*Bottom*) Complementary Cumulative Distribution with MLE fits of Power Law (dashed) and Log-normal (solid) distributions.

2.3.2 Thread duration

How long do conversations last? Users in a forum have many threads that they can join, and new threads are created every day. Thus, most threads grow during a first period, and this growth stops some time later. We illustrate this in Figure 2.10 where we plot the time between each new post and the start of the thread (*time to the root*). Only around 15% of posts arrive in the first hour. The main growth of thread is after the first hour and during the first day, when most posts arrive. Almost no posts are written after the first day.

2.3.3 Motifs of interaction in threads

What are the dominant micro-structures of the threads? A *triad census* counts how many times each of the 16 triads appear in a graph. In this section we show the triad census using the interaction graph and the tree graph of a thread. We will see that triads in interaction graph are ambiguous while in the tree graph there are also two possible triads. This will actually motivate the extension of triads in trees that we propose in Chapter 3.

Triads in interaction graphs. For every thread, we build an interaction graph

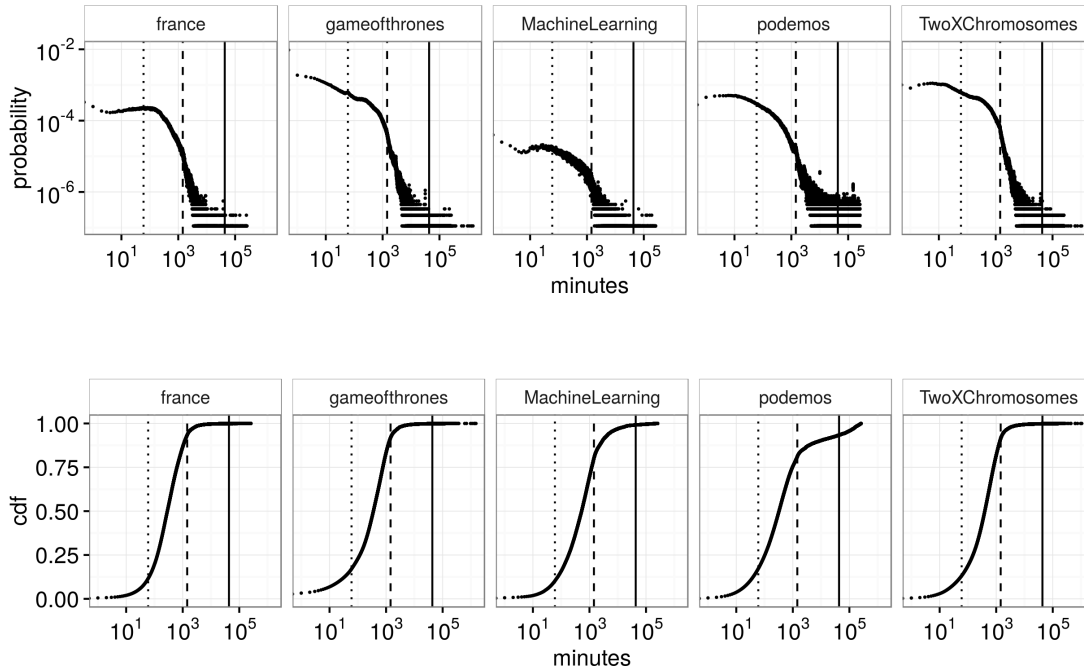


Figure 2.10 Time from posts to root. Vertical lines mark the first hour, the first day and the first 30 days. Most posts arrive between the first hour and the first day. Conversations are practically over after the first day.

(recall Figure 1.1) and we compute its triad census. Figure 2.11 shows the sum of census for each month. Although the relative dominance of the triads is different among the forums, it is very constant for each forum. That means that each forum has a different type of conversation and that the type of conversation in a forum does not change much. In general, we see a dominance of the *in-star* (021U) and the dyad (012), which is very likely due to the root posts that attract single or multiple replies. The triad 111D is as frequent as the dyad in *MachineLearning* and *podemos*. Chains (021C) are specially important in *MachineLearning*

Unfortunately, the interpretation of triads in the interactions graph is ambiguous. We do not know, for instance, if a vertex is in a lot of *in-star* because the user tends to get replies from different people or because he got replies from different people in *one* post.

Triads in tree graphs. Unlike triads in the interactions graph, triads over the tree graphs are unambiguous. There are four possible triads in a tree: the empty triad (003), the *directed edge* (012) the *in-star* (021U), and the *chain* (021C). The empty triad is not interesting and the *directed edge* is trivial in a tree. Thus, we focus on the *in-star* and the *chain*. To know how significant the frequency of a triad is, we need to compare

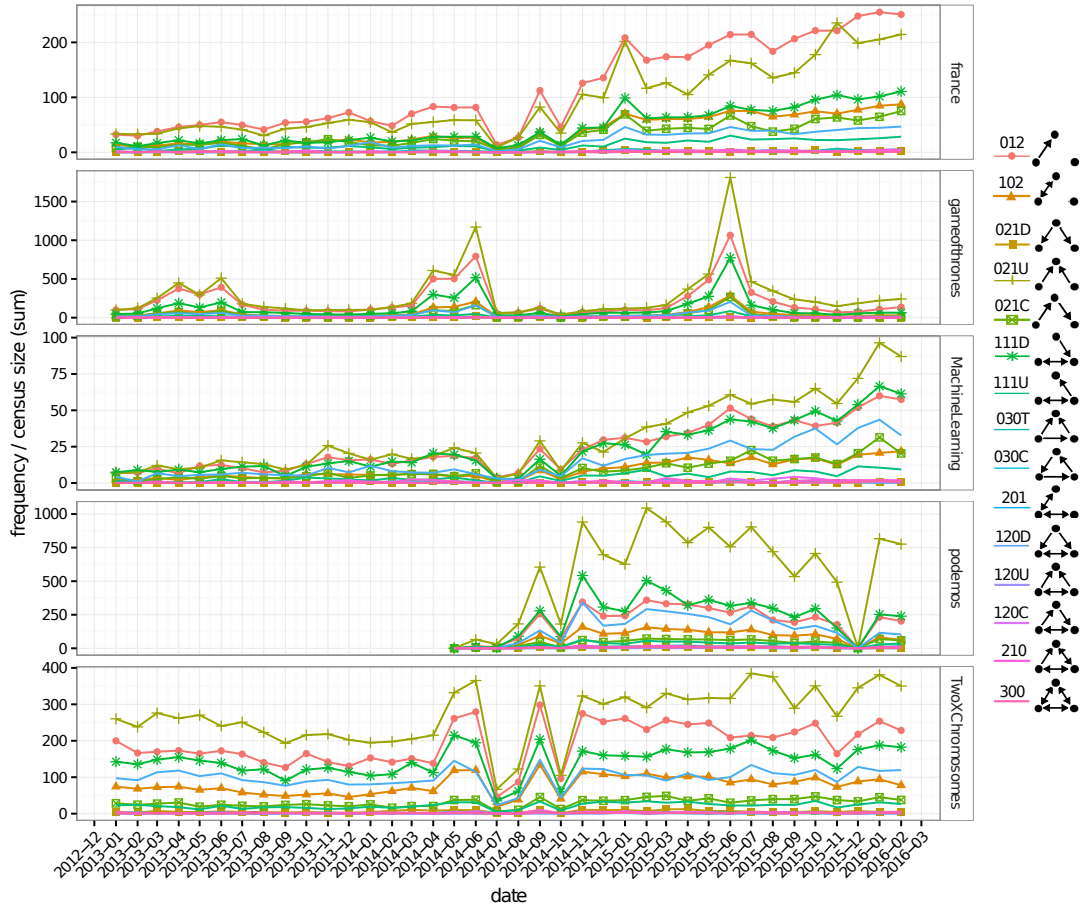


Figure 2.11 Evolution of triad census over time

it to its frequency in a null model. Our null model is a random tree that grows by new vertices choosing a random parent among the existing vertices; this is actually the most entropic way of building a tree (assuming that the tree has a fixed number of vertices that are added one by one).

We compute the z -score of a triad in a tree graph g of size $|V(g)|$ as follows. With our null model, we generate 100 random trees of size $|V(g)|$. Let f_g be the frequency of the triad in g . Let μ_r and σ_r be the mean and the standard deviation of that triad in the set of random trees. The z -score is

$$z = \frac{f_g - \mu_r}{\sigma_r} \tag{2.1}$$

Triads with z -score near zero are not relevant because they have the same frequency as in random graphs. On the other hand, triads with higher (or lower) z -scores are more (or less) frequent than in random graphs, which means that the null model cannot

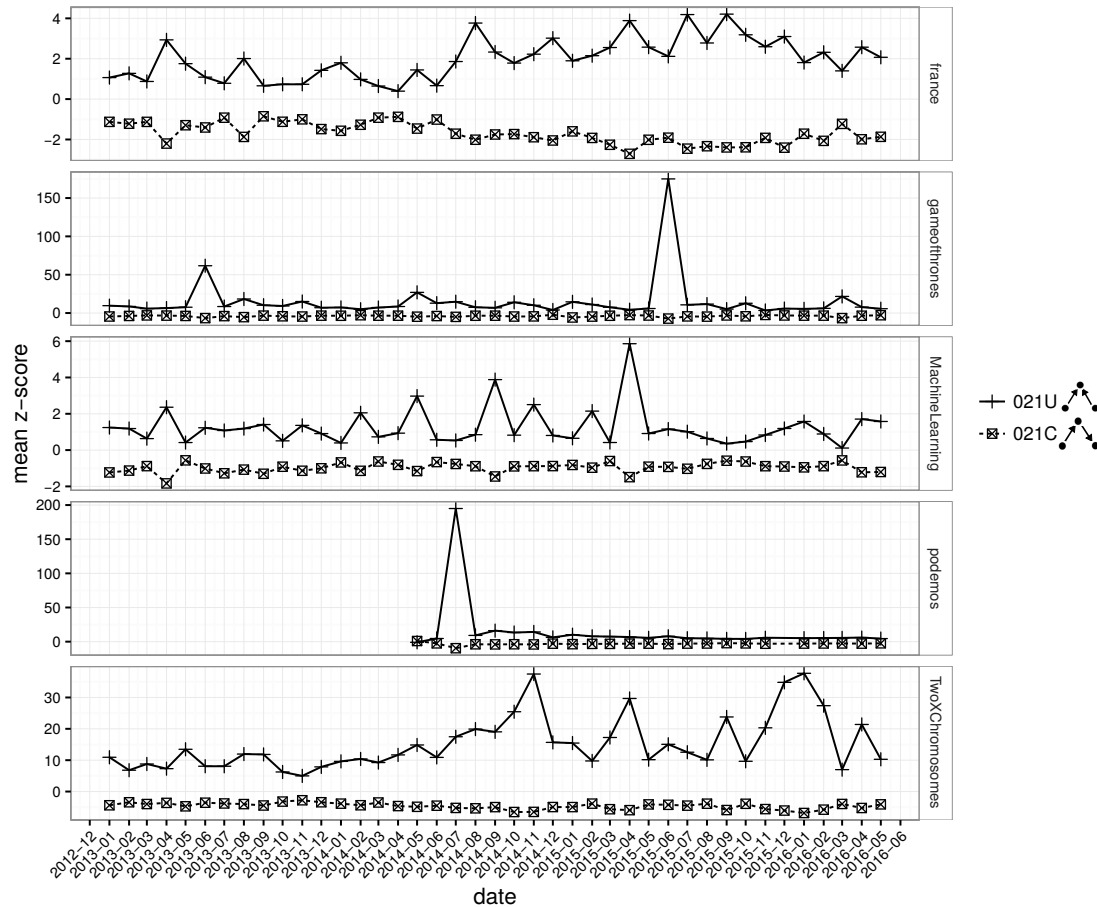


Figure 2.12 Triads motifs in the tree graphs. z-score with respect to the expected frequency in a random tree where all posts have the same probability to be chosen as parent.

justify their high (or low) frequency. Figure 2.12 shows the evolution of the triad census in our forums. The *in-star* is usually higher than in the random graphs, especially in the activity peaks with long threads. That means that long threads are made of posts that attract many replies rather than long chains of posts. In other words, long threads are wide rather than deep. `MachineLearning` and `france` have the slowest frequency of *in-stars*.

When compared to the original set of 16 triads, a set of two triads looks like a very limited tool to analyze conversational structures. We will come back to this point in Chapter 3, where we will attempt to classify users by the discussions where they participate—the motifs where they appear.

2.4 Summary

In this chapter, we have introduced our datasets and a descriptive analysis to understand some general characteristics of the forums under study. Some of the most relevant findings are:

Forums: In the short term, forums are very regular regarding the number of posts per day. Yet, they often have increasing or decreasing long-term tendencies that may be related to the health of the forum. A decreasing tendency in `podemos`, for example, is also linked to a decreasing in the number of users, which signals that users lost their interest in the forum.

External events related to the topic of a forum create peaks in the number posts—concentrated in a few threads—as well as the number of users. Some external triggers that can be observed in our datasets are the two terrorist attacks in France in 2015 (January and November), the broadcasting of a TV show (`gameofthrones`) or even a popular episode, the initial congress of a political party (`podemos`) or *Ask me anything* sessions where some celebrity is invited to answer users questions (`MachineLearning`).

This activity peaks also attracts lots of new users. In general, and despite being a majority, new users create less than 25% of the content. All forums have a core of users that have an ongoing presence and make the most of the content, and a much bigger periphery of users who only participate once and then just *lurk* or abandon the forum.

Users: As commonly observed in social networks, a minority of users creates most of the content. The distribution of number of posts per user is better approximated by Log-normal distribution centered at 1 rather than by a power law. The number of users with more than 100 posts is less than 5000 for all the forums in our dataset. Regarding lifespan, the most frequent case is a user that participates one day or two and then leaves the forum.

Conversations: Conversation lengths do not follow power laws either, but rather Log-normal distributions centered at 1. Threads receive most of the comments during the first day, probably due to the competition between threads and the daily creation of new ones. Structurally, thread conversations have a strong tendency towards the in-star, meaning that conversations are more dominated by stars (i.e.: replies to the root) than by cascades (chained replies).

In the beginning, we considered roles as community-based. A role would be defined by the community and not by the individual, and therefore a new individual would just fill in a role after some initial integration. We think this may happen for closer communities or task-oriented communities such as Wikipedia. However, the dimensions of the forums make them more similar to cities than task-oriented communities, and the role study needs to take another approach. The fact that there is only a small elite of active users makes the community-based approach especially hard. The approach that we take in the following chapters is to look at a role from the point of view of the individual behavior.

3 *Role detection based on conversation structures*

It is impossible to speak in such a way that you cannot be misunderstood

Karl Popper

In this chapter, we present three definitions of neighborhood in conversation trees and a simple algorithm to extract post neighborhoods in a forum. We describe users by their number of participations in each type of neighborhood (motif) and we find clusters of users that tend to occupy the same position in the same type of conversation.

Contents

3.1	Discussion trees	34
3.2	Methodology	35
3.2.1	Neighborhood extraction	35
3.2.2	Clustering	37
3.3	Radius-based neighborhoods	37
3.3.1	Results for radius-based neighborhood	39
3.4	Order-based neighborhoods	47
3.4.1	Pruning and coloring	47
3.4.2	Results for order-based neighborhood	48
3.5	Time-based neighborhoods	51
3.5.1	Choice of radius	54
3.5.2	Results for time-based neighborhood	54
3.6	Comparative analysis	57
3.7	Summary	58

USER behavior in online forums is often analyzed through metrics such as the number of posts they write, the number of threads, the number of replies per post or their centrality in the social graph. These features allow us to detect some important roles. A *celebrity*, for instance, defined as a *prolific poster who spends a great deal of time and energy contributing to their community and whom everybody knows*, can be discovered by their high in-degree and out-degree—among other features (Forestier et al., 2012). This way of summarizing users activity by a vector of metrics is, however, unable to capture user behaviors at a conversational level. Unfortunately, there is a lack of tools to analyze

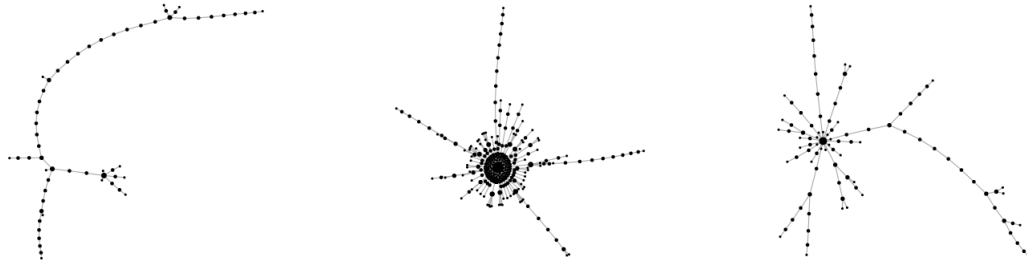


Figure 3.1 Trees with different local conversation structures.

conversational styles. In this chapter, we study the suitability of graph motifs to detect different types of conversationalists.

We represent a thread as a tree of posts where each post has an author. Threads are usually made of a variety of *local conversations* between different users; as shown in Figure 3.1, this diversity is reflected in the tree representations where some parts may have long branches (e.g.: debates or flame wars) while others may have vertices with lots of replies. (e.g.: a controversial post that gets the consensual disapproval from the other users or a good question that gets many answers). The *neighborhood* of a post is an induced subgraph that contains the local part of the tree surrounding the post. Thus, the neighborhood captures the structure of the local conversation where the post is embedded. As we will see in this chapter, there is no “natural” definition of neighborhood in the context of conversation trees. We will propose three alternative definitions, and we will analyze their strengths and weaknesses.

We recall that our goal is to detect user roles. In this chapter, we propose the following definition of role:

Definition Two users have the same role if they tend to participate in the same positions of the same type of neighborhoods (motif).

Hence, We will cluster users according to the type of neighborhood where they tend to appear. We will introduce and analyze three new definitions of neighborhoods that we name *radius-based*, *order-based* and *time-based* neighborhoods. Our strategy to detect different types of conversationalists is as follows: first, we extract the neighborhood of each user’s post; second, we cluster users according to their tendency to appear in each type of neighborhood.

3.1 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 tell us 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 distinct vertices v_i and v_k are *siblings* if and only if $p(v_i) = p(v_k)$. A vertex v_l is an ancestor of a vertex v_i and v_i is descendant of v_l if and only if v_l is in the path from v_i to the root. A *leaf* is a vertex with no descendants (a post with no replies). A *branch* is the path between a leaf and the root.

3.2 Methodology

To cluster users based on the structure of conversations where they participate, we follow two steps. First, a *neighborhood extraction* step where we detect the structure of the conversation around each post. Second, a *clustering* where we find groups of users that tend to participate (i.e.: their posts tend to be embedded) in the same type of neighborhood.

For the experiments, we analyze the 1,000 most active users of **france**, **podemos** and **gameofthrones**. We take a sample of 100 of posts for each user.

3.2.1 Neighborhood extraction

The first step consists on detecting the type of neighborhood that surrounds each post. We will call *motif* or *neighborhood motif* the set of neighborhoods that (after the extraction process) are isomorphic to each other. The general algorithm of neighborhood extraction (or motif extraction) is as follows: we initialize an empty dictionary to store pairs of motif descriptions and numerical identifiers. We will also initialize a table to where we will note, for each post, the identifier of its neighborhood motif. For every post, we extract its neighborhood and check whether it is isomorphic to any of the motifs that have been previously found. If there is a coincidence, we update the table by writing, for that post, the identifier of its neighborhood motif. Else, we add a new entry in the dictionary with the new motif and a new identifier, and then we update the table of posts. In some cases, a user will be seen in the same neighborhood more than once. It will happen, for instance, in a thread of length two where all posts are written by the same user. To avoid counting the user in the same neighborhood twice, we compute the hash of each neighborhood and remove duplicates at the end. The algorithm is detailed

in Algorithm 1.

Algorithm 1: Neighborhood extraction

Data: Set of trees $\mathcal{G} = \{G_1, \dots, G_n\}$
Result: Table *posts*; Dictionary *motifs*.

```

1 Dictionary motifs  $\leftarrow \emptyset$ ;
2 Table post_motif  $\leftarrow \emptyset$ ;
3 for  $G \in \mathcal{G}$  do
4   for  $v \in G$  do
5     # Extract post neighborhood
6      $N_v \leftarrow \text{extract\_neighborhood}(v)$ ;
7      $N'_v \leftarrow \text{color\_neighborhood}(N_v)$ ;
8      $N''_v \leftarrow \text{prune\_neighborhood}(N'_v)$ ;
9      $hn \leftarrow \text{hash}(N_v)$ ;
10    # Compare to motifs in dictionary
11    for  $i \in 1, \dots, \text{size}(\text{motifs})$  do
12      if  $\text{is\_isomorphic}(N''_v, \text{motifs}[i])$  then
13         $\text{post\_motif}[v] \leftarrow (i, hn)$ ;
14        break;
15      end
16    end
17    # If new, add motif to dictionary
18    if  $N''_v \notin \text{motifs}$  then
19       $\text{motifs}[\text{size}(\text{motifs})+1] \leftarrow N''_v$ ;
20       $\text{post\_motif}[v] \leftarrow (\text{size}(\text{motifs})+1, hn)$ ;
21    end
22  end
23 end
24 # Duplicated hashes have the same hash.
25 # Leave one entry per hash
26  $\text{post\_motif} \leftarrow \text{remove\_duplicated\_hash}(\text{post\_motif})$ ;

```

The slowest part is the search in the dictionary since we have to check whether the current neighborhood is isomorphic to any of the motifs. Specifically, the complexity of the algorithm is $O(nm)$ where n is the number of posts, and m is the number of different motifs. In practice, we can reduce the number of operations if, when looking for an isomorphic motif in the dictionary, we do it following their frequency.

The function *extract_neighborhood* correspond to any of the three neighborhood definitions that we will describe in sections 3.3, 3.4 and 3.5. The functions *color_neighborhood* and *prune_neighborhood* will be described in Section 3.3.

3.2.2 Clustering

In the first step, we obtained the motif associated to the neighborhood that surrounds each post (Algorithm 1). Now we create, for each user u , a feature vector of counts $\mathbf{n}_u = (n_1, \dots, n_m)$ where m is the number of motifs and n_i is the number of times that a post written by u has a neighborhood of class i . We normalize each feature vector so that the sum of its components is one, hence obtaining a discrete probability distribution. We compute the distance between two users as the Pearson correlation of their vectors, and finally, we apply a hierarchical clustering over this distance matrix¹.

Feature selection. After normalizing the feature vectors—obtaining probability vectors—, we remove some of the features as follows. First, we sort the vector entries by *median probability*: we compute the medians of the probability vectors \mathbf{n}_u over all the users, and we sort the entries of the feature vectors so that the first entry has the highest median—over all users—and the last entry has the lowest median. We look for an elbow in the plot—it usually lays before the 20th motif. Then we check the user outliers for each motif that falls beyond the elbow. Outliers indicate that a motif might still be important for some users. Thus, if the number of user outliers is less than 10% of users, we mark the motif as irrelevant—that is intended to keep those features that might be relevant for more than 10% of users. Finally, we merge all the irrelevant motifs into a new category *others* so that the feature vectors are still probabilities that sum up to one. Outliers are detected using Tukey’s Method: they are defined as those who lay below $Q_1 - 1.5IQR$ or above $Q_3 + 1.5IQR$, where Q_1 and Q_3 are the first and the third quartiles, and IQR (that stands for Interquartile Range) is $IQR = Q_3 - Q_1$. We will show the elbow and the outliers for every experiment (see, for instance, Figure 3.2(b))

Number of clusters. Since our main goal is to compare different definitions of neighborhood in conversation trees, we apply the same simple criteria to all the cases: for each type of neighborhood and given the dendrogram of the hierarchical clustering obtained in our forums, we cut all dendrograms at different cutting distances h and we choose the distance that gives a more meaningful—explainable—separation. Because we have the same number of users with the same number of posts for each of the forums, fixing h allows comparing the number—and the content—of clusters that we find in each forum using the same levels of granularity.

When we will talk about dictionaries of features, we will say that the most *popular* features are those with higher medians. In the context of each cluster, we will say that the most *dominant* triads are those with higher means.

3.3 Radius-based neighborhoods

Our first type of neighborhood is based on the classical definition of neighborhood in graph theory:

¹We use Ward linkage, although it is possible to use other distances and clustering methods.

Definition Given a tree graph G , the *radius-based neighborhood* with radius r of post $v \in G$, denoted as $\mathcal{N}_v(r)$, is the induced graph of G that contains all vertices at distance equal or less than r from post v .

According to this definition, the number of possible neighborhoods with a radius r is infinite. This poses a problem when trying to categorize conversations since many conversations, while structurally different, can be considered semantically similar (e.g.: we might not want to put in different categories two structures representing, respectively, a post with 50 replies and another with 40). One way of fixing this is to prune the neighborhood to remove those parts that we consider non-informative.

Besides, two isomorphic neighborhoods may correspond to different types of conversation: the top vertex of an *in-stars*, for example, may be either a regular post or the root of the thread. While the former might indicate that the post is controversial, the latter is more often related to a good opening question. We will reduce this ambiguity by assigning colors to vertices.

Coloring

Even if the structure contains some important information about the type of conversation, there is still some ambiguity left: two similar neighborhoods can represent very different types of conversation. We can easily reduce this ambiguity by assigning colors to vertices, which allows us to identify some relevant property of the post.

In particular, we assign to each post one of these three colors (see, for instance, Figure 3.2(a)):

- *Red*: ego post and posts written by the same author as ego. It allows identifying re-entries (when the same author participates several times in the discussion [Backstrom et al. \(2013\)](#))
- *White*: root post. Differentiating the root post from the rest has been proven useful by previous research on online discussions. For instance, some types of users seem to get more replies than others when they initiate a thread ([Himmelboim et al. \(2009\)](#); [Lumbreras et al. \(2013\)](#)). Also, concerning preferential attachment, root posts usually get more replies than the non-roots ([Gómez et al. \(2010, 2012\)](#)).
- *Grey*: root post if written by the author as ego (i.e.: gray are red-white vertices).
- *Black*: none of the above.

It is possible to apply other coloring schemes. For instance, we might give a different color to the leafs, or we might color posts according to their content length, or they positive or negative sentiment. To detect discussions between two users, we have also tried giving another color to those posts written by the user to which ego replies. Nonetheless, the more colors we add, the more diverse and large the list of motifs that we will find, which may be an undesirable effect. Hence, we decided to keep only three colors after playing with other schemes.

Pruning

At this point, we can still find structures the only difference of which is that one of them has more leafs hanging from some of the nodes. Thus, we prune every neighborhood by leaving a maximum of two consecutive siblings of the same color and where neither of them has 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. Moreover, this choice encourages smaller structures, which allows the computation of isomorphisms (Section 3.2.1) even in real time.

In the experiments, we will see that some of the found motifs could be merged into the same one since they represent, without much doubt, the same type of conversation. That means that there is still room for a more aggressive pruning. Yet, we have decided to stay with the current pruning in order not to further distort the comparison between neighborhood definitions.

Choice of radius

We have to be careful when choosing a radius. With a small radius $r = 1$ we cannot include the siblings and grandchildren of the ego post and we will only detect the parent and the children (direct replies). Thus, we will lose structures like *ego replies to a post that has more replies* or like *ego writes a post, someone replies and then ego replies back*; all these structures need a radius $r \geq 2$. With larger radius we will find two problems: first, even with the proposed pruning the dictionary of motifs will become larger, and the number of motifs with almost zero frequencies will increase; second, the motifs will be bigger, and this will increase the computational time to check the isomorphisms. As we will see in the experiments, a good middle ground is $r = 2$, which is the minimum radius necessary to include the grandchildren and the siblings of the ego post.

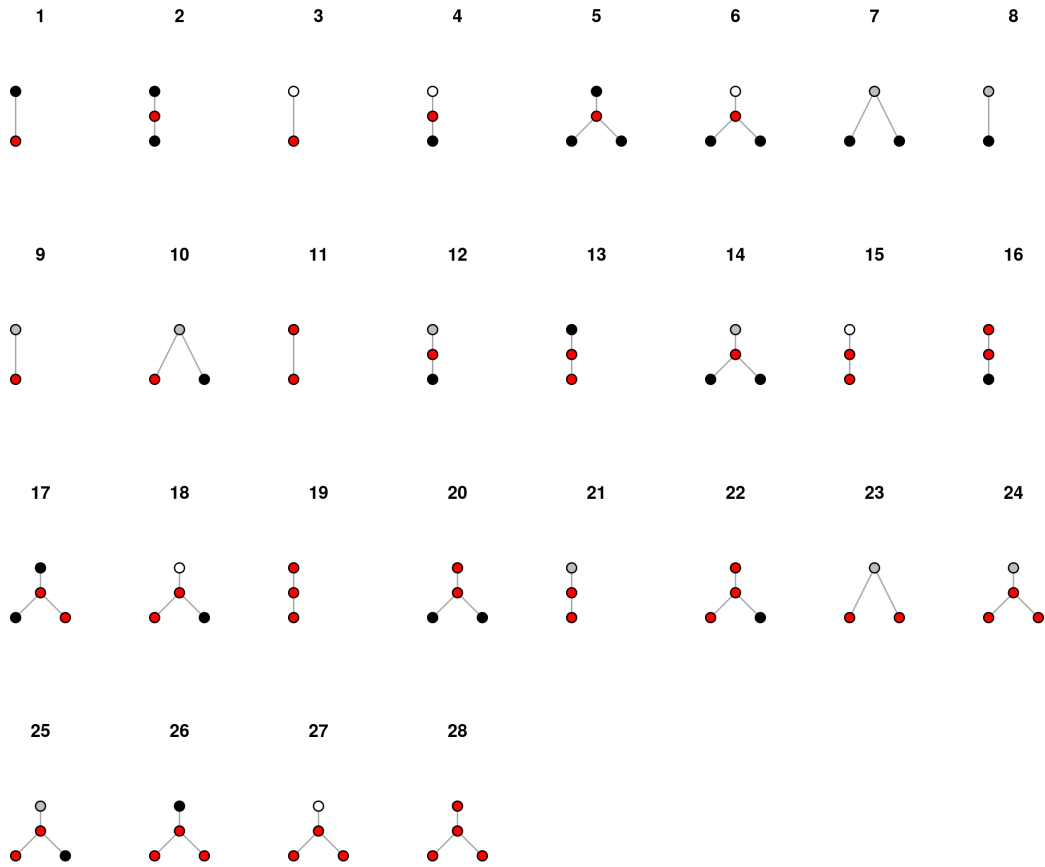
3.3.1 Results for radius-based neighborhood

We extracted the radius-based neighborhood around each post written by our selected users. We tried radiuses $r = 1$ and $r = 2$. For radius $r = 1$. Figure 3.2 shows the most frequent motifs and their frequency in every forum. Figure 3.4 shows the most frequent motifs for radius two.

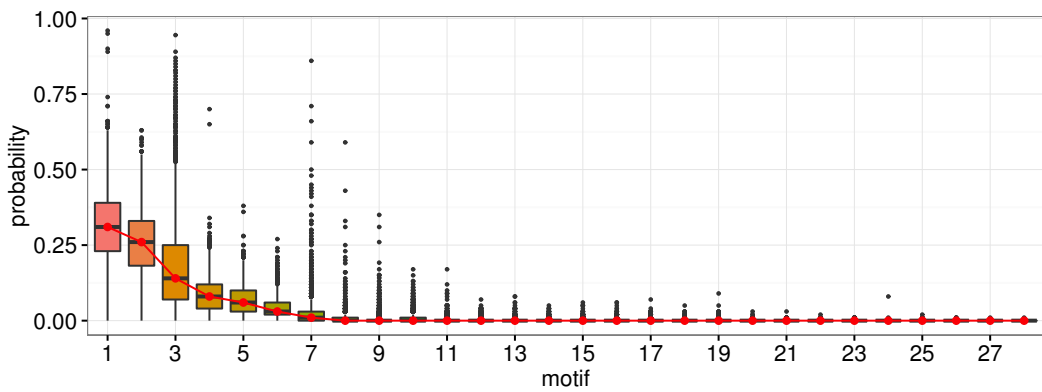
Results for $r = 1$

Dictionaries. We found 28 different motifs in the analyzed forums (Figure 3.2). The motifs dictionary gives us already an interesting perspective of what we can see within each radius. Note that only four of them would be possible without coloring. The most popular triads are replies, replies to root posts and root posts with replies. The more we go into less popular triads, the stranger the patterns that we observe. A common characteristic of the less popular motifs is that they represent conversations where the

3. ROLE DETECTION BASED ON CONVERSATION STRUCTURES

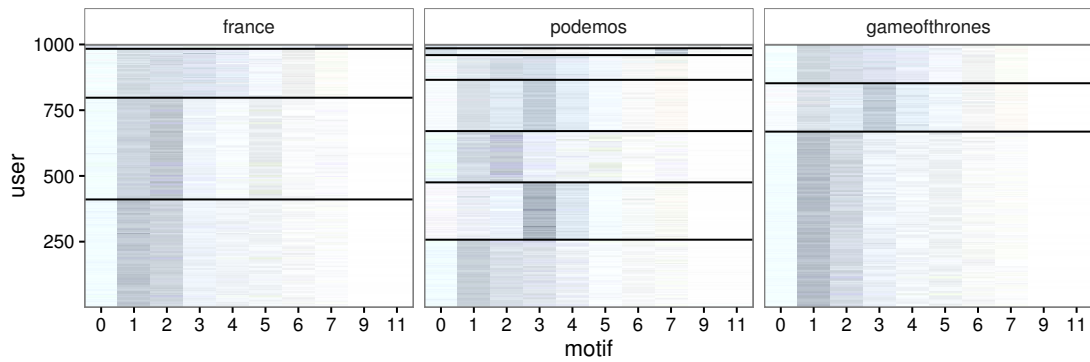


(a) Dictionary of motifs sorted by median probability.

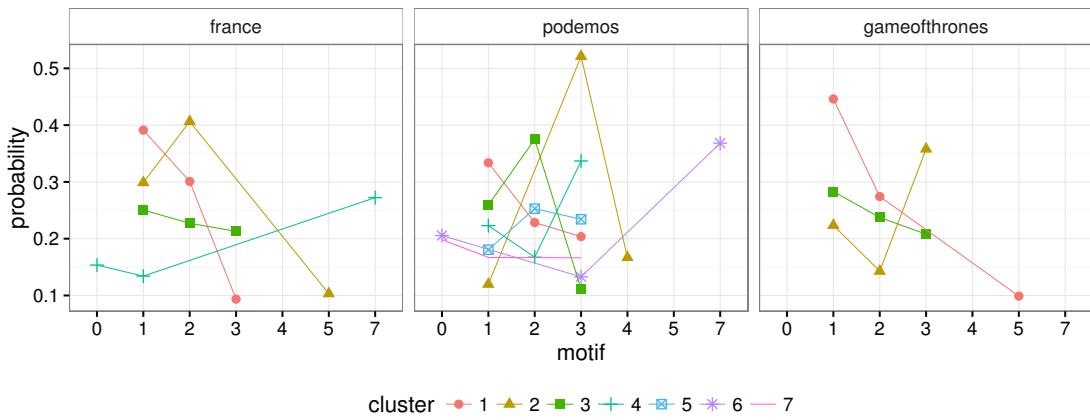


(b) Boxplots of the entries (probabilities) in the feature vectors (each dot is a user). The red solid line indicates the medians.

Figure 3.2 Radius-based neighborhoods ($r=1$)



(a) Heatmap of feature matrices re-arranged after clustering. Clusters are sorted by size (the biggest in the bottom)



(b) Mean feature vectors by cluster (three most dominant features). Clusters are labeled by size (1 for the biggest).

Figure 3.3 Clusters with radius-based neighborhoods ($r=1$)

3. ROLE DETECTION BASED ON CONVERSATION STRUCTURES





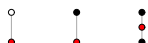


Role	main motifs	ID motifs	forums
<i>Radius-based neighborhoods (r = 1)</i>			
repliers		1,2,*	fr, pod, got
successful repliers		2,1,*	fr, pod
successful repliers		2,3,1	pod
root repliers		3,4,1	pod
root repliers		3,1,2	pod, got
initiators		7,0,*	fr, pod
others (self-repliers)		0,3,1	pod

Table 3.1 Summary of clusters with radius-based neighborhoods ($r=1$). Clusters with similar first and second motif, but different third motif, have been collapsed into the same group (the third motif is marked with an asterisk). The question mark corresponds to the *others* category.

user replies to themselves. This happens, for instance, when users wish to clarify or extend some point made in their previous post. Beyond the seventh motif, medians are all zero. Hence, our set of selected features is made of the first seven motifs plus those with a high enough number of outliers (10%), which are the 9 and the 11.

Clusters. We collapse the non-selected motifs into the *others* category (label 0) and look for clusters in the features matrix. The results are shown in Figure 3.3. We name the cluster by their dominant motifs. We denote as (i, j, k) the three most dominant motifs in a cluster. Since the third motif is usually much less dominant than the two first, we merge those clusters that have the first two dominant motifs and we denote the resulting cluster as $(i, k, *)$. We show a summary in Table 3.1.

- *repliers* (1,2,*): we find this type of user in all three forums. The activity of the *repliers* is focused on replying non-root posts. In **gameofthrones** and **france** they are the majority group.
- *Successful repliers* ((2,1,*), (2,3,1)): these are repliers that tend to get replies to their replies. We find them in **france** and **podemos**.
- *root repliers* (3): these are users that tend to reply to root posts and who mostly get no replies. In **gameofthrones**, this group also reply to non-root posts. In **podemos** the mean probability of motif 3 for this group is specially high (near 0.5).
- *initiators* (7): this is a minority group found in **france** (17 users) and **podemos** (15 users). These are people that open successful threads —the *initiator* term was coined in Chan et al. (2010).








Role	main motifs	ID motifs	forums
<i>Radius-based neighborhoods (r = 2)</i>			
root repliers		1,0,*	fr, pod
terminators		3,7,4	fr
terminators		3,0,7	got
persistent root repliers		21,0,1	got
others-debaters		0,8,1	pod
others-initiators		0,31,37	pod
others		0,*,*	fr, pod, got

Table 3.2 Summary of clusters with radius-based neighborhoods ($r=2$). Clusters with similar first and second motif, but different third motif, have been collapsed into a same group (the third motif is marked with an asterisk). The question mark corresponds to the *others* category.

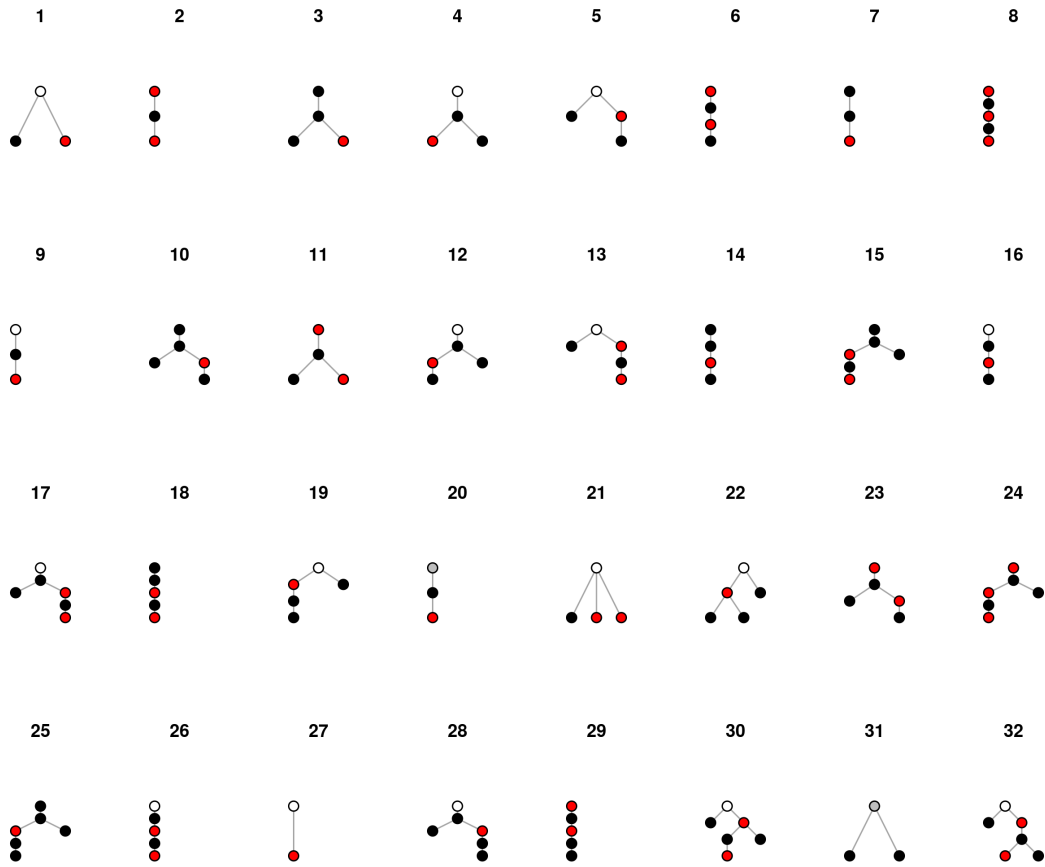
- *self-repliers* (0): this is a micro cluster (15 users) in `podemos` whose preferred motifs are the *other* category. That is, motifs that include a self-reply from the user. The fact that there are a lot of debates in the threads of `podemos`—sometimes very heated debates, either between supporters or between supporters and trolls— may be the cause of this pattern where a user wants to make their point clear.

Results for $r = 2$

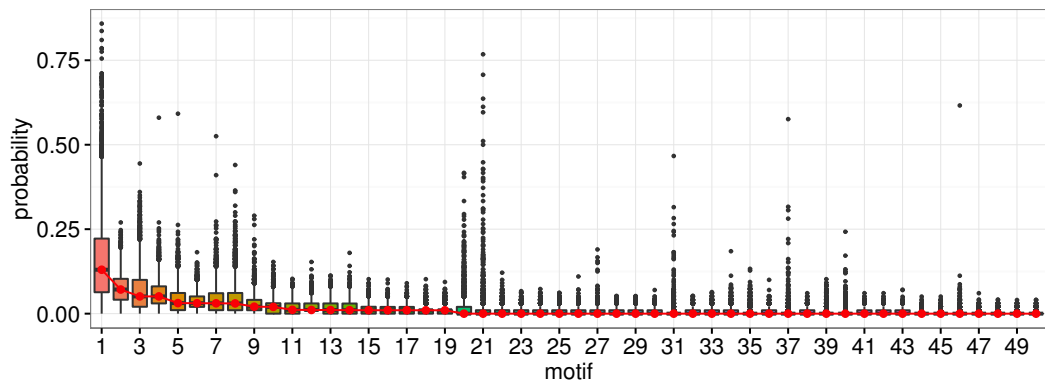
Dictionary. We found 1269 motifs for $r = 2$ (Figure 3.4). A good property of this radius is that we detect some interesting motifs that were undetectable with $r = 1$ —to see how a motif with $r = 2$ would be detected if we applied $r = 1$ we just have to remove those vertices that are further than two steps from any red vertex. For instance, the most popular motif for $r = 2$, a reply to a root that has more replies, is non-detectable by $r = 1$; a neighborhood with $r = 1$ sees no difference between being the only replier to a root (in a thread that grabbed not attention) or being one among many repliers. A cascade that starts at the root and ends in the ego (motif 9), would be seen by $r = 1$ as a dyad between the ego and a non-root post. Since these cases are among the most popular motifs, the number of meaningful structures that were being overlooked with $r = 1$ is not negligible. Nonetheless, the price to pay is a much larger dictionary of motifs.

On the other hand, some motifs that were relevant in $r = 1$ are now split apart in multiple motifs. Recall, for instance, that a successful root post (motif 7 in $r = 1$) was the dominant motif in the *initiators* cluster. Now this motif has been split in multiple

3. ROLE DETECTION BASED ON CONVERSATION STRUCTURES

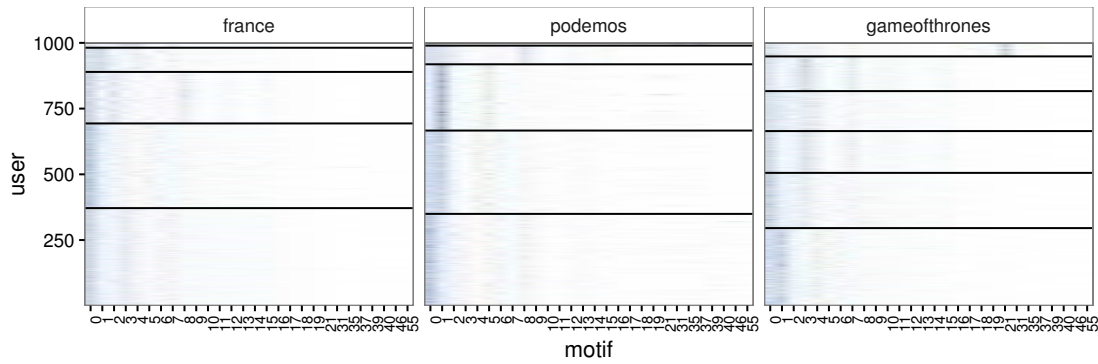


(a) Dictionary of the first motifs sorted by median probability (out of 1269).

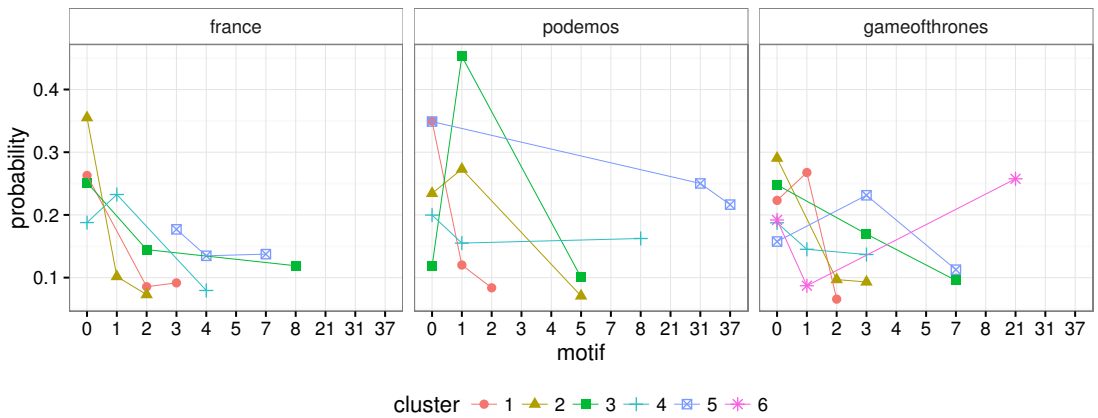


(b) Boxplots of the entries (probabilities) in the feature vectors (each dot is a user). The red solid line indicates the medians.

Figure 3.4 Radius-based neighborhoods ($r=2$)



(a) Heatmap of feature matrices re-arranged after clustering. Clusters are labeled by size (1 for the biggest)



(b) Mean feature vectors by cluster (three most dominant features). Clusters are labeled by size (1 for the biggest).

Figure 3.5 Clusters with radius-based neighborhoods ($r=2$)



Figure 3.6 Four motifs detected using radius-based neighborhood with $r = 2$ that are considered the same with $r = 1$. For $r = 1$, all four are seen as the three-vertices motif on the left.

sub-cases (see Figure 3.6). The risk of such granularity is that each sub-case has lower probability and they may become insignificant.

Beyond the nineteenth motif, medians are all zero. Hence, our set of selected features is made of the first nineteen motifs plus those with a high enough number of outliers (10%), which are the 21, 31, 35, 37, 39, 40, 46, and 55.

Clusters. A consequence of a large dictionary of motifs is that the *others* category accumulates a lot of the probability mass (see Figure 3.5). Indeed, this feature is the strongest in many clusters —and among the strongest in all of them. We show a summary in Table 3.1.

- *root repliers* (1,0,*): they are mostly found in **france** and **podemos**.
- *terminators* ((3,7,4), (3,0,7)): these are users that tend to have the last word in a conversation. Their posts are found at the end of chains (consecutive replies). They are found in **france** and **gameofthrones**.
- *persistent repliers* (21,0,1): a small group of users in **gameofthrones** that tend to give many replies to the root post.
- *debaters* (0,8,1): debaters are found participating multiple times in the same. The other participant is often the same ($A \leftarrow B \leftarrow A \leftarrow B$). These are found in **podemos**. Indeed, reading the content of the posts we have confirmed that they are usually debates. Note that this behavior cannot be detected with radius $r = 1$.
- *initiators* (0, 31,37): *initiators* are still detected in **podemos** but not in **france** due to the specialization effect (Figure 3.6).
- *others* (0,*): users whose dominant motifs are in the *other* category.

Overall, there are some groups such as the *persistent repliers*, the *terminators* that cannot be detected with radius $r = 1$. Yet the *initiators*, which is clearly a meaningful group, is not detected in **france**. Moreover, the *other* category of motifs is dominant in several clusters, meaning the probability is more spread among the motifs (specialization effect), which makes some patterns to go unnoticed.

3.4 Order-based neighborhoods

The previous definition only considers the structural distance to the ego. The closer a post to the ego, the higher its preference to be selected as a neighbor. However, it might be reasonable to also consider time when deciding which posts are neighbors and which posts are not.

There are multiple possibilities to using time to decide what a neighborhood is, although some of them are impractical. We might say, for instance, that the neighborhood is composed of the r posts that are closest in time to the ego. The problem of this is that the closest in time might be far away in the tree —therefore part of another local conversation. We might then say that the neighbors are the r posts that are closest in time and that can reach the ego through a path made by other neighbors. Unfortunately, it happens very often that all the r temporally closest posts are not structurally close as well. In these cases, the ego would be left with no neighbors.

Considering this, we propose a second definition that selects posts first by geodesic distance (to guarantee locality) and then (in the case of a tie) by time distance to the ego. That way we will guarantee the connectivity of the graph. The formal definition is the following:

Definition Given a tree graph G with a timestamp annotated in each vertex, the *order-based neighborhood* of radius r of vertex v , denoted as $\mathcal{N}_v^O(r)$, is the subgraph of the r -th closest neighbors (including the ego) where, if a post with timestamp t is included in the subgraph, then all posts with equal distance and less t_i are also included.

That is, given r we collect the neighbors of v starting from the adjacent vertices and increase the distance until we fill a neighborhood with r vertices. If, in the last distance, we cannot include every vertex because that would overflow the r limit, we select those that are closest in time to the ego.

One of the strengths of this definition is that, unlike the radius-based neighborhood, the neighborhood size is explicitly upper bounded by r . This will make the post-pruning unnecessary.

3.4.1 Pruning and coloring

Order-based neighborhoods $\mathcal{N}_v^O(r)$ contain at most r vertices and thus they need no post-pruning. Yet, it has some effects that we want to avoid. Imagine a post with many replies. If we extract its order-based neighborhood of $r = 3$ we may get two of the children (the closest in time) or one child and the parent (if the parent is closer than most children). But we the grandparent of the post will never have an opportunity to enter the neighborhood because there neighborhood will be filled with the parent and children. In the case of cascades, for instance, if the last post of the cascade has many replies, we would systematically ignore the cascade and detect a *in-star*.

To avoid this we will make our pruning *before* the neighborhood extraction. That this, we color the tree as in Section 3.3, we prune it as in Section 3.3 and *then* we extract the order-based neighborhood around the ego.

3. ROLE DETECTION BASED ON CONVERSATION STRUCTURES

Role	main motifs	ID motifs	forums
<i>Order-based neighborhoods (r = 3)</i>			
Successful repliers		1,3,*	fr, got
Successful repliers		1,4,2	fr, pod
Successful repliers		1,2,*	pod, got
root repliers		2,4,1	fr, pod, got
initiators		9,1,0	fr
initiators		9,13,2	pod
terminators		6,8,3	fr
others		0,1,2	pod

Table 3.3 Summary of clusters with order-based neighborhoods ($r=3$). Clusters with similar first and second motif, but different third motif, have been collapsed into a same group (the third motif is marked with an asterisk). The question mark corresponds to the *others* category.

As for the coloring of vertices, we apply the same colors as in the radius-based neighborhoods.

Choice of radius

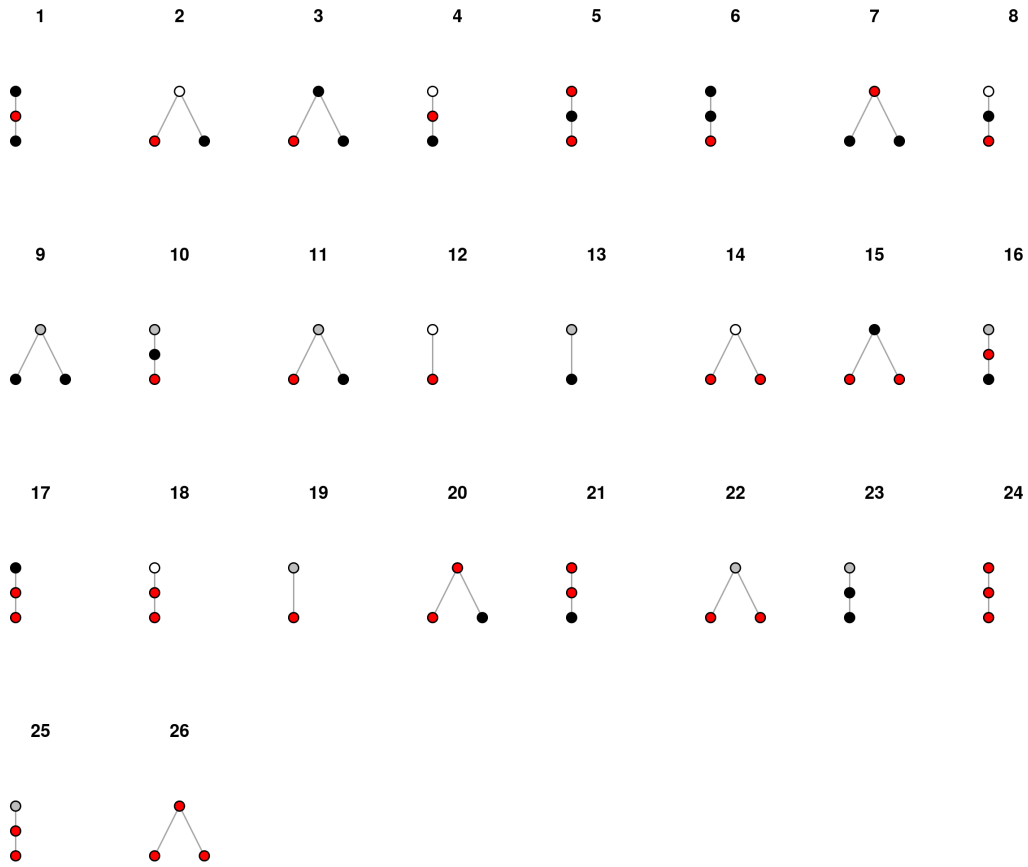
Similarly to the radius-based neighborhood, the choice of the radius is a compromise between the expressiveness of the motifs and the size of the motifs dictionary. Order-based motifs with radius $r = 2$ can only capture dyads (the ego and its parent, or the ego and is son), and thus we rule out this radius. Radius $r = 3$ can capture the siblings, the grandparents and the grandchildren of the ego –two of them– which might be enough expressiveness whereas keeping the dictionary smaller than radius-based neighborhoods with $r = 2$.

3.4.2 Results for order-based neighborhood

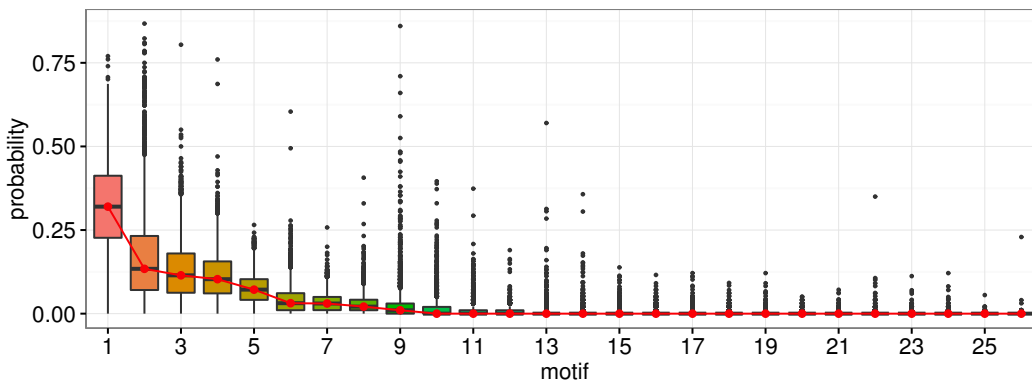
We repeat the workflow of the experiments in the former section for order-based neighborhoods of radius $r = 3$.

Dictionary. The dictionary of motifs has a similar size to that of radius-based neighborhoods with of $r = 1$ (26 motifs) but different content (Figure 3.7). The most popular triads are chains, replies to root posts, posts with replies and root posts with replies. Compared to the radius-based of $r = 1$, the order-based has no the group of 4-vertices motifs that were detected by the radius-based where users reply to themselves.

3.4. Order-based neighborhoods



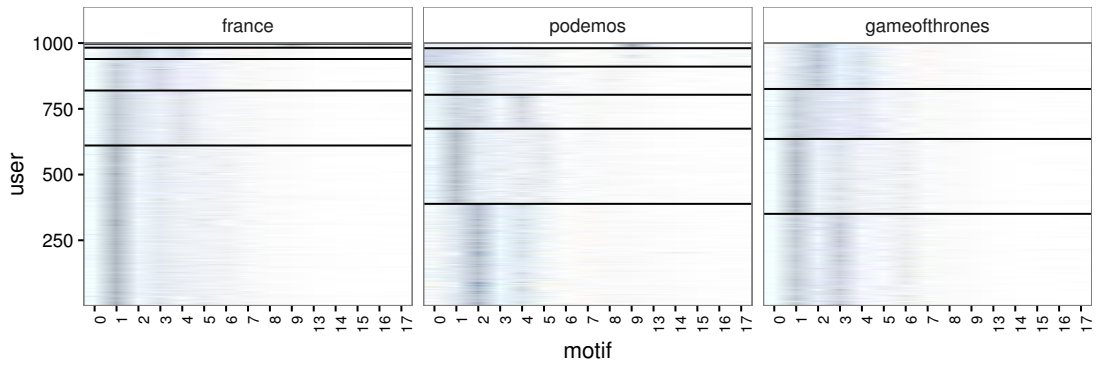
(a) Dictionary of the first motifs sorted by median probability



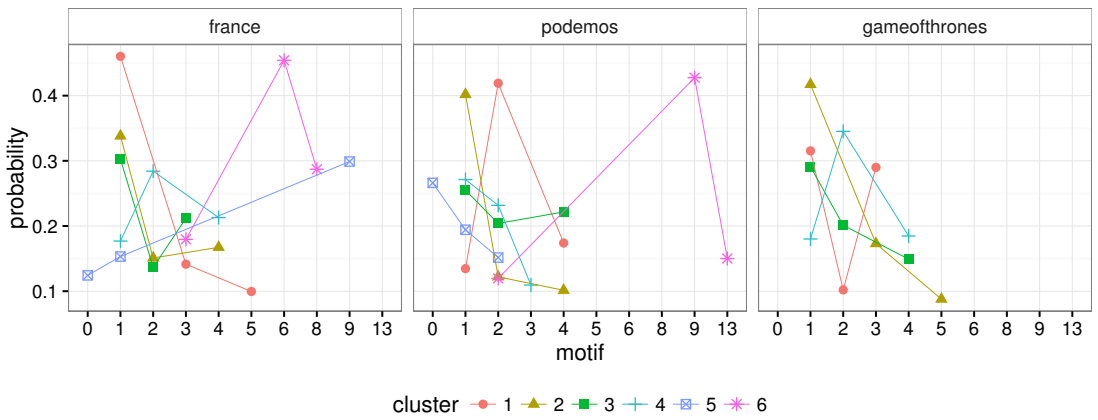
(b) Boxplots of the entries (probabilities) in the feature vectors (each dot is a user). The red solid line indicates the medians.

Figure 3.7 Order-based neighborhoods ($r=3$)

3. ROLE DETECTION BASED ON CONVERSATION STRUCTURES



(a) Heatmap of feature matrices re-arranged after clustering



(b) Mean feature vectors by cluster (three most dominant features).

Figure 3.8 Clusters with order-based neighborhoods ($r=3$)

In exchange, it contains popular 3-vertices motifs that were only detected with the neighborhood-based with $r = 2$. Compared to the radius-based of $r = 2$, the order-based does not suffer from the specialization effect (Figure 3.6). Another compelling characteristic of this dictionary is its similarity to the 16-triad list. Indeed, Note this dictionary correspond to the dyad and the two only triads that are possible in trees: the *in-start* and the *chain* (see Chapter 1). The novelty here is that they have been colored, giving them the ability to capture more conversational structures.

Beyond the ninth motif, medians are all zero. Hence, our set of selected features is made of the first seven motifs plus those with a high enough number of outliers (10%), which are the 10, 11, 12, 13, 14, 15, 16, 17, 18 and 19.

Clusters. Results of the detected clusters are shown in Figure 3.8, and a summary in Table 3.3. We find *successful repliers*, *root repliers*, *initiators*, and *terminators*. Besides the ability to detect behaviors such as the *terminators* (undetected in radius-based neighborhood with $r = 1$), another good property is that the *other* category is only dominant in one cluster (as happened in radius-based $r = 1$).

Order-based neighborhoods of $r = 2$ are able to capture the deep repliers (captured by radius-based of $r = 2$ but not $r = 1$) and the initiators in **france** and **podemos** (the *initiators* in **france** went unnoticed for radius-based of $r = 2$ due to the specialization effect.)

3.5 Time-based neighborhoods

Lastly, we propose a third definition based on time –not just order. To take time explicitly into account, one might set fixed time-based boundaries for the neighborhood and include only those posts whose timestamp t_j is at a distance less than τ from the ego post i , i.e.: $|t_j - t_i| < \tau$. 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 instead decide it by looking at changes in the pace how new posts are added to the thread. In statistical analysis, a *change point* in a sequence x_1, \dots, 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 change points will correspond to sudden pauses in the conversation. In the following, we will differentiate *horizontal change points*, that arise between siblings, from *vertical change points*, that arise within a branch. We say that a sequence posts with timestamps t_1, \dots, t_n belong to the same (vertical or horizontal) *local dynamic* if there is no change point t_i in the sequence such that $1 < i \leq n$. For the detection of change points, we use the PELT algorithm Killick et al. (2012)². Now we can introduce our new definition:

²An implementation of this algorithm written by the authors themselves is available in the R library `change point`. We use their `cpt.meanvar` function.

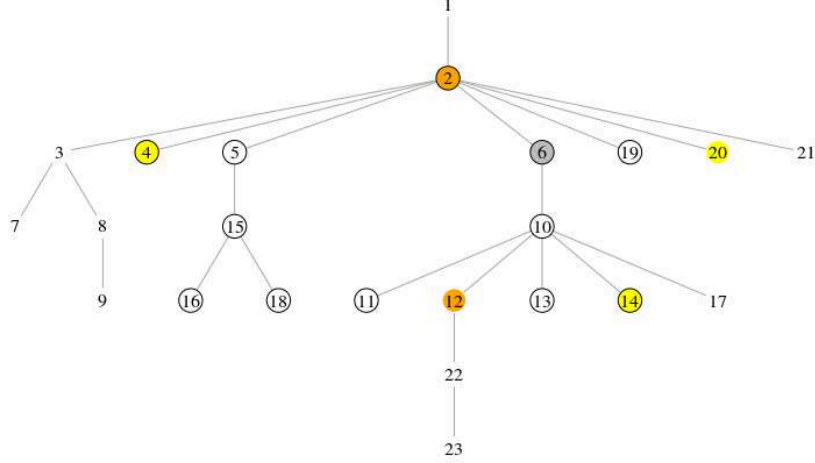


Figure 3.9 Illustration of time-based neighborhoods. The time-based neighborhood has parameter $r = 3$. In the time-based neighborhood, horizontal changepoints (yellow) and vertical changepoints (orange) represent posts that are temporally far from their predecessors (siblings or parents) and therefore set the limits of the neighborhood. In this example $\mathcal{N}_6^T(3)$ is the subtree induced by $\{2,4,5,6,10,11,13,14,15,16,18,19\}$.

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

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 v . This definition can be seen as a *radius-based neighborhood with a time-sensitive pruning*.

Algorithm 2 extracts time-based neighborhoods of a given post according to the above definition. Since the changepoints only depend on the tree and not on the particular post we analyze, we previously detect the horizontal and vertical changepoints in the tree. In the case of multiple branches with some common posts, we consider that a common post is a vertical changepoint if it is a vertical changepoint in any of the branches. Once we have the changepoints, we can proceed with the algorithm. First, we extract the radius-based neighborhood. The time-based neighborhood will be a subset of the later. Then, we look for horizontal and vertical changepoints, which mark the frontiers of the time-based neighborhood. There are four possible cases:

- *A vertical changepoint in the ancestors:* If the changepoint is in the path between the post and the root, then the changepoint started the new local dynamic to which the ego post belongs. Thus, we remove the ancestors of the changepoint, but not the changepoint itself.

- *A vertical changepoint in the descendants:* If the changepoint is a descendant then it started a new different dynamic and therefore we must remove the descendants of the changepoint and the changepoint itself.

- *A horizontal changepoint either in the older siblings or the ancestors:* If the changepoint is among the older siblings, then it started the horizontal dynamic to which the ego post belongs. Similarly, if the changepoint is among the ancestors, then every older sibling of the changepoint belongs to a previous local dynamic. In both cases, we remove the older siblings of the changepoint but not the changepoint itself.

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

Any time a vertex is removed we also remove its descendants. $\mathcal{N}_v^T(r)$ is a connected induced subgraph of $\mathcal{N}_v(r)$. If there are no changepoints in the structural neighborhood,

then $\mathcal{N}_v^T(r) = \mathcal{N}_v(r)$. An example of a time-based neighborhood is given in Figure 3.9.

Algorithm 2: Extraction of time-based neighborhood

Data: Posts tree G , vertical changepts, horizontal changepts, ego post i ,
radius r

Result: $\mathcal{N}_i^T(r)$: Time-based neighborhood of i at radius r

- 1 Compute structural neighborhood $\mathcal{N}_i(r)$;
- 2 $\text{ancestors} \leftarrow \text{ancestors}(i)$ in $\mathcal{N}_i(r)$;
- 3 $\text{older_siblings} \leftarrow \text{older_siblings}(i)$ in $\mathcal{N}_i(r)$;
- 4 $\text{dump} \leftarrow \emptyset$;
- 5 **for** $bp \in \text{horizontal changepts}$ **do**
- 6 **if** $bp \in (\text{older_siblings} \cup \text{ancestors})$ **then**
- 7 $\text{dump} \leftarrow \text{dump} \cup \text{ancestors}(bp)$;
- 8 **else**
- 9 $\text{dump} \leftarrow \text{dump} \cup \text{descendants}(bp) \cup bp$;
- 10 **end**
- 11 **end**
- 12 **for** $bp \in \text{horizontal changepts}$ **do**
- 13 **if** $bp \in (\text{older_siblings} \cup \text{ancestors})$ **then**
- 14 $\text{dump} \leftarrow \text{dump} \cup \text{older_siblings}(bp)$;
- 15 **else**
- 16 $\text{dump} \leftarrow \text{dump} \cup \text{younger_siblings}(bp) \cup bp$;
- 17 **end**
- 18 **end**

Pruning and coloring are applied as in the radius-based definition.

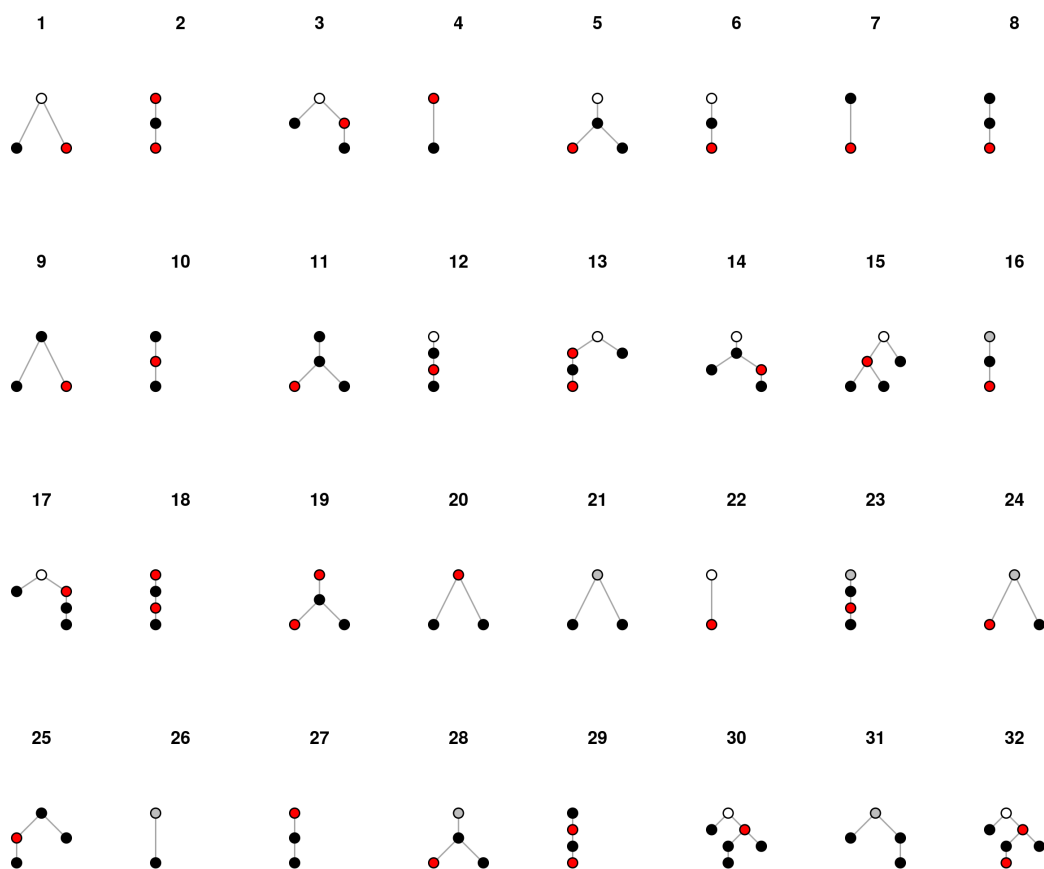
3.5.1 Choice of radius

Because changepts are not necessarily close from each other, laying only on the changepts, or choosing large radiuses would give too large motifs. Thus, we choose a radius $r = 2$ so that the dictionary is, at most, the size of the radius-based dictionary of $r = 2$. In this sense, a time-based neighborhood may be seen as a radius-based neighborhood with an extra pruning based on speed changes.

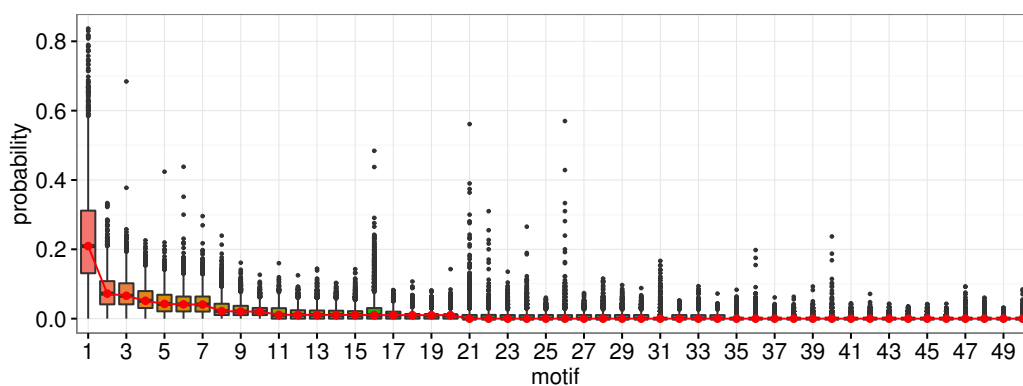
3.5.2 Results for time-based neighborhood

We repeated the workflow of the experiments in the former section. We chose a radius $r = 2$ so that the dictionary has, at most, the same motifs than a radius-based dictionary with the same radius.

Dictionary. The final dictionary contains 168 motifs, much less than the 1269 for radius-based with $r = 2$, meaning that changepts help to considerably reduce the size of neighborhoods (Figure 3.10). Beyond the nineteenth motif, medians are all zero. Hence, our set of selected features is made of the first nineteen motifs plus those with a high enough number of outliers (10%), which are the 21, 31, 36 and 40.



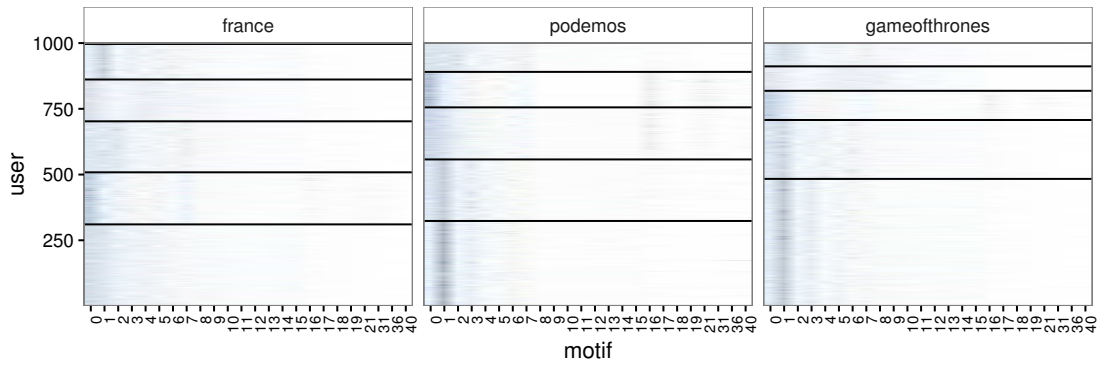
(a) Dictionary of the first motifs sorted by median probability (out of 746).



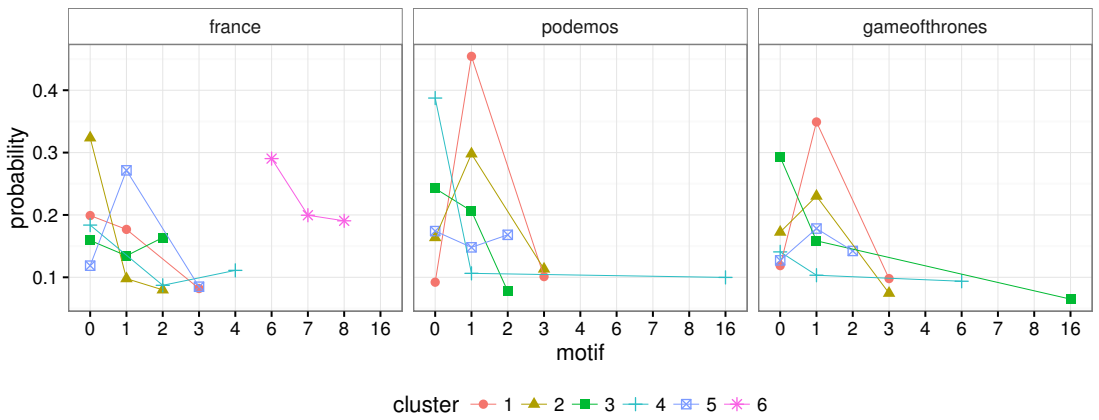
(b) Boxplots of the entries (probabilities) in the feature vectors (each dot is a user). The red solid line indicates the medians.

Figure 3.10 Time-based neighborhoods ($r=2$)

3. ROLE DETECTION BASED ON CONVERSATION STRUCTURES



(a) Heatmap of feature matrices re-arranged after clustering



(b) Mean feature vectors by cluster (three most dominant features)

Figure 3.11 Clusters with time-based neighborhoods ($r=2$)


















Role	main motifs	ID motifs	forums
<i>Time-based neighborhoods (r = 2)</i>			
other - repliers	?  	0,4,2	fr
other - repliers	?  	0,2,1	pod
other - root repliers	?  	0,1,*	fr, pod, got
root repliers	 ? 	1,0,3	fr, pod, got
root repliers	  ?	1,3,0	pod
root repliers	  ?	1,2,0	got
debaters	 ? 	2,0,1	fr, pod
terminators	  	6,7,8	fr

Table 3.4 Summary of clusters with time-based neighborhoods ($r=2$). Clusters with similar first and second motif, but different third motif, have been collapsed into a same group (the third motif is marked with an asterisk). The question mark corresponds to the *others* category.

Clusters. Results are shown in Figure 3.11, and a summary in Table 3.4. We detect *repliers*, *root repliers*, and *terminators* and *debaters*. The debaters, however, are in shorter chains than those found in radius-based with $r = 2$. The reason is that there is usually a changepoint in the chain. Besides, debaters are not only found in *podemos* but also in *france*.

3.6 Comparative analysis

In this section, we give an overview of the results for each the neighborhoods.

Radius-based with $r = 1$ detects *repliers*, *successful repliers*, *root repliers* and *initiators*. It gives a dictionary of 28 motifs, 7 of which have a median higher than 0. With our feature selection criteria we selected 9 features.

Radius-based with $r = 2$ detects *root repliers*, *persistent repliers*, *debaters*, *initiators* and *terminators*. It gives a dictionary of 1269 motifs, 19 of which have a median higher than 0. With our feature selection criteria we selected 27 features.

Order-based with $r = 3$ (three vertices) detects *successful repliers*, *initiators* and *terminators*. It gives a dictionary of 26 motifs, 9 of which have a median higher than 0. With our feature selection criteria we selected 14 features.

Time-based with $r = 2$ detects *repliers*, *root repliers*, *debaters*, and *terminators*. It gives a dictionary of 746 motifs, 19 of which have a median higher than 0. With our feature selection criteria we selected 23 features.

It is clear that the *radius-based* neighborhood with a radius $r = 1$ misses some interesting structures. It is unable, for instance, to detect users that tend to end conversations. Nonetheless, since it does not suffer from the specialization problem observed in other neighborhoods, the *initiators* are properly detected.

The main disadvantage of the *radius-based* neighborhood with a radius $r = 2$ is its huge dictionary. The consequence of a big dictionary is that the counts, and therefore the probability mass, is too spread and some relevant motifs are actually seen as two different ones (the specialization problem). Since we collapse the motifs with low probability into an *other* category, we can see that this category is actually dominant in many clusters, which means that the total probability mass in these less frequent motifs is not negligible. In practice, the method does not see the *initiators* in *france*. Nevertheless, it is able to detect new types of clusters such as the *debaters* and the *terminators*.

The main advantage of the *order-based* neighborhood with $r = 3$ is that it keeps a small dictionary. Besides, it sees some clusters that radius-based with $r = 1$ does not see (*terminators*). Among the repliers, only *successful repliers* are detected.

Regarding the *time-based* neighborhood, proposed as an improvement of the radius-based with $r = 2$, it achieved a considerable reduction of the dictionary. However, *initiators* are even more ignored, even if some motifs that represent initiators have higher popularity in the time-based dictionary.

Given the cost of computing isomorphisms in large dictionaries, the order-based neighborhood seems to us the most practical method to detect conversational structure.

Overall, there are some reasonable arguments to prefer the order-based of $r = 2$ versus any of the others, namely: (a) a small dictionary (b) with more expressiveness than a radius-based with $r = 1$. Moreover, we stay in the realm of triads, which are already a common unit of analysis in social networks. Yet some motifs might be merged in order generate a base of relevant motifs where the each motif represents a unique type of conversation.

3.7 Summary

The goal of this chapter was to shed some light on what would be a proper method to characterize local conversations in order to detect different types of *conversationalists*. We have introduced a new approach to describing users behaviors based on the structure of their conversations in the tree representation of discussion threads. Particularly, we have compared three different definitions of *neighborhood* to capture the structure where a post is embedded: *radius-based*, *order-based* and *time-based*. The choice of the proper neighborhood is not trivial because of a trade-off between expressiveness (i.e.: the variety of structures that a neighborhood definition captures) and sparsity of the dictionary (i.e.: the number of neighborhood classes, or motifs, that have very low frequencies).

To take into account the posts authorship and to differentiate, for instance, between a common reply and a reply to a root post, we have used colors to label posts written by the user under study and the root post. We also apply a pruning strategy to remove redundant parts of the conversations that add no extra information.

Radius-based neighborhoods are extensions of classic neighborhoods in graphs. While neighborhoods of radius $r = 1$ cannot detect some interesting patterns, the dictionary of motifs for $r = 2$ grows to more than 1000 motifs, which makes the dictionary too sparse to be useful for clustering and further interpretations. Time-based neighborhoods are built upon radius-based. Once we have the radius-based neighborhood, we apply a pruning in those points of the neighborhood where a sudden decrease in the speed is detected. We have shown that, although this considerably decreases the size of the dictionary, there is no clear improvement in the type of clusters detected.

Order-based neighborhoods are similar to radius based except that it accepts a maximum of r vertices in the neighborhood. First, we add neighbors at distance one, then those at distance two, and so on until we have r neighbors. At each distance, we first add those vertices that are closer to the ego post. Order-based neighborhood with $r = 3$ give triad motifs, which makes them similar to the classic triads used in social network analysis. Because the reduced size of its dictionary and the type of conversations that it is able to detect, we consider this neighborhood the most promising among all the proposed neighborhoods. Besides, a possible enhancement of the order-based neighborhood is to merge manually—probably under some sociological criteria—those whose difference is not meaningful.

We have shown that the structure of conversation provides novel information that is unveiled by other feature-based methods. We think that the clear cluster structure found in some cases (i.e.: users with a very high preference towards some type of structures) supports this claim.

Also, other variations of these neighborhoods may also be studied to analyze different phenomena. For instance, we might take only those neighbors who are descendants of the ego post; such kind of neighborhood may be used to classify users with respect to the reaction they trigger in the discussion. The inverse technique, that is, take all the neighbors that are not descendants of the ego, may help to understand the kind of discussion that a user tends to be attracted to. This might be useful for posts recommendations or even to improve current generative models that try to reproduce the way how a thread grows [Gómez et al. \(2012\)](#); [Kumar et al. \(2010\)](#).

Language analysis might also be included in the definition of neighborhood. For instance, we might detect the sentiment of each post and add this information to the motif using a larger color code (e.g.: positive/negative sentiment root, positive/negative sentiment ego, and so on). Nonetheless, this will likely increase the dictionary of motifs, so it probably should be combined with a more aggressive pruning or merging strategies.

4 *Role detection based on thread growth models*

This appraisal of the hypothesis relies solely upon deductive consequences (predictions) which may be drawn from the hypothesis: There is no need even to mention “induction”.

Karl Popper

In this chapter, we propose a method to cluster users based on behavioral functions. A behavioral function is a probability distribution that models the replying behavior of users. We infer latent groups of users where each group has different parameters of the behavioral function.

Contents

4.1	Network Growth models	63
4.1.1	Barabasi-Albert (1999)	64
4.1.2	Kumar et al. (2010)	65
4.1.3	Gómez et al. (2010)	65
4.1.4	Gómez et al. (2012)	66
4.2	A new role-based network growth model	67
4.2.1	Formalization	68
4.2.2	Expectation-Maximization for the role-based growth model	68
4.3	Experiments	71
4.3.1	Dataset	71
4.3.2	Inference	71
4.3.3	Structural properties	74
4.3.4	Link prediction	74
4.4	Summary	79

IMAGINE that we observe the behaviors of individuals in a given population. 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. Their observed behaviors are eating, sleeping, exercising, firefighting and rescuing cats from trees. 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.

In the previous chapter, we proposed detecting roles based on the structure of the conversations where users participate. This method may be a good complement to other feature-based method since it considers a new dimension of user behavior: conversations. Nonetheless, this approach is purely descriptive. There is still a missing link that connects a user’s observed behavior to some *behavioral function* that models why the user has chosen that action. 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 users who share the same behavioral function hold the same role. This, our definition of role in this chapter is:

Definition Two users have the same role if they tend share the same (parameters of a) behavioral function.

In this chapter we set three main 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 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 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 :

$$p(x_t \sim i | G_{t-1}; \theta) \tag{4.1}$$

where G_{t-1} is the state of the graph before x_t is attached, and θ 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 set of parameters $\theta_1, \dots, \theta_K$, and the above probability will therefore depend on the θ associated to the author of the post x_i —the author’s role.

The structure of this chapter is as follows. In Section 4.1, we present the different growth models that can be applied to tree graphs. In Section 4.2, 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 θ . The idea is very simple and consists in estimating, by Expectation-Maximization, clusters of users with their own parameters. In Section 4.3, 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.

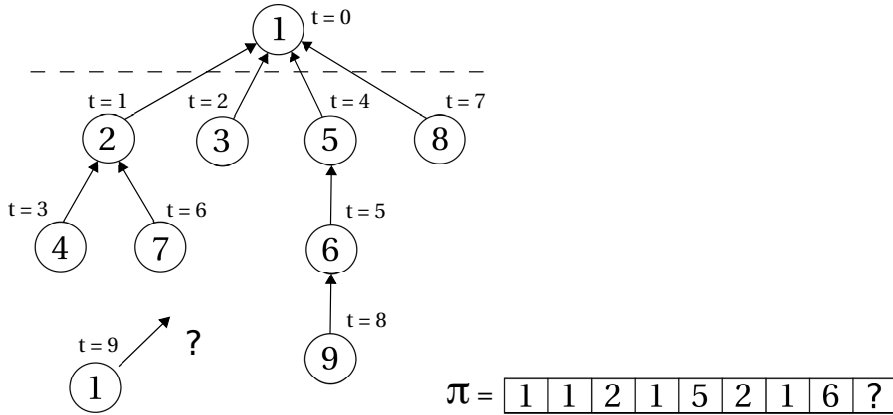


Figure 4.1 An example of a thread and its parents vector π . π starts at $t=1$ (π_1) because the root (π_0) has not parent.

4.1 Network Growth models

Random graph models are stochastic generators of graphs that 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. If the generated graphs and the real-world graphs share some relevant properties, then the proposed growth mechanism 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 $\pi_{(1:(t-1))} = (\pi_1, \dots, \pi_{t-1})$ where π_n is the parent of the post written at the time-step n . Note that this representation does not lose any structural information (see Figure 4.1). With this notation, Equation 4.1 can be re-expressed as:

$$p(\pi_t = i | \pi_{1:(t-1)}, \theta) \quad (4.2)$$

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 ($\pi_1 = 1$). Then, at each time-step t a post is added to the tree creating a new vertex (a reply) to an older post i ($\pi_t = i$). 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 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

Authors	$p(\pi_t = k \boldsymbol{\pi}_{1:t-1}) \propto$	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 4.1 Growth models for online discussions

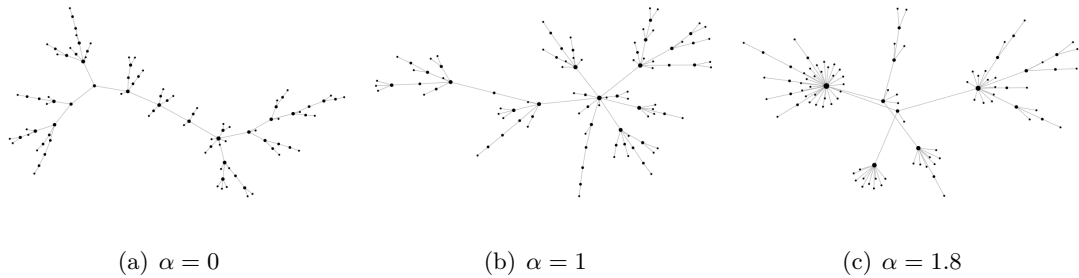


Figure 4.2 Barabasi-Albert graphs with one edge created at every step.

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 5.1.

4.1.1 Barabasi-Albert (1999)

The *preferential attachment* model proposed by Barabási and Albert (1999) is the best known growth model. 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

$$p(\pi_t = i | \boldsymbol{\pi}_{1:(t-1)}) = \frac{d_{i,t}^\alpha}{Z_t}; \quad Z_t = \sum_{j=1}^t d_{j,t}^\alpha \quad (4.3)$$

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. This model reproduces a rich-get-richer phenomena controlled by the parameter α . Figure 4.2 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.

4.1.2 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:

$$p(\pi_t = i | \boldsymbol{\pi}_{1:t-1}) = \frac{\alpha d_{i,t} + \tau^{t-i}}{Z_t} \quad \text{for } \alpha \geq 0 \quad \tau \in (0, 1) \quad (4.4)$$

and the probability of stopping the thread is:

$$p(\pi_t = \emptyset | \boldsymbol{\pi}_{1:t-1}) = \frac{\delta}{Z_t} \quad (4.5)$$

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:

$$Z_t = \delta + \sum_{j=1}^t \alpha d_{j,t} + \tau^{t-j} = \delta + 2(t-1) + \frac{\tau(\tau^t + 1)}{\tau - 1} \quad (4.6)$$

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:

$$a(v) = \begin{cases} a(w) & \text{with probability } \gamma \\ a(u) & \text{with probability } \epsilon \\ a \in A & \text{with probability } 1 - \gamma - \epsilon \end{cases} \quad (4.7)$$

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.

4.1.3 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

$$p(\pi_t = i | \boldsymbol{\pi}_{1:t-1}) = \frac{(\beta_i d_{i,t})^{\alpha_i}}{Z_t} \quad (4.8)$$

where

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

$$\beta_i = \begin{cases} \beta & \text{for } i = 1 \\ 1 & \text{for } i \in \{2, \dots, t\} \end{cases} \quad (4.9)$$

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:

$$Z_t = \sum_{l=1}^t (\beta_l d_{l,t})^{\alpha_l} \quad (4.10)$$

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 4.3 shows some trees generated with their estimated parameters for four different datasets.

4.1.4 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:

$$p(\pi_t = i | \boldsymbol{\pi}_{1:t-1}) = \frac{\beta_i + \alpha d_{i,t} + \tau^{t-i}}{Z_t} \quad \text{for } \alpha, \beta \geq 0, \quad \tau \in (0, 1) \quad (4.11)$$

where

$$\beta_i = \begin{cases} \beta & \text{for } i = 1 \\ 0 & \text{for } i \in \{2, \dots, t\} \end{cases} \quad (4.12)$$

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:

$$Z_t = \beta + 2(t-1) + \frac{\tau(\tau^t - 1)}{\tau - 1} \quad (4.13)$$

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.

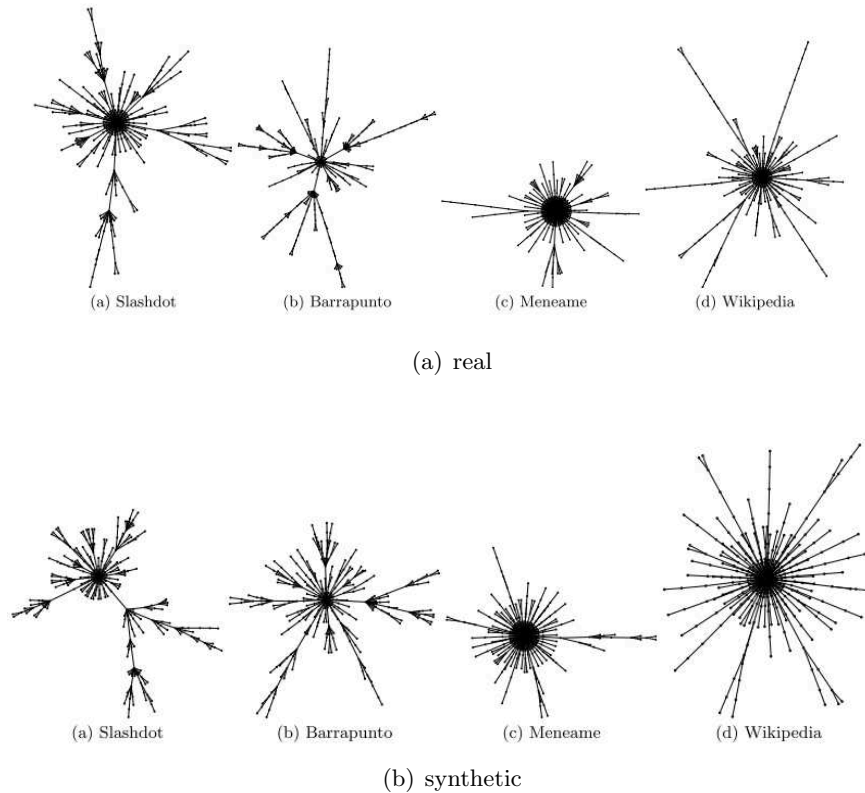


Figure 4.3 Random graphs for discussion threads (Gómez et al., 2010).

4.2 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 θ_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.

4.2.1 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_n denote the degree of its parent; let r_n be 1 if the parent of n is the root, and 0 otherwise. Let l_n be the number of time-steps elapsed between the parent of n and n ($l_n \geq 1$). Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ be the set of posts 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 $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K$ where $\boldsymbol{\theta}_k = \{\alpha_k, \beta_k, \tau_k\}$. Let z_u be the cluster—or role—of user u . Let N_u be the set of posts written by u . The log-likelihood of the whole dataset can be expressed as:

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{u=1}^U \sum_{n \in N_u} \ln \left(\alpha_{z_u} d_n + \beta_{z_u} r_n + \tau_{z_u}^{l_n} \right) - \ln Z_n \quad \text{for } \alpha, \beta \geq 0, \quad \tau \in (0, 1) \quad (4.14)$$

where Z_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:

$$\ln Z_n = \ln \left\{ \sum_{m \in M} \alpha_{z_n} d_m + \beta_{z_n} r_m + \tau_{z_n}^{l_m} \right\} \quad (4.15)$$

$$= \ln \left\{ \alpha_{z_n} \sum_{m \in M} d_m + \beta_{z_n} \sum_{m \in M} r_m + \tau_{z_n} \sum_{m \in M} \tau_{z_n}^{l_m} \right\} \quad (4.16)$$

$$= \ln \left\{ \alpha_{z_n} (2t - 1) + \beta_{z_n} + \frac{\tau_{z_n} (\tau_{z_n}^t - 1)}{\tau_{z_n} - 1} \right\} \quad (4.17)$$

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 thread¹. 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.

4.2.2 Expectation-Maximization for the role-based growth model

We want to estimate the parameters of each role $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K$ and the latent role of every user z_1, \dots, z_U . Let $\mathbf{z}_u = (z_{u1}, \dots, z_{uK})$ be the membership vector of user u where $z_{uk} = 1$

¹We denote as z_n the cluster of the author of post n in order to be consistent with the literature as well as with our following section and our following chapter. This should not be confused with Z_t , which refers to the normalization factor. Nevertheless, this normalization factor will not appear any longer.

if u belongs to cluster k and 0 otherwise. Let \mathbf{Z} be the matrix of membership vectors. If there was one group of θ 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 a considering 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} :

$$\ln p(\mathbf{X}|\theta) = \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta) \quad (4.18)$$

Unfortunately we cannot analytically maximize the parameters θ 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:

$$\ln p(\mathbf{X}|\theta) = \ln \sum_{\mathbf{Z}} \overbrace{q(\mathbf{Z}) \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}}^{\mathbb{E}_{\mathbf{Z}}[\frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}]}} \quad (4.19)$$

Thanks to this trick, we can use Jensen's inequality to get the sum of outside the logarithm. We know, by Jensen's inequality, that the logarithm of a expected value is always greater or equal than the expected value of the logarithm². Therefore:

$$\ln p(\mathbf{X}|\theta) = \ln \sum_{\mathbf{Z}} \overbrace{q(\mathbf{Z}) \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}}^{\mathbb{E}_{\mathbf{Z}}[\frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}]}} \geq \sum_{\mathbf{Z}} \overbrace{q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}}^{\mathbb{E}_{\mathbf{Z}}[\ln \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}]}} \quad (4.20)$$

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

$$q(\mathbf{Z}) = \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{\int_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta)} = p(\mathbf{Z}|\mathbf{X}, \theta) \quad (4.21)$$

which is the posterior distribution of \mathbf{Z} given the observed data and the parameters. Replacing $q(\mathbf{Z})$ by the posterior in Equation 4.20 we obtain:

$$\underbrace{\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta) (\ln p(\mathbf{Z}|\pi) + \ln p(\mathbf{X}|\mathbf{Z}, \theta))}_{\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z}|\theta)]} - \sum_{\mathbf{Z}} \overbrace{p(\mathbf{Z}|\mathbf{X}, \theta) \ln p(\mathbf{Z}|\mathbf{X}, \theta)}^{\mathcal{H}(\mathbf{Z})} \quad (4.22)$$

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

where $\boldsymbol{\pi}$ 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 $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K$ and the class assignments $\mathbf{z}_1, \dots, \mathbf{z}_U$ until a lower bound of the likelihood converges. That is, we maximize this term:

$$\underbrace{\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) \overbrace{(\ln p(\mathbf{Z}|\boldsymbol{\pi}) + \ln p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\theta}))}^{\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}}_{\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})]} \quad (4.23)$$

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

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

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

$$\mathbb{E}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})] = \mathbb{E} \left[\sum_{u=1}^U \sum_{k=1}^K z_{uk} \{ \ln \pi_k + \ln p(\mathbf{X}_u|\boldsymbol{\theta}_k) \} \right] \quad (4.24)$$

$$= \sum_{k=1}^K \sum_{u=1}^U \mathbb{E}[z_{uk}] \{ \ln \pi_k + \ln p(\mathbf{X}_u|\boldsymbol{\theta}_k) \} \quad (4.25)$$

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

$$\arg \max_{\boldsymbol{\theta}_k} \sum_{u=1}^U \mathbb{E}[z_{uk}] \{ \ln \pi_k + \ln p(\mathbf{X}_u|\boldsymbol{\theta}_k) \} \quad (4.26)$$

and for the π parameter:

$$\boldsymbol{\pi}_k = \frac{1}{U} \sum_{u=1}^U \mathbb{E}[z_{uk}] \quad (4.27)$$

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

$$p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) = \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{\sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})} = \frac{\prod_{u=1}^U \prod_{k=1}^K p(\mathbf{X}_u|\boldsymbol{\theta}_k)^{z_{uk}}}{\sum_{\mathbf{Z}} \prod_{u=1}^U \prod_{k=1}^K p(\mathbf{X}_u|\boldsymbol{\theta}_k)^{z_{uk}}} \quad (4.28)$$

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

$$\mathbb{E}[z_{uk}] = \sum_{z_{uk}} z_{uk} \frac{\pi_k p(\mathbf{X}_u | \boldsymbol{\theta}_k)}{\sum_{k=1}^K \pi_k p(\mathbf{X}_u | \boldsymbol{\theta}_k)} = \frac{\pi_k p(\mathbf{X}_u | \boldsymbol{\theta}_k)}{\sum_{k=1}^K \pi_k p(\mathbf{X}_u | \boldsymbol{\theta}_k)} \quad (4.29)$$

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

$$p(\mathbf{X}_u | \boldsymbol{\theta}_k) = \prod_{n \in N_u} p(\mathbf{x}_n | \boldsymbol{\theta}_k) \quad (4.30)$$

The E-step is done with Equation 4.29 while the M-step is done with Equation 4.25. And because Equation 4.25 cannot be analytically maximized due to the form of our likelihood, we use Nelder-Mead optimization as in Gómez et al. (2012).

4.3 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 of each cluster, and to use these parameters to predict a user behavior in new threads. If the model can predict, 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 4.3.2). 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 4.3.3). Lastly, we test whether our model can make better predictions of post replies in a test set (Section 4.3.4).

4.3.1 Dataset

In order to test the predictive power of our model, 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* ($\boldsymbol{\pi}, \alpha_k, \beta_k, \tau_k$ 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.

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.

4.3.2 Inference

To infer the cluster of each user $\mathbf{z}_1, \dots, \mathbf{z}_U$ and the cluster parameters $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K$ we initialize the parameters α, β, τ for each cluster and run the Expectation and Maximization steps

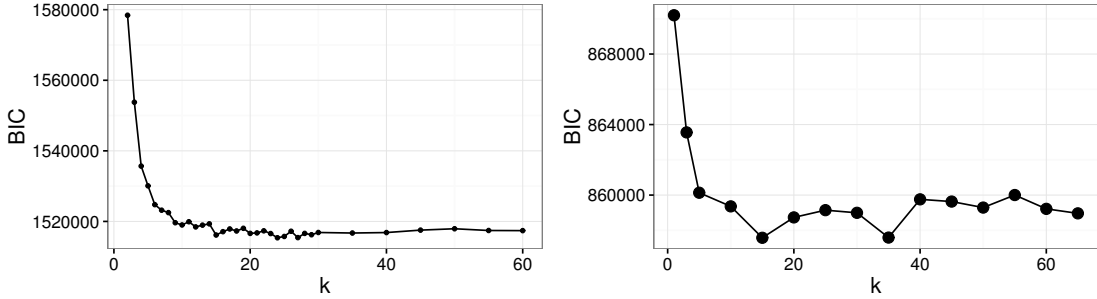


Figure 4.4 BIC values in `podemos` and `gameofthrones`

defined in Section 4.2.2. 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.

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

$$BIC = -2\mathcal{L} + \log(n)n_p \quad (4.31)$$

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 the BIC. We computed the BIC in the validation set and we found a minimum BIC at $k = 24$ in `podemos`. In `gameofthrones`, we found a first minimum at $K = 15$. In Figure 4.4 we show the BIC curves for `podemos` and `gameofthrones`.³ We also computed, for each user, the uncertainty of its classification as $(1 - \max(z_{i1}, \dots, z_{iK}))$ (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. We finally chose $K = 15$ for `gameofthrones`, and $K = 15$ for `podemos` to keep a low complexity of the model—as a sanity check, we repeated the experiments below for higher K and obtained very similar results.

The estimated parameters are shown in Table 4.2 and Table 4.3.

All clusters have different parameters than `gomez`, meaning that not all users have behaved similarly in our training set. For instance, members of cluster 6 in `podemos` and members of cluster 8 in `gameofthrones` show a extremely high tendency to reply to root posts (high β). Another detected group with an extreme behavior is the cluster

³We also computed the Akaike Information Criteria (AIC), defined as $AIC = -2\mathcal{L} + 2n_p$ and obtained similar curves.

cluster	α	β	τ	π	users
1	0.03	1.48	0.69	0.12	121
2	0.02	3.63	0.73	0.09	92
3	0	3.92	0.95	0.02	21
4	0.01	1.00	0.78	0.13	135
5	0.14	4.31	0.95	0.02	23
6	0.05	76.26	0.81	0.06	56
7	0.1	13.24	0.83	0.12	118
8	0.13	8.08	0.95	0.06	64
9	0.03	4.22	0.87	0.13	129
10	0.01	0.39	0.56	0.06	52
11	0	1.35	0.9	0.02	21
12	0	0.18	0.69	0.07	66
13	0.03	1.29	0.88	0.07	69
14	0.22	2.52	0.99	0	8
15	0.01	0.83	0.96	0.01	16
Gomez	0.00	3.58	0.93	-	

Table 4.2 Estimated parameters for podemos

cluster	α	β	τ	π	users
1	0.1	0.66	0.96	0.08	89
2	0.14	2.27	0.93	0.06	59
3	0.59	4.49	0.99	0.02	26
4	0.03	0.81	0.98	0.12	114
5	0.02	1.09	0.91	0.13	131
6	0.1	2.70	0.78	0.06	62
7	0.1	3.39	0.99	0.05	54
8	0.01	81.89	0.98	0.03	26
9	0.03	2.84	0.8	0.08	77
10	0	4.12	0.99	0.04	39
11	0.4	12.16	0.95	0.02	19
12	0.07	9.05	0.85	0.05	54
13	0	0.1	0.43	0	8
14	0.02	0.93	0.76	0.13	128
15	0.06	5.13	0.96	0.12	120
Gomez	0.06	2.71	0.93	-	

Table 4.3 Estimated parameters for gameofthrones

13 of `gameofthrones`, made of 8 users whose only predictive parameter is the recency — they tend to penalize old posts less than the other users. Note that the closer to zero are the parameters, the more random the behavior—degree, recency or root posts would not have any effect and all posts would have the same probability of being chosen as parent.

4.3.3 Structural properties

After estimating the parameters for *gomez* and *lumbreras* we generated 10,000 synthetic threads with each in order to see whether there are structural difference 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.

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

The results are shown in Figure 4.5. We observe that the ability to reproduce real structures is very similar in both models. This is not entirely unsurprising since, as we said above, only posts written by the top 1,000 have different parameters than the *gomez* model.

4.3.4 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

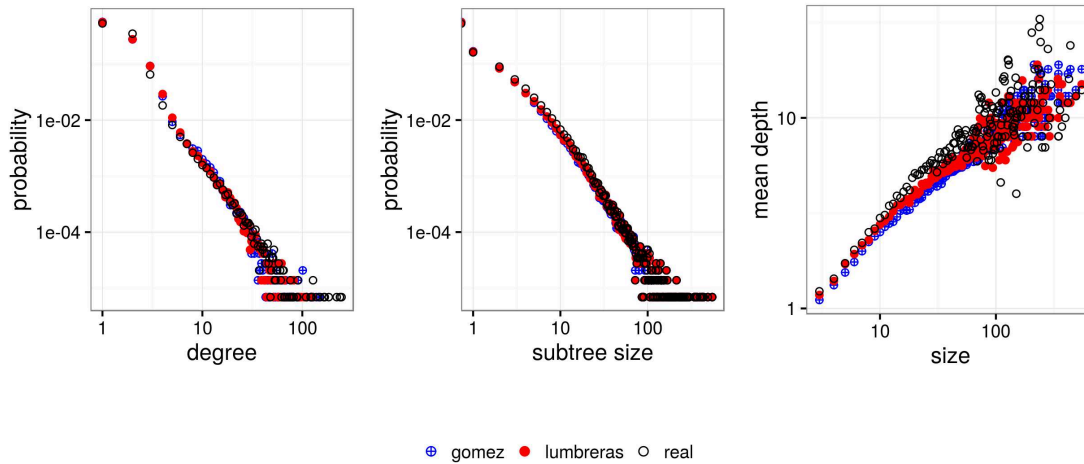


Figure 4.5 Properties of synthetic trees and real trees in podemos

gomez. We took three different metrics (likelihood of the test observations, percentage of hits and ranking error) and compared the two models in each clusters. 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 (*tau*).

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 4.11. For *lumbreras*, we first get the most likely cluster of the author

$$z'_i = \arg \max_k p(z_{ik} | \mathbf{X}, \boldsymbol{\theta}_k) \quad (4.32)$$

and then apply the parameters of the cluster z'_i to the same equation.

Figure 4.6 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 6 in *podemos* and cluster 8 in *gameofthrones*), are specially predictable; the small group of users in cluster 13 of *gameofthrones* (the ones with the smallest α and β) are also very predictable in terms of the likelihood of their choices. Even if these are also the clusters with better likelihood in *gomez* the improvement in *lumbreras* is bigger.

Hits. We define as a *hit* when the chosen parent was the most likely parent according to the model. Figure 4.7 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

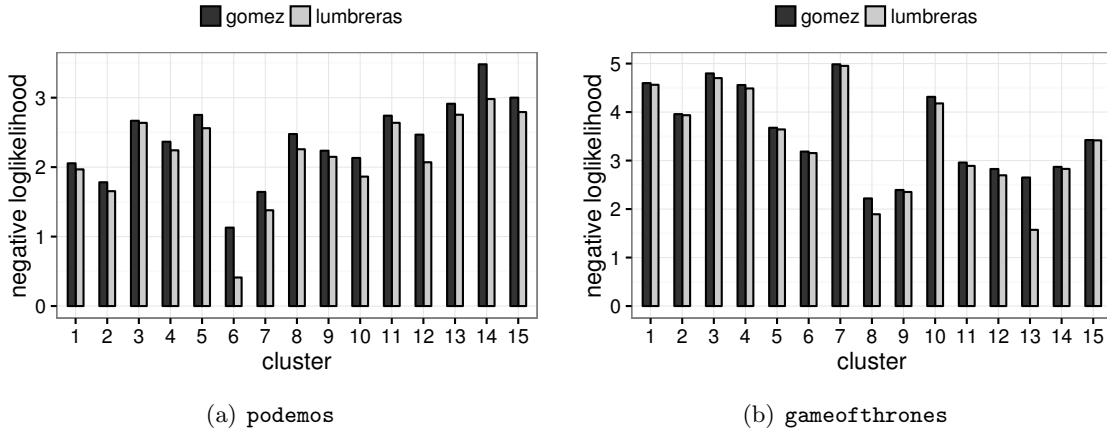


Figure 4.6 Mean negative log-likelihoods per cluster in the test set (lower is better)

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 *tau* model is always the worst model except for the cluster with the lowest β (clusters 10 and 12 in *podemos* and cluster 13 in *gameofthrones*), where *tau* 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*.

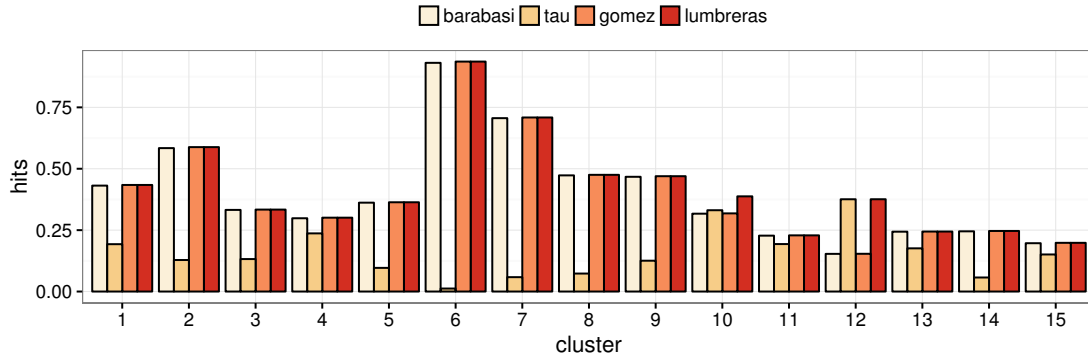
Normalized Ranking Error. Our *hits* metric only considers whether the model did a perfect prediction. Yet it is interesting to give some score, for instance, to *almost-perfect* predictions. If the chosen parent was given the second highest likelihood in a very long thread, we might give assign a near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

$$NRE = \frac{r - 1}{l - 1} \quad (4.33)$$

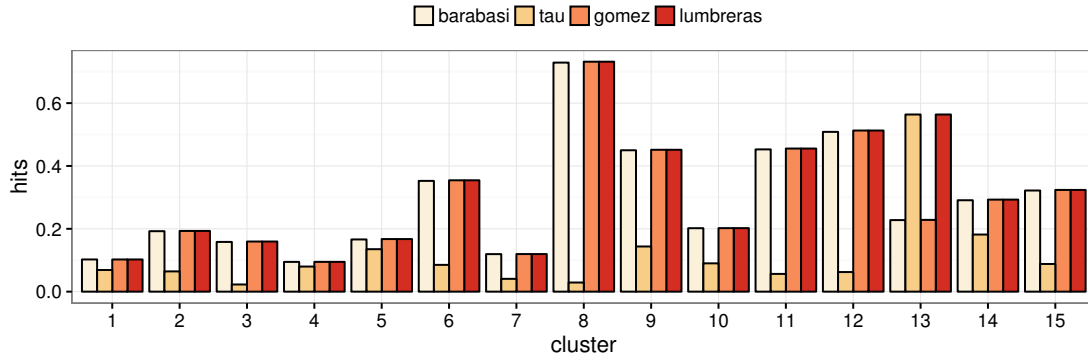
where r is the position of the chosen parent in the predicted ranking and l is the length of the thread, or the number of parents to choose from ($1 \leq r \leq l$).

While *hits* are low for almost every cluster, the ranking error shows a more optimistic picture (Figure 4.8): the medians for *gomez* and *lumbreras* are clearly better than those of the reference models. Yet, although for clusters 12 and 14 in *podemos* and for clusters 3, 4, 11 and 13 in *gameofthrones* the median score in *lumbreras* is slightly better than in *gomez*, there is barely any difference for the rest of clusters.

Finally, Figure 4.9 shows how thread sizes affect the accuracy of the models. We see that the longer the thread, the easier for *tau* to make better predictions and the harder



(a) podemos



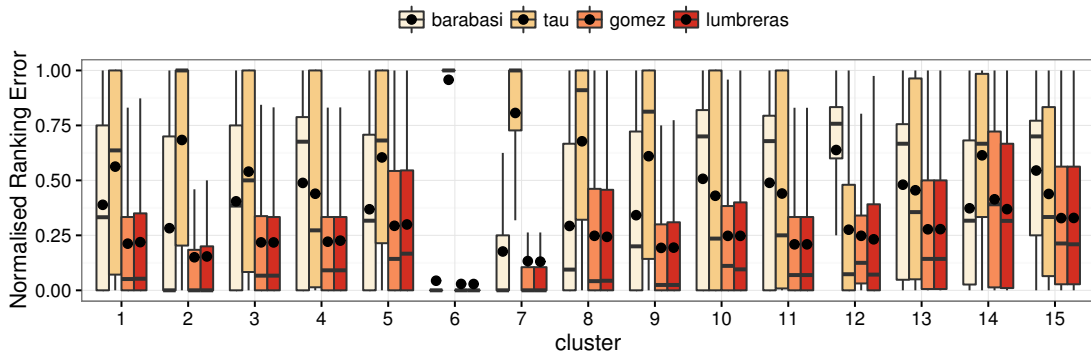
(b) gameofthrones

Figure 4.7 Hits per cluster in the test set

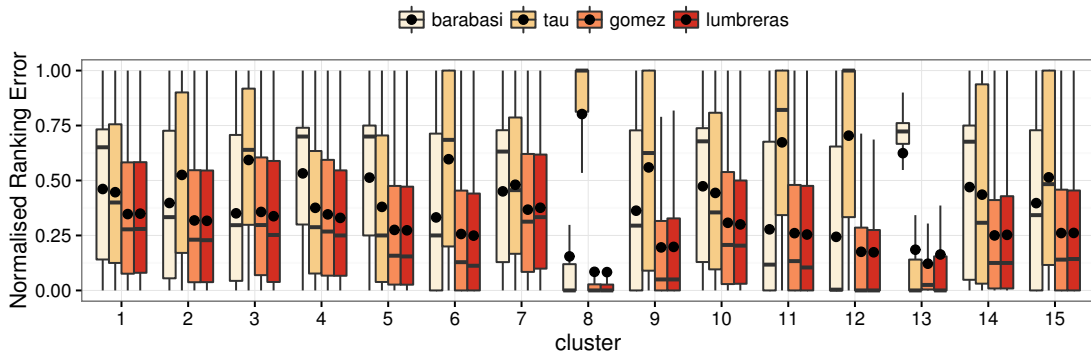
for *barabasi*. In other words, the longer the thread, the less importance of the degree and the more importance of the recency—until 75 posts where it gets stable. *gomez* and *lumbreras* are almost equivalent for all sizes.

To summarize, we showed that our model, which infers groups of users with different behavioral, gets better likelihoods than *gomez* when measured over unobserved behaviors. This supports the hypothesis that users behave, to some extent, following different behavioral functions and that they keep some consistency in their behaviors. That is, there is some role structure. Moreover, for some groups of users with outlier behaviors, our model is able to make better predictions 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.

4. ROLE DETECTION BASED ON THREAD GROWTH MODELS



(a) podemos



(b) gameofthrones

Figure 4.8 Normalized Ranking Error per cluster in the test set. Boxplots and means (black points).

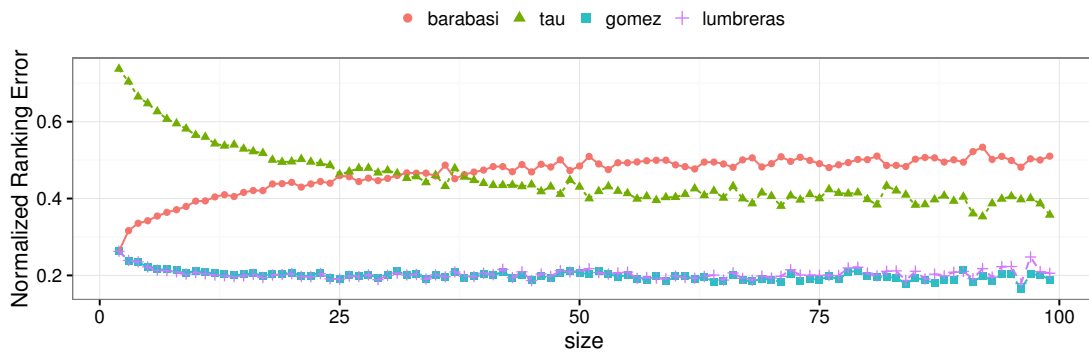


Figure 4.9 Normalized Ranking Error by size of thread in podemos

4.4 Summary

In this chapter, 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 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). Nonetheless, this improvement in the likelihood is not enough to make better predictions. Indeed, in terms of practical predictions of *which post will be the next to be replied*, our role-based model makes no better predictions —for most roles— than a model without roles.

Based on that, our conclusion is that our proposed concept of behavioral role has some descriptive power but that the predictive power is almost marginal. It might be that consistency in the 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 chapter are only able to capture a small part of this behavioral signal.

5 *Role detection based on features and latent behavioral functions with dual-view mixture models*

The whole is simpler than its parts.

Josiah Willard Gibbs

In this chapter, we present a dual-view mixture model to cluster users based on their features and latent behavioral functions. The model infers the groups of users as well as their latent behavioral functions. We also propose a non-parametric version based on a Dirichlet Process to automatically infer the number of clusters. We test the properties and performance of the model on a synthetic dataset that represents the participation of users in the threads of an online forum. We show that dual-view models outperform single-view ones when one of the views lacks information.

Contents

5.1	Introduction. Why a dual-view model?	82
5.2	Related models	82
5.3	Model description	84
5.3.1	Mixture models	84
5.3.2	Dual-view mixture model	85
5.3.3	Infinite number of clusters	87
5.4	Application to role detection in online forums	88
5.4.1	Feature view	89
5.4.2	Behavior view	90
5.4.3	Shared parameters	91
5.5	Inference	91
5.5.1	Predictive distribution	92
5.6	Experiments	92
5.6.1	Compared models	93
5.6.2	Metrics	94
5.6.3	Agreement between views	95
5.6.4	Disagreement between views	97
5.6.5	Iris dataset	99
5.6.6	Computational cost	101
5.6.7	Summary of the experiments	102
5.7	Summary	103

5.1 Introduction. Why a dual-view model?

A common issue we have had to deal with in previous chapters is the scarcity of user data. To assert that a user belongs to a given cluster requires enough data from the user so that we can be confident that her observed behaviors are really a trait of her role and not just a casual behavior without any relevant meaning. We have faced this problem when clustering users based on feature vectors extracted from the structure of conversations where they participate, and also when clustering users based on their—behavioral—parameters in a stochastic process that models user replies in conversation threads. To overcome that issue we propose, in this chapter, a dual-view model that integrates both approaches.

The dual-view model clusters users based on these two types of information—observed features and latent behavioral functions—to infer user clusters that are more robust and meaningful. The goal is to infer the behavioral function of each user and a clustering of users that takes into account both their features and their behavioral functions. In our model, users in the same cluster are assumed to have similar features *and* behavioral functions, and thus the inference of the clusters depends on the inference of the behavioral functions, and vice versa. The model prioritizes user partitions where each group has similar features *and* similar behavioral functions.

In this chapter, we have tested the model on synthetic data in order to evaluate and understand its properties. Both the features and the behavior coefficients in our data are drawn from Gaussian distributions. Nonetheless, the model may be applied to the neighborhood-based features of Chapter 3 and the reply-based behaviors of Chapter 4 if the probability distributions are adapted. In that case, the underlying assumption would be that users who tend to appear in the same type of neighborhood have also similar α , β and τ parameters.

5.2 Related models

Latent behavioral functions are used to model individual behaviors such as pairwise choices over products, reactions to medical treatments or user activities in online discussions. The inclusion of latent behavioral functions in clustering methods has several potential applications. On the one hand, it allows richer clusterings in settings where users, besides being described through feature vectors, also perform some observable actions that can be modeled as the output of a latent function. On the other hand, by assuming that users in the same cluster share similar latent functions, it may leverage inference of these functions. In the case, for instance, of recommender systems, this may be used to alleviate the *cold-start problem*—the fact that we cannot infer user preferences until they have interacted with the system for a while— if we have a set of features describing user profiles. In that context, a user with low activity will be given the same cluster as those users with similar features. Then, the inference of its behavioral function (e.g.: its preferences) will be based on the behavioral functions of the users in the same cluster.

One of the difficulties in dealing with latent behavioral functions is that, since these functions are latent, they are not representable in a feature-like way and therefore traditional clustering methods are not directly applicable. Our approach is to think of features and behaviors as two different *views* or representations of users, and to find the partition that is most consensual between the different views. In this sense, this is a multi-view clustering problem (Bickel and Scheffer, 2004). However, the clustering in one of the views depends on latent variables that need to be inferred. In this regard, it has similarities to Latent Dirichlet Allocation when used for clustering (e.g., cluster together documents that belong to the same latent topics).

Multi-view clustering ranges from Kumar et al. (2011), which finds a consensual partition by co-training (Mitchell and Blum, 1998), to Greene and Pádraig (2009), which proposes a two-step multi-view clustering that allows both consensual groups and groups that only exist in one of the views. In Niu et al. (2012), the multi-view approach is presented as the problem of finding multiple cluster solutions for a single description of features.

In this chapter, we extend the idea of multi-view clustering to deal with cases where one of the views comprises latent functions that are only indirectly observed through their outputs. The proposed method consists of a dual-view mixture model where every component represents two probability densities: one over the space of features and the other over the space of latent behavioral functions. The model allows to infer both the clusters and the latent functions. Moreover, the inference of the latent functions allows to make predictions on future user behaviors. The main assumption is that users in the same cluster share both similar features and similar latent functions and that users with similar features and behaviors are in the same cluster. Under this assumption, we show that this dual-view model requires less examples than single-view models to make good inferences.

The idea of using similarities in one view to enhance inference in the other view is not new. In bioinformatics, Eisen et al. (1998) found evidence suggesting that genes with similar DNA microarray expression data may have similar functions. Brown et al. (2000) exploit this evidence to train a Support Vector Machine (SVM) for each functional class and predict whether an unclassified gene belongs to that class given its expression data as an input. Pavlidis et al. (2002) extend the method of Brown et al. by also exploiting evidence that similar phylogenetic profiles between genes suggested a same functional class as well (Pellegriani et al., 1999). Pavlidis et al. propose a multi-view SVM method that uses both types of gene data as input.

More recently, Cheng et al. (2014) applied multi-view techniques to predict user labels in social networks such as LinkedIn (e.g., engineer, professor) or IMDb (e.g., directors, actors). Their method lies in the maximization of an objective function with a co-regularization that penalizes predictions of different labels for users that are similar either in terms of profile features or in terms of graph connectivity.

In the context of preference learning, Bonilla et al. (2010) also work with the assumption that similar users may have similar preferences, and models this assumption via a Gaussian Process prior over user utility functions. This prior favors utility functions

that account for user similarity and item similarity. To alleviate the computational cost of this model, Abbasnejad et al. (2013) propose an infinite mixture of Gaussian Processes that generates utility functions for groups of users assuming that users in each community share one single utility function. The main difference between our model and Abbasnejad et al. (2013) is that their model clusters users only based on their utility functions, while ours considers user features as well. In short, ours is a multi-view clustering model that also infers latent behavioral functions, while theirs is a single-view model focused on the inference of latent functions.

The chapter is structured as follows: first, we briefly recall mixture models. Second, we present our model as an extension of classic mixture models. The description of the model ends up with a generalization to an infinite number of clusters, which makes the model non-parametric. We finally describe an application to cluster users in online forums and end the chapter with experiments on synthetic data to demonstrate the properties of the model.

5.3 Model description

In this section, we introduce our model through three sequential steps. First, we start with a simple mixture model. Second, we extend the mixture model to build a *dual-view* mixture model. And third, we extend the *dual-view* mixture model so that the number of clusters is automatically inferred.

5.3.1 Mixture models

When a set of observations x_1, x_2, \dots, x_n cannot be properly fitted by a single distribution, we may get a better fit by considering that different subsets of observations come from different component distributions. Then, instead of a unique set of parameters θ of a single distribution, we need to infer K sets of parameters $\theta_1, \dots, \theta_K$ of K components and the assignments z_1, \dots, z_n of individual observations to one of these components. The model can be expressed as follows:

$$\begin{aligned} x_i | z_i, \theta_{z_i} &\sim F(\theta_{z_i}) \\ z_i &\sim \text{Discrete}(\boldsymbol{\pi}) \end{aligned} \tag{5.1}$$

where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ contains the probability of belonging to each component and F is the likelihood function over the observations. In Bayesian settings it is common to add priors over these parameters, resulting in a model such as:

$$\begin{aligned} x_i | z_i, \theta_{z_i} &\sim F(\theta_{z_i}) \\ \theta_j &\sim G_0 \\ z_i | \boldsymbol{\pi} &\sim \text{Discrete}(\boldsymbol{\pi}) \\ \boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \end{aligned} \tag{5.2}$$

where G_0 is the *base distribution* and α the *concentration parameter*. Mixture models are mostly used for *density estimation*. Nonetheless, inference over \mathbf{z} allows to use them as *clustering* methods. In this case, every component is often associated to a cluster.

5.3.2 Dual-view mixture model

In this section, we present an extension of mixture models to account both for features and latent behavioral functions. We denote by *behavioral functions* any function which, if known, can be used to predict the behavior of a user in a given situation. In the context of preference learning (Bonilla et al., 2010; Abbasnejad et al., 2013) or recommender systems (Cheung et al., 2004), behavioral functions may indicate hidden preference patterns, such as utility functions over the items, and the observed behavior may be a list of pairwise choices or ratings. In the context of online forums, behavioral functions may indicate the reaction of a user to a certain type of post and the observed behavior may be the set of replies to different posts. In general, behavioral functions are linked to observed behaviors through a likelihood function $p(y|f)$ where y represents an observation and f the latent behavioral function.

Let \mathbf{a}_u be the set of (observed) features of user u . Let f_u be a (latent) function of user u . Let y_u be the (observed) outcome of f_u . By slightly adapting the notation from last section we can describe the variables of our dual model as follows:

$$\begin{aligned}
 \mathbf{a}_u | z_u, \boldsymbol{\theta}_{z_u}^{(a)} &\sim F^{(a)}(\boldsymbol{\theta}_{z_u}^{(a)}) \\
 f_u | z_u, \boldsymbol{\theta}_{z_u}^{(f)} &\sim F^{(f)}(\boldsymbol{\theta}_{z_u}^{(f)}) \\
 y_u | f_u &\sim p(y_u | f_u) \\
 \boldsymbol{\theta}_j^{(a)} &\sim G_0^{(a)} \\
 \boldsymbol{\theta}_j^{(f)} &\sim G_0^{(f)} \\
 z_u | \boldsymbol{\pi} &\sim \text{Discrete}(\boldsymbol{\pi}) \\
 \boldsymbol{\pi} &\sim \text{Dirichlet}(\alpha)
 \end{aligned} \tag{5.3}$$

where we use the superindex (a) for elements in the *feature view* and the superindex (f) for elements in the latent functions view, henceforth *behavior view*. Otherwise, the structures are similar except for y_u , which represents the set of observed behaviors for user u . The corresponding Probabilistic Graphical Model is shown in Figure 5.1.

Every component has two distributions: one for features and one for latent behavioral functions. Latent behavioral functions are not directly observable, but they may be inferred through some observations if we have a likelihood function of observations given the latent functions.

The model can also be expressed in terms of a generative process:

- For every component k :

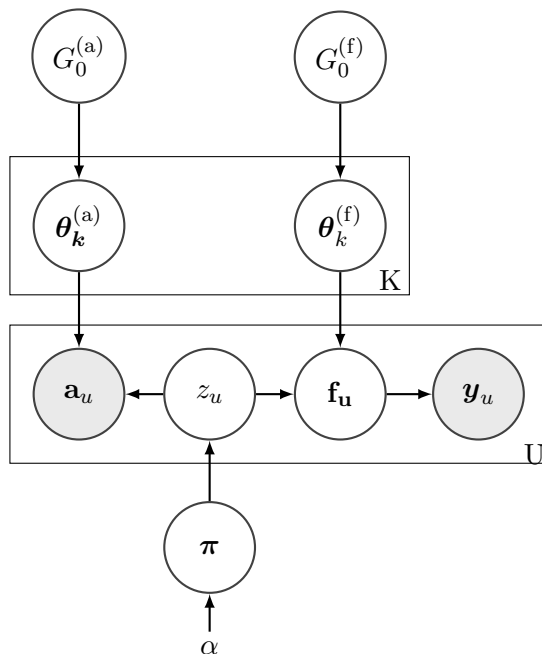


Figure 5.1 Graphical model of the generative process for U users and K clusters. Shaded circles represent observations and white circles represent latent variables. Views are connected through the latent assignments \mathbf{z} . A user u draws a feature vector \mathbf{a}_u and a behavior \mathbf{f}_u from the cluster indicated by z_u (the u -th element of \mathbf{z}).

- Draw feature and function parameters from their base distributions $\theta_k^{(a)} \sim G_0^{(a)}$ and $\theta_k^{(f)} \sim G_0^{(f)}$.
- Draw the mixture proportions $\boldsymbol{\pi} \sim \text{Dirichlet}(\boldsymbol{\alpha})$.
- For every user u :
 - Draw a component assignment $z_u \sim \text{Multinomial}(\boldsymbol{\pi})$.
 - Draw user features $\mathbf{a}_u \sim F^{(a)}(\boldsymbol{\theta}_{z_u}^{(a)})$.
 - Draw a user latent function $f_u \sim F^{(f)}(\boldsymbol{\theta}_{z_u}^{(f)})$.
 - Draw a set of observed behaviors $y_u \sim p(y_u | f_u)$.

Left and right branches of Figure 5.1 correspond to the *feature view* and the *behavior view*, respectively. Note that every component contains two sets of parameters, one for features and one for behaviors, so that the two views can be generated from the same component. This encodes our prior assumption that users who belong to the same cluster should be similar in both views.

Given the user assignments \mathbf{z} , variables in one view are conditionally independent from variables in the other view. That means their inferences can be considered separately. However, inference of \mathbf{z} uses information from both views. The conditional probability of \mathbf{z} given all the other variables is proportional to the product of its prior and the likelihood of both views:

$$p(\mathbf{z}|\cdot) \propto p(\mathbf{z}|\boldsymbol{\pi})p(\mathbf{a}|\boldsymbol{\theta}^{(a)}, \mathbf{z})p(\mathbf{f}|\boldsymbol{\theta}^{(f)}, \mathbf{z}) \quad (5.4)$$

The information given by each view is conveyed through the likelihood factors $p(\mathbf{a}|\boldsymbol{\theta}^{(a)}, \mathbf{z})$ and $p(\mathbf{f}|\boldsymbol{\theta}^{(f)}, \mathbf{z})$. The ratio between the conditional probability of a partition \mathbf{z} and the conditional probability of a partition \mathbf{z}' is:

$$\frac{p(\mathbf{z}|\cdot)}{p(\mathbf{z}'|\cdot)} = \frac{p(\mathbf{z}|\boldsymbol{\pi})}{p(\mathbf{z}'|\boldsymbol{\pi})} \frac{p(\mathbf{a}|\boldsymbol{\theta}^{(a)}, \mathbf{z})}{p(\mathbf{a}|\boldsymbol{\theta}^{(a)}, \mathbf{z}')} \frac{p(\mathbf{f}|\boldsymbol{\theta}^{(f)}, \mathbf{z})}{p(\mathbf{f}|\boldsymbol{\theta}^{(f)}, \mathbf{z}')} \quad (5.5)$$

where the contribution of each view depends on how much more likely \mathbf{z} is over the other assignments in that view. An extreme case would be a uniform likelihood in one of the views, meaning that all partitions \mathbf{z} are equally likely. In that case, the other view leads the inference.

The two views provide reciprocal feedback to each other through \mathbf{z} . This means that if one view is more confident about a given \mathbf{z} , it will not only have more influence on \mathbf{z} but also it will force the other view to re-consider its beliefs and adapt its latent parameters to fit the suggested \mathbf{z} .

Note also that inference of latent behavioral functions may be used for prediction of future behaviors.

5.3.3 Infinite number of clusters

So far, we have considered the number of components K to be known. Nonetheless, if we let $K \rightarrow \infty$ and marginalize over the mixture weights $\boldsymbol{\pi}$, it becomes a non-parametric model with a Dirichlet Process (DP) based on a Chinese Restaurant Process (CRP) prior over the user assignments, which automatically infers the number of *active* components (see the Appendix for the full derivation). Since we integrate out $\boldsymbol{\pi}$, user assignments are not independent anymore. Instead, the probability of a user u to be assigned to a non-empty (active) component k , given the assignments of all other users \mathbf{z}_{-u} , is:

$$p(z_u = k|\mathbf{z}_{-u}) \propto n_k \quad \text{for } k = 1, \dots, c \quad (5.6)$$

where c denotes the number of non-empty components and n_k is the number of users already assigned to the k -th component. The probability of assigning a user u to an empty (non-active) component, that would be labelled as $c + 1$, is:

$$p(z_u = k|\mathbf{z}_{-u}) \propto \alpha \quad \text{for } k = c + 1 \quad (5.7)$$

These two equations reflect a generative process that assigns users to clusters in a *rich get richer* manner. The more users in a component, the more attractive this component becomes. Empty components also have a chance of being filled. Despite the appearance of these equations, the idea behind the inference of \mathbf{z} remains the same. The only differences between the finite and the infinite cases are the number of components and the probabilities to be assigned to each component.

5.4 Application to role detection in online forums

The above model provides a general framework that can be adapted to many scenarios. In this section, we apply our model to the clustering of users in online forums —role detection. Although clustering based on user features may provide interesting insights (see Chapter 3), we think that the notion of *role* should include information that allows to predict behaviors (see Chapter 4). After all, as we have stated in previous chapters, this is what roles are useful for. We expect that a person holding a role behaves according to their role.

Let us specify the two views of our model for this scenario, given U users who participate in a forum composed of T discussion threads. For the feature view, we describe every user through a feature vector $\mathbf{a}_u = (a_{u1}, a_{u2}, \dots, a_{uD})^T$ that will typically contain features such as centrality metrics or number of posts. For the behavior view, we define a latent behavioral function that we call *catalytic power* and denote by b_u , which represents the ability of a user to promote long discussions; we refer to \mathbf{b} as the vector of all user catalytic powers. Let the *participation vector* of the discussion thread $\mathbf{p}_t = (p_{1t}, \dots, p_{Ut})^T$ be a binary array indicating which users participated among the first m posts of the thread t . Assuming that the dynamic of a discussion is strongly conditioned by the first participants, we model the final length of a thread y_t :

$$y_t \sim \mathcal{N}(\mathbf{p}_t^T \mathbf{b}, s_y^{-1})$$

where $\mathbf{p}_t^T \mathbf{b}$ is the cumulated catalytic power of users who participated in its first m posts and s_y represents the precision (inverse of the variance) of the unknown level of noise.

If the assumption that there exist groups of users sharing similar features and similar catalytic power holds, then our model will not only find a clustering based on features and behaviors (catalytic powers), but it will also exploit feature information to infer catalytic powers and, vice versa, the inferred catalytic powers will be treated by the model as an extra feature dimension.

Note that, unlike the model presented in Figure 5.1, the observed output y_t is common to all users who participated in the first m posts of thread t . Moreover, y_t depends on the observed participations \mathbf{p}_t . We also defined the noise factor s_y which was not explicitly present in the general model. The graphical model would be similar to that of Figure 5.1 but with the thread lengths \mathbf{y} , the participation matrix \mathbf{P} and the noise s_y out of the users plate. In the remaining of this section we provide more details about the components of the two views.

5.4.1 Feature view

In this section we specify the component parameters $\theta_k^{(a)}$ and the base distribution $G_0^{(a)}$ of the feature view. Our choice of priors follows that of the Infinite Gaussian Mixture Model (IGMM) as described by [Rasmussen \(2000\)](#) and [Görür and Rasmussen \(2010\)](#).

The feature vectors are assumed to come from a mixture of Gaussian distributions:

$$\mathbf{a}_u \sim \mathcal{N}\left(\boldsymbol{\mu}_{z_u}^{(a)}, \left(\mathbf{S}_{z_u}^{(a)}\right)^{-1}\right) \quad (5.8)$$

where the mean $\boldsymbol{\mu}_{z_u}^{(a)}$ and the precision $\mathbf{S}_{z_u}^{(a)}$ are component parameters common to all users assigned to the same component. The component parameters are given Normal and Wishart priors:

$$\boldsymbol{\mu}_k^{(a)} \sim \mathcal{N}\left(\boldsymbol{\mu}_0^{(a)}, \left(\mathbf{R}_0^{(a)}\right)^{-1}\right) \quad (5.9)$$

$$\mathbf{S}_k^{(a)} \sim \mathcal{W}\left(\beta_0^{(a)}, \left(\beta_0^{(a)} \mathbf{W}_0^{(a)}\right)^{-1}\right) \quad (5.10)$$

where the mean $\boldsymbol{\mu}_0^{(a)}$, the precision $\mathbf{R}_0^{(a)}$, the covariance $\mathbf{W}_0^{(a)}$, and the degrees of freedom $\beta_0^{(a)}$ are hyperparameters common to all components. The hyperparameters themselves are given non-informative priors centered at the observed data¹

$$\boldsymbol{\mu}_0^{(a)} \sim \mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a) \quad (5.11)$$

$$\mathbf{R}_0^{(a)} \sim \mathcal{W}(D, (D\boldsymbol{\Sigma}_a)^{-1}) \quad (5.12)$$

$$\mathbf{W}_0^{(a)} \sim \mathcal{W}(D, \frac{1}{D}\boldsymbol{\Sigma}_a) \quad (5.13)$$

$$\frac{1}{\beta_0^{(a)} - D + 1} \sim \mathcal{G}(1, \frac{1}{D}) \quad (5.14)$$

where $\boldsymbol{\mu}_a$ and $\boldsymbol{\Sigma}_a$ are the mean and covariance of all the features vectors and \mathcal{G} is the Gamma distribution. Note that, as pointed out in [Görür and Rasmussen \(2010\)](#), this choice of hyperparameters, which is equivalent to scaling the data and using unit priors, makes the model invariant to translations, rotations, and rescaling of the data.

Conjugate priors are chosen whenever possible to make the posteriors analytically accessible. As for $\beta_0^{(a)}$, the prior in Equation 5.14 guarantees that the degrees of freedom in the Wishart distribution in Equation 5.10 are greater than or equal to $D - 1$. The density $p(\beta_0^{(a)})$ is obtained by a simple transformation of variables (see Appendix).

¹Note that the expectation of a random matrix drawn from a Wishart distribution $X \sim \mathcal{W}(v, W)$ is $\mathbb{E}[\mathbf{X}] = v\mathbf{W}$. Our parametrization of the Gamma distribution corresponds to a one-dimensional Wishart. Its density function is therefore given by $\mathcal{G}(\alpha, \beta) \propto x^{\alpha/2-1} \exp(-\frac{x}{2\beta})$ and the expectation of a random scalar drawn from a Gamma distribution $x \sim \mathcal{G}(\alpha, \beta)$ is $\mathbb{E}[x] = \alpha\beta$.

5.4.2 Behavior view

In this section we specify the component parameters $\theta_k^{(f)}$ and the base distribution $G_0^{(f)}$ of the behavior view. Our choice corresponds to a Bayesian linear regression where coefficients are drawn not from a single multivariate Gaussian but from a *mixture* of one-dimensional Gaussians.

The thread lengths are assumed to come from a Gaussian distribution whose mean is determined by the catalytic power of users who participated in the first posts and whose variance represents the unknown level of noise:

$$y_t \sim \mathcal{N}(\mathbf{p}_t^T \mathbf{b}, s_y^{-1}) \quad (5.15)$$

where the precision s_y is given a Gamma prior centered at the sample precision σ_0^{-2} :

$$s_y \sim \mathcal{G}(1, \sigma_0^{-2}) \quad (5.16)$$

The power coefficients b_u come from a mixture of Gaussians:

$$b_u \sim \mathcal{N}\left(\mu_{z_u}^{(f)}, \left(s_{z_u}^{(f)}\right)^{-1}\right) \quad (5.17)$$

where the mean $\mu_{z_u}^{(f)}$ and the precision $s_{z_u}^{(f)}$ are component parameters common to all users assigned to the same component z_u . The component parameters are given Normal and Gamma priors:

$$\mu_k^{(f)} \sim \mathcal{N}\left(\mu_0^{(f)}, \left(r_0^{(f)}\right)^{-1}\right) \quad (5.18)$$

$$s_k^{(f)} \sim \mathcal{G}\left(\beta_0^{(f)}, \left(\beta_0^{(f)} w_0^{(f)}\right)^{-1}\right) \quad (5.19)$$

where the mean $\mu_0^{(f)}$, the precision $r_0^{(f)}$, the variance $w_0^{(f)}$, and the degrees of freedom $\beta_0^{(f)}$ are hyperparameters common to all components. Because the coefficients are not observed, we cannot center the hyperparameters in the data as we did in the feature view. Instead, we use their Maximum Likelihood Estimates, computed as $\hat{\mathbf{b}} = (\mathbf{P}\mathbf{P}^T + \lambda\mathbf{I})^{-1}\mathbf{P}^T\mathbf{y}$, with a regularization parameter $\lambda = 0.01$, where \mathbf{P} is the participation matrix $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_T\}$. Then the hyperparameters are given non-informative priors centered at $\hat{\mathbf{b}}$:

$$\mu_0^{(f)} \sim \mathcal{N}(\mu_{\hat{\mathbf{b}}}, \sigma_{\hat{\mathbf{b}}}^2) \quad (5.20)$$

$$r_0^{(f)} \sim \mathcal{G}(1, \sigma_{\hat{\mathbf{b}}}^{-2}) \quad (5.21)$$

$$w_0^{(f)} \sim \mathcal{G}(1, \sigma_{\hat{\mathbf{b}}}^2) \quad (5.22)$$

$$\frac{1}{\beta_0^{(f)}} \sim \mathcal{G}(1, 1) \quad (5.23)$$

where $\mu_{\hat{\mathbf{b}}}$ and $\sigma_{\hat{\mathbf{b}}}^2$ are the mean and the variance of the Maximum Likelihood Estimators $\hat{\mathbf{b}}$ of the coefficients. Note that this choice is data-driven and, at the same time, the risk of overfitting is reduced since hyperparameters are high in the model hierarchy.

5.4.3 Shared parameters

As for the common variables \mathbf{z} and α , \mathbf{z} is given a Chinese Restaurant Process prior:

$$p(z_u = k | \mathbf{z}_{-u}, \alpha) \propto \begin{cases} n_k & \text{for } k = 1, \dots, c \\ \alpha & \text{for } k = c + 1 \end{cases} \quad (5.24)$$

where c denotes the number of non-empty components before the assignment of z_u and n_k is the number of users already assigned to the k -th component. The concentration parameter α is given a vague inverse gamma prior:

$$\frac{1}{\alpha} \sim \mathcal{G}(1, 1)$$

5.5 Inference

The latent parameters of our model can be inferred by using Gibbs sampling, and sequentially taking samples of every variable given the others. Conditional distributions are detailed in the appendices. A single iteration of the Gibbs sampler goes as follows:

- Sample component parameters $\mathbf{S}_k^{(a)}, \boldsymbol{\mu}_k^{(a)}$ conditional on the indicators \mathbf{z} and all the other variables of the two views.
- Sample hyperparameters $\boldsymbol{\mu}_0^{(a)}, \mathbf{R}_0^{(a)}, \mathbf{W}_0^{(a)}, \beta_0^{(a)}$ conditional on the indicators \mathbf{z} and all the other variables of the two views.
- Sample component parameters $s_k^{(f)}, \mu_k^{(f)}$ conditional on the indicators \mathbf{z} and all the other variables of the two views.
- Sample hyperparameters $\mu_0^{(f)}, r_0^{(f)}, w_0^{(f)}, \beta_0^{(f)}$ conditional on the indicators \mathbf{z} and all the other variables of the two views.
- Sample coefficients \mathbf{b} conditional on the indicators \mathbf{z} and all the other variables of the two views.
- Sample s_y conditional on the indicators \mathbf{z} and all the other variables of the two views.
- Sample indicators \mathbf{z} conditional on all the variables of the two views.

Since we use conjugate priors for almost all the variables, their conditional probabilities given all the other variables are analytically accessible. The degrees of freedom $\beta_0^{(a)}, \beta_0^{(f)}$ and the concentration parameter α can be sampled by Adaptive Rejection Sampling (Gilks and Wild, 1992), which exploits the log-concavity of $p(\log \beta_0^{(a)} | \cdot)$, $p(\log \beta_0^{(f)} | \cdot)$ and $p(\log \alpha | \cdot)$ (see Appendix). As for the sampling of \mathbf{z} , the conditional probability of assigning user u to an active component k is proportional to the prior times the likelihoods:

$$p(z_u = k | \mathbf{z}_{-u}, \alpha, \cdot) \propto n_k p(\mathbf{a}_u | \boldsymbol{\mu}_k^{(a)}, \mathbf{S}_k^{(a)}) p(b_u | \mu_k^{(f)}, s_k^{(f)}) \quad \text{for } k = 1, \dots, c \quad (5.25)$$

and for the conditional probability of assigning z_u to a non-active component:

$$\begin{aligned} p(z_u = k | \mathbf{z}_{-u}, \alpha, \cdot) \propto & \alpha \int p(\mathbf{a}_u | \boldsymbol{\mu}_k^{(a)}, \mathbf{S}_k^{(a)}) p(\boldsymbol{\mu}_k^{(a)}) p(\mathbf{S}_k^{(a)}) d\boldsymbol{\mu}_k^{(a)} d\mathbf{S}_k^{(a)} \\ & \times \int p(b_u | \mu_k^{(f)}, s_k^{(f)}) p(\mu_k^{(f)}) p(s_k^{(f)}) d\mu_k^{(f)} ds_k^{(f)} \quad \text{for } k = c + 1 \end{aligned} \quad (5.26)$$

Unfortunately, these integrals are not analytically tractable because the product of the factors does not give a familiar distribution. Neal (2000) proposes to create m auxiliary empty components with parameters drawn from the base distribution, and then computing the likelihoods of \mathbf{a} and \mathbf{b} given those parameters. The higher the m , the closer we will be to the real integral and the less autocorrelated the cluster assignments will be. However, the equilibrium distribution of the Markov Chain is exactly correct for any value of m . To speed up the computations, m is usually small. For our experiments, we chose $m = 3$. That is, we generate 3 additional empty tables each one with its own parameters $\boldsymbol{\mu}', \mathbf{S}', \mu', s'$. We also run some experiments with $m = 4$ and $m = 5$, without observing significant differences neither in the clustering nor in the predictions, while it significantly increased the computational time. See Neal (2000) for a more systematic study on the effect of m .

5.5.1 Predictive distribution

We are also interested in the ability of the model to predict new thread lengths. The posterior predictive distribution over the length of a new thread is:

$$p(y_* | \mathbf{p}_*, \mathbf{P}, \mathbf{y}) = \int_{\boldsymbol{\theta}} p(y_* | \mathbf{p}_*, \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathbf{y}, \mathbf{P}) d\boldsymbol{\theta} \quad (5.27)$$

where \mathbf{p}_* is the participation vector of the new thread, and y_* its predicted length. If we have samples $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}$ from the posterior $p(\boldsymbol{\theta} | \mathbf{y}, \mathbf{P})$, we can use them to approximate the predictive distribution:

$$p(y_* | \mathbf{p}_*, \mathbf{P}, \mathbf{y}) \approx \frac{1}{N} \sum_{i=1}^N p(y_* | \mathbf{p}_*, \boldsymbol{\theta}^{(i)}) \quad (5.28)$$

where $\boldsymbol{\theta}^{(i)}$ are the i -th samples of \mathbf{b} and σ_y .

5.6 Experiments

We generated three scenarios to study in which situations dual-view models outperform single-view ones. The data reproduces the scenario of online forums presented in Section 5.4. In the first scenario, users belong to five clusters and those who belong to the

same cluster in one view also share the same cluster in the other view (*agreement between views*). In the second scenario, users belong to five clusters in the behavior view but two of the clusters are completely overlapped in the feature view (*disagreement between views*). In order to reproduce a more realistic clustering structure, in the last scenario user features and coefficients are taken from the *iris dataset*.

We will see in Section 5.6.6 that the main bottleneck of the model is the sampling of coefficients b_1, \dots, b_U since they are obtained by sampling from a U -dimensional Gaussian distribution that requires, at each Gibbs iteration, inverting a $U \times U$ matrix to get its covariance. This issue would disappear if the inference of the behavioral function parameters for a user were independent from the parameters of the other users. In this chapter, we use the *iris* dataset to demonstrate the properties of the model as a whole, without making any statement on the convenience of the presented behavioral functions.

5.6.1 Compared models

We compared two dual-view models and one single-view model. We call them `dual-fixed`, `dual-DP` and `single`. The `dual-fixed` corresponds to the present model where the number of clusters is set to the ground truth (five clusters). The `dual-DP` corresponds to the present model where the number of clusters is also inferred (Section 5.3.3). The `single` model corresponds to a Bayesian linear regression over the coefficients \mathbf{b} , which is equivalent to the behavior view specified in Section 5.4.2 where the number of clusters is set to one (that is, no clusters at all) and therefore there is no information flowing from the feature view to the behavior view; this model can only learn the latent coefficients \mathbf{b} .

Table 5.1 presents these three models as well as other related models that appear when blinding the models from one of the views. Note that we left out of the analysis those models that use clustering but are blinded of one view. The first two of these (IGMM and GMM), are regular clustering methods over feature vectors; we discarded them because they do not make inferences on latent behaviors. The last two (we call them latent-IGMM and latent-GMM) are Bayesian linear regressions where coefficients are assumed to come from a mixture of Gaussians; because these are in practice very similar to a simple Bayesian linear regression (they can be seen as Bayesian linear regressions with priors that tend to create groups of coefficients), we chose to benchmark only against the `single` model.

Posterior distributions of parameters are obtained by Gibbs sampling. We used the `coda` package in R (Plummer et al., 2006) to examine the traces of the Gibbs sampler. For the convergence diagnostics, we used Geweke’s test available in the same package. After some initial runs and examinations of the chains to see how long it took to converge to the stationary distribution for the dual models, we observed that convergence for all the models is usually reached before 5,000 samples. Since we run a large number of automatized experiments, we set a conservative number of 30,000 samples for every run, from which the first 15,000 are discarded as burn-in samples. For the first two experiments we initialized our samplers with all users in the same cluster. For the *iris* experiment we used the result of a k-means with 10 clusters over the feature view. We did

not systematically benchmark the two initialisation strategies. Nevertheless, this second strategy is, in general, more convenient in order to arrive to the true target distribution within less iterations.

Table 5.1 Compared and related models. Both single models are the same since if the number of clusters is fixed to one they cannot use the feature view. The row marked as – corresponds to a model that has no interest in this context since it simply finds the Gaussian distribution that best fits the observed features and makes neither clustering nor predictions.

	features	behaviors	clusters
dual-DP	yes	yes	∞
dual-fixed	yes	yes	fixed
single	yes	yes	1
IGMM	yes	no	∞
GMM	yes	no	fixed
-	yes	no	1
latent-IGMM	no	yes	∞
latent-GMM	no	yes	fixed
single	no	yes	1

5.6.2 Metrics

We used two metrics for evaluation, one for clustering (within dual models) and one for predictions of thread lengths (within the three models).

Metric for clustering: Clustering by mixtures models suffers from identifiability. The posterior distributions of \mathbf{z} has $k!$ reflections corresponding to the $k!$ possible relabelling of the k components. Due to this, different MCMC samples of \mathbf{z} may come from different reflections making it hard to average the samples. A common practice is to summarize the pairwise posterior probability matrix of clustering, denoted by $\hat{\pi}$, that indicates the probability of every pair of users to be in the same component (no matter the label of the component). In order to obtain a full clustering \mathbf{z} from $\hat{\pi}$, [Dahl \(2006\)](#) proposes a *least-squares model-based clustering* which consists of choosing as \mathbf{z} the sample $\mathbf{z}^{(i)}$ whose corresponding pairwise matrix has the smaller least-squares distance to $\hat{\pi}$:

$$\mathbf{z}_{LS} = \arg \min_{\mathbf{z} \in \mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N)}} \sum_i^U \sum_j^U (\delta_{i,j}(\mathbf{z}) - \hat{\pi})^2 \quad (5.29)$$

where $\delta_{i,j}(\mathbf{z})$ indicates whether i and j share the same component in \mathbf{z} . Finally, to assess the quality of the proposed clustering \mathbf{z}_{LS} we use the *Adjusted Rand Index*, a measure of pair agreements between the true and the estimated clusterings.

Metric for predictions: For predictions, we use the *negative loglikelihood*, which measures how likely the lengths are according to the predictive posterior:

$$p(y^{\text{test}} | \mathbf{p}_t^{\text{test}}, \mathbf{P}^{\text{train}}, \mathbf{y}^{\text{train}}) \quad (5.30)$$

and that can be approximated from Equation 5.28. Negative loglikelihoods are computed on test sets of 100 threads.

5.6.3 Agreement between views

To set up the first scenario, for a set of U users and five clusters we generated an assignment z_u to one of the clusters so that the same number of users is assigned to every cluster. Once all assignments \mathbf{z} had been generated, we generated the data for each of the views. For the feature view, every user was given a two-dimensional feature vector $\mathbf{a}_u = (a_{u_1}, a_{u_2})^T$ drawn independently from:

$$\mathbf{a}_u \sim \mathcal{N}(\boldsymbol{\mu}_{z_u}, \Sigma_a) \quad (5.31)$$

where $\boldsymbol{\mu}_{z_u} = (\cos(2\pi \frac{z_u}{5}), \sin(2\pi \frac{z_u}{5}))^T$ for $z_u = 1, \dots, 5$ (see Figure 5.2). For the behavior view, every user was given a coefficient drawn independently from:

$$b_u \sim \mathcal{N}(-50 + 25z_u, \sigma^2) \quad \text{for } z_u = 1, \dots, 5 \quad (5.32)$$

where coefficients for users in the same cluster are generated from the same Normal distribution and the means of these distributions are equidistantly spread in a $[-200, 200]$ interval (see Figure 5.2). To simulate users participations in a forum we generated, for each user u , a binary vector $\mathbf{p}_u = (p_{u1}, \dots, p_{uT})^T$ of length T that represents in which threads the user participated among the first m posts. We supposed each user participated in average in half the threads.

$$p_{ut} \sim \text{Bernoulli}(0.5) \quad (5.33)$$

Finally, we assumed that the final length of a thread is a linear combination of the coefficients of users who participated among the first posts:

$$y_t \sim \mathcal{N}(\mathbf{p}_t^T \mathbf{b}, \sigma_y) \quad (5.34)$$

If both views agree and there is enough data for both of them, we expect dual models to find the true clusters and true latent coefficients, and the single model to find also the true latent coefficients. In this case, the feature view brings no competitive advantage when there is enough information in the behavior view (and conversely, dual models should not outperform simple IGMM and GMM for the clustering task since there is enough information in the feature view).

On the contrary, when one of the views lacks information, then dual-view models should outperform single-view ones. In our model, the lack of information may come

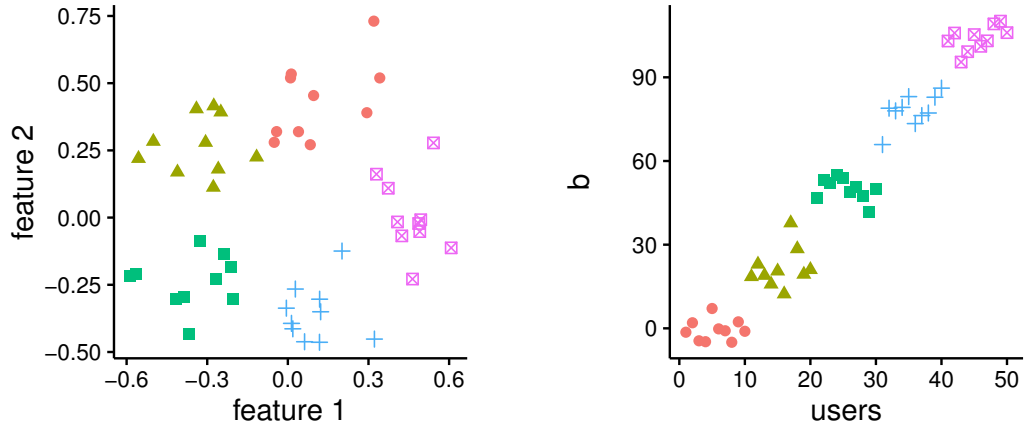


Figure 5.2 Dataset for agreement between the views. User features (left) and user coefficients (right). Every group (shape) of users has a well differentiated set of features and coefficients.

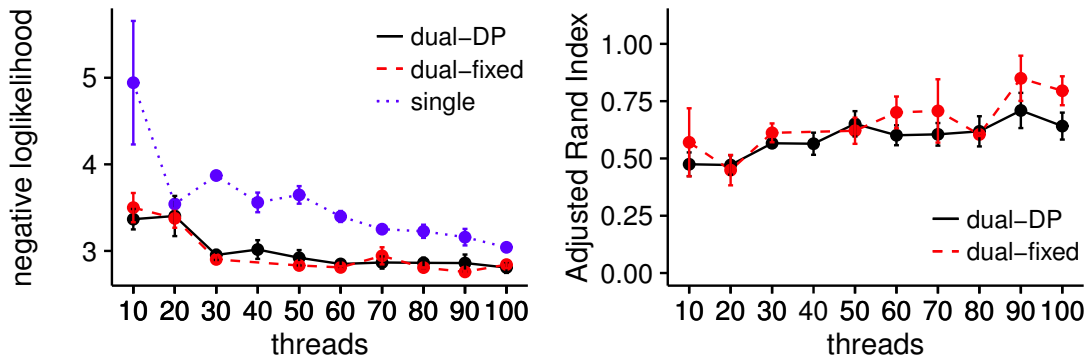


Figure 5.3 Results for agreement between the views. Comparison of models under different threads/users ratios (50 users and variable number of threads). Means and standard errors over 5 runs.

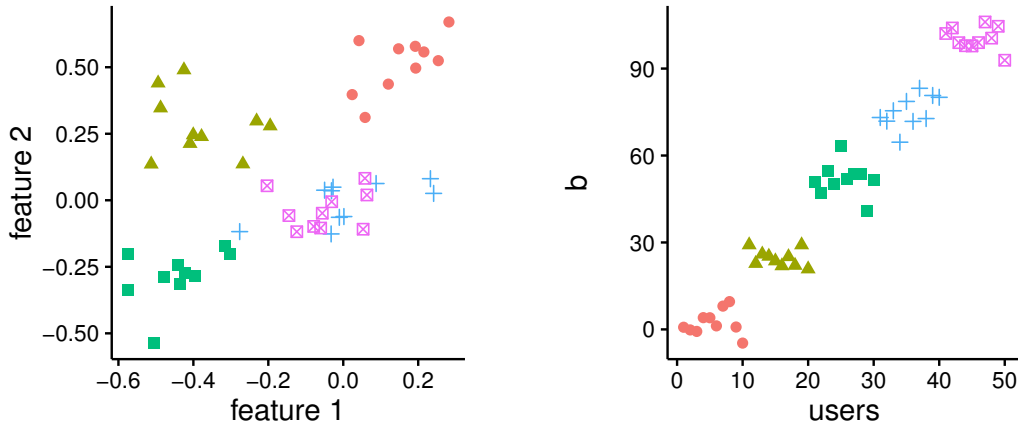


Figure 5.4 Dataset for disagreement between the views. User features (left) and user coefficients (right). Two of the groups (shapes) of users have similar features but different coefficients.

either from having too few threads to learn from or from having a high ratio of users versus threads since we have too many user behavior coefficients to learn.

Figure 5.3 shows how increasing the threads vs users ratio affects the accuracy of each model. When the number of threads is too low with respect to the number of users neither view has enough information and thus the three models make bad predictions the inference is difficult for the three models. Yet, dual-view models need less threads than the single model to make good inferences. When the number of threads is high enough, the three models tend to converge.

The number of users and threads in the experiments ranges from 10 to 100 users and from 10 to 100 threads. We avoided larger numbers to prevent the experiments from taking too long. 30,000 iterations of the Gibbs sampler described in Section 5.5 for 50 users and 100 threads take around three hours in a Pentium Intel Core i7-4810MQ @2.80GHz. Nevertheless, the ratio users/threads remains realistic. In the real forums that we analyzed from www.reddit.com a common ratio is 1/10 for a window of one month.

5.6.4 Disagreement between views

If each view suggests a different clustering \mathbf{z} , dual models should find a consensus between them (recall Equation 5.4). We generated a new dataset (Figure 5.4) where there are four clusters according to the feature view and five clusters according to the behavior view.

Figure 5.5 shows the posterior distributions (over thread lengths and over pairwise clustering) when (a) the behavior view has more information (b) both views lack data (c) the feature view has more information. By *having more information* we mean that a view dominates the inference over the posterior of the clustering \mathbf{z} .

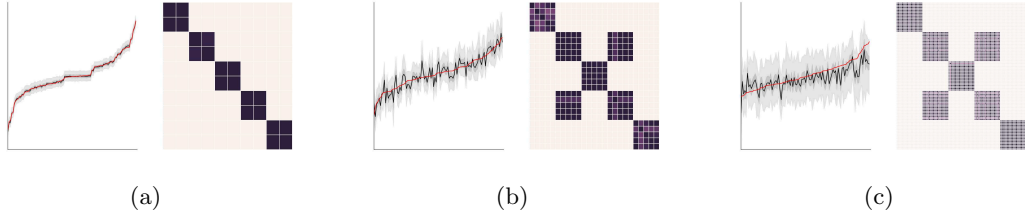


Figure 5.5 Posteriors for DP-dual when the two views see a different number of clusters. (a) 10 users and 100 threads. (b) 25 users and 25 threads. (c) 100 users and 10 threads. Figures on the left: examples of posterior distributions of thread length predictions over 100 test threads with 50% and 95% credible intervals in test set with 50 users and 10, 50 and 100 threads. x-axis correspond to threads sorted by their (true) length while y-axis correspond to predicted thread lengths. True lengths are plotted in red (smoothest lines). Figures on the right: examples of posterior pairwise clustering matrices $\hat{\pi}$. x-axes and y-axes correspond to the users. A dark point means a high probability of being in the same cluster. The worst case is (c), which makes a similar clustering to (b) but worse predictions, because the feature view receives misleading information from the behavior view and the number of threads is not high enough to compensate for it.

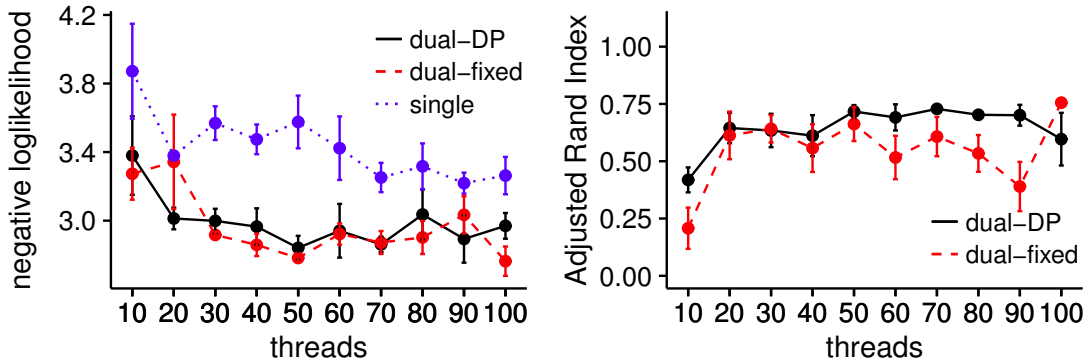


Figure 5.6 Results for disagreement between the views. Comparison of models under different threads/users ratios (50 users and variable number of threads) when the two views see a different number of clusters. Means and standard errors over 5 runs.

(a) Lack of information in feature view: If the number of users is low but they participated in a sufficient number of threads, the behavior view (which sees five clusters) will be stronger than the feature view (which sees four clusters) and will impose a clustering in five groups. User coefficients (used for thread length predictions) are also well estimated since the number of threads is high enough to learn them despite the lack of help from the other view (Figure 5.5a).

(b) Lack of information in both views: In the middle ground where neither view has enough evidence, the model recovers four clusters and the predictive posterior over thread lengths gets wider though still making good inferences (Figure 5.5b).

(c) Lack of information in behavior view: If the number of users is high and the number of threads is low, the feature view (four clusters) will have more influence in the posterior than the behavior view (five clusters), (Figure 5.5c). Predictions get worse because the model imposes a four clusters prior over coefficients that are clearly grouped in five clusters.

In order to compare between the performance in case of agreement and the performance in case of disagreement, we repeated the experiments of the last section with the current configuration. Figure 5.6 shows the performance of the models for 50 users and a different number of threads. While predictions and clustering improve with the number of threads, clustering with a small number of threads is worse in case of disagreement since the feature view imposes its 4 clusters vision. To recover the five clusters we would need either more threads or less users.

For the predictions, the dual models still outperform the single one because the feature view mostly agrees with the behavior view except for the users in one of the clusters. If all user features were in the same cluster, (no clustering structure) the performance of the predictions would be similar for the three models since the feature view would add no extra information. If we randomize the features so that, for instance, there are five clusters in the feature view that are very different from the clusters in the behavior view, we may expect the dual-view models to give worse predictions than the single-view one in those cases where they now perform better. In those cases, dual-models would be getting noise in the feature view (or very bad priors) and only a large enough number of threads could compensate for it.

5.6.5 Iris dataset

To reproduce a more realistic clustering structure we performed a last experiment based on the *iris* dataset. We used the *iris* data available in R, which corresponds to the original dataset reported in Anderson (1935). In our experiment, features correspond to three of the features of the *iris* dataset (Figure 5.7). We chose three out of the four features (sepal width, petal length and petal width) as well as a random subset of 50 observations so that the clustering task is harder if we only use the feature view. The coefficients of the behavior view are similar to those used in the former experiments. We

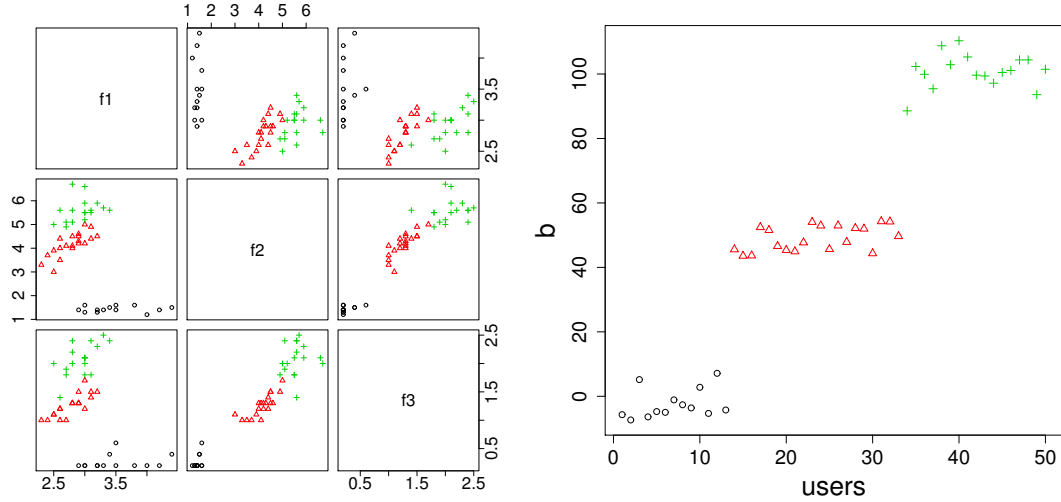


Figure 5.7 Iris dataset. User features (left) and synthetic user coefficients (right)

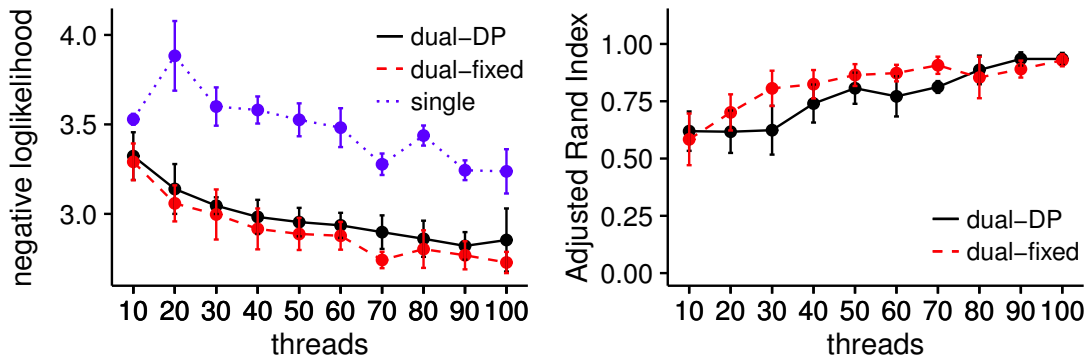


Figure 5.8 Results for the iris dataset. Comparison of models under different threads/users ratios (50 users and variable number of threads). Means and standard errors over 5 runs.

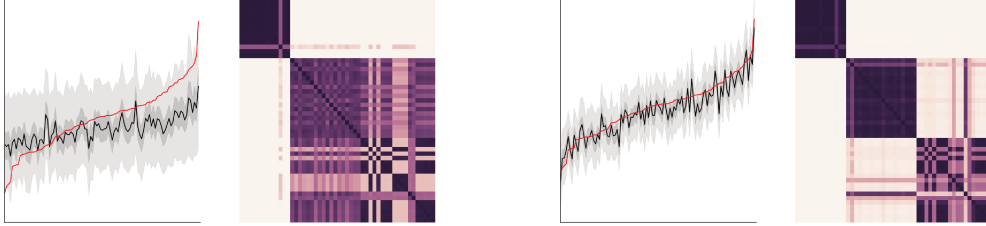


Figure 5.9 Predictive posteriors given by the DP-dual model over thread lengths and pairwise clustering matrices with 50 users and a training set of 10 threads (left) and 100 threads (right). Shaded areas indicate 95% and 50% credible intervals. True lengths are plotted in red (smoothest lines). As the number of thread increases, the model starts to see the three cluster structure as well as to make better predictions.

run a standard EM-GMM, from the R package `mclust` (Fraley et al., 2012), over the features to have an idea of what we should expect from our model when there are almost no threads and the coefficients are difficult to infer. We also run the same EM-GMM over the features and the true coefficients to have an idea of what we should expect from our model when the number of threads is high enough to make a good inference of the coefficients. This gave us an ARI of 0.48 and 0.79, respectively. Indeed, starting nearer to 0.48 when the number of threads is small, our model gets closer to 0.79 as we keep adding threads (Figure 5.8). Of course, the inference of the coefficients and thus the predictions over the test set also improve by increasing the number of threads. Since the single model does not take profit of the feature view, it needs more threads to reach the same levels than its dual counterparts. Figure 5.9 shows two examples of confusion matrices and predicted lengths for 10 and 100 threads.

Figure 5.10 shows the histogram of the number of clusters within the MCMC chain and the distribution of cluster sizes. The model infers three clusters but it also places some probability over a two clusters structure due to the closeness of two of the clusters in the feature view.

5.6.6 Computational cost

We analyzed the computational cost of the dual-DP model since it is the most complex of the three compared. Unlike the `single` model, it makes inferences in the two views, meaning about twice the number of variables. And unlike the `fixed-dual`, it has to infer the number of cluster and does it by creating m empty candidate clusters every time we sample a cluster assignment for a user at each iteration of the Gibbs sampler. This means creating $U \times iterations \times m$ empty clusters and computing, as many times, whether a user belongs to one of these auxiliary clusters (m possible extra clusters at each iteration), which makes it the slowest of the three models in terms of time per iteration.

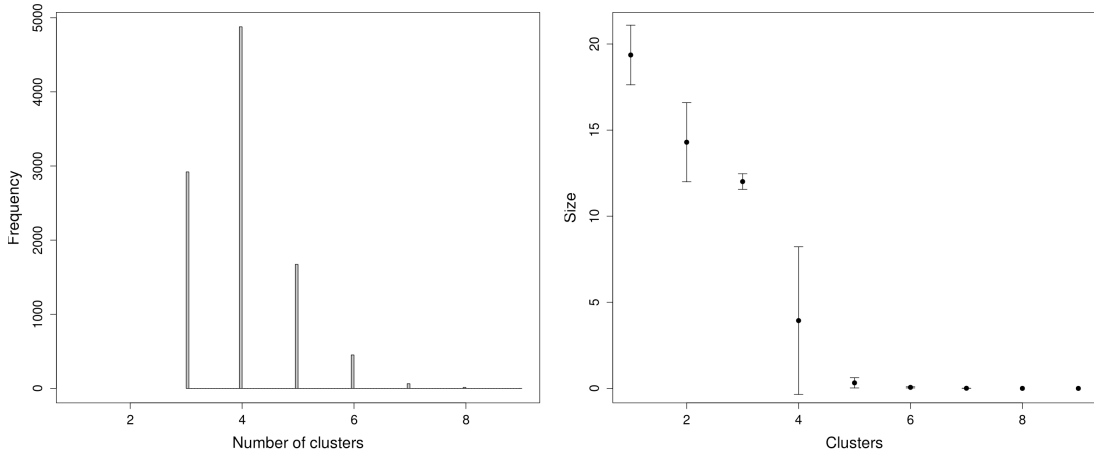


Figure 5.10 Left: histogram of the number of active clusters estimated by the DP-model in the *iris* scenario with 50 threads. Right: mean and standard deviations of the number of users assigned to each cluster during the chain. Most users are assigned to the three major clusters and a small group of users is assigned to a fourth cluster.

We look at the autocorrelation time to estimate the distance between two uncorrelated samples:

$$\tau = 1 + 2 \sum_{n=1}^{1000} |\rho_n|$$

where ρ_n is the autocorrelation at lag n . The variable with the higher autocorrelation is $\mu_0^{(f)}$ which has an autocorrelation time of 79. Since we drop the first 15000 samples for burn-in, we get an Effective Sample Size of 189 independent samples.

The bottlenecks of the algorithm are computing the likelihoods of the features and the coefficients given a cluster assignment, and the sampling of the coefficients. The sampling of the coefficients is relatively slow because it implies sampling from a multivariate Gaussian distribution with a $U \times U$ covariance matrix (see Appendix). This is due to the fact that the coefficient of each user depends on the coefficients of the users who have co-participated in the same thread. Note that this bottleneck would disappear in other scenarios where the inference of the behavioral function of a user is independent from the other users.

5.6.7 Summary of the experiments

To summarize, the experiments show on the one hand that dual-view models outperform single-view ones when users can be grouped in clusters that share similar features and latent behavioral functions and, on the other hand, that even when this assumption is not true, as long as there is enough data, the inference will lead to a consensual

partition and a good estimation of latent functions. Indeed, each view acts as a prior, or a regularization factor, on the other view. Good priors improve inference, while bad priors misguide the inferences unless there is sufficient amount of evidence to ignore the prior.

5.7 Summary

We presented a dual-view mixture model to cluster users based on features and latent behavioral functions. Every component of the mixture model represents a probability density over two *views*: a feature view for observed user attributes and a behavior view for latent behavioral functions that are indirectly observed through user actions or behaviors. The posterior distribution of the clustering represents a consensual clustering between the two views. Inference of the parameters in each view depends on the other view through this common clustering, which can be seen as a proxy that passes information between the views. An appealing property of the model is that inference on latent behavioral functions may be used to make predictions of users future behaviors. We presented two versions of the model: a parametric one where the number of clusters is treated as a fixed parameter and a nonparametric, based on a Dirichlet Process, where the number of clusters is also inferred.

We have adapted the model to a hypothetical case of online forums where behaviors correspond to the ability of users to generate long discussions. We clustered users and inferred their behavioral functions in three datasets to understand the properties of the model. We inferred the posteriors of interest by Gibbs sampling for all the variables but two of them which were inferred by Adapted Rejection Sampling. Experiments confirm that the proposed dual-view model is able to learn with less instances than its single-view counterpart due to the fact that dual-view models use more information. Moreover, inferences with the dual-view model based on a Dirichlet Process are as good as inferences with the parametric model even if the latter knows the true number of clusters.

There are some enhancements that can be applied to the model. On the one hand, we can make the inference faster and more scalable by using other inference methods such as Bayesian Variational Inference (for a comprehensible introduction see, for instance [Bishop \(2006\)](#)). On the other hand, it may be interesting to consider latent functions at a group level, that is, that users in the same cluster share *exactly* the same latent behavior (as we considered in Chapter 4). Not only it would reduce the computational cost but, if we have few data about every user, a group-level inference may also be more grounded and statistically sound.

6 Contributions and perspectives

6.1 Contributions

In this thesis we set out to automatically detect roles in online discussion forums. We tried to stick as much as possible to the sociological definition of role as the *behavioral repertoire characteristic of a person or a position* (Biddle, 1979). Following this idea, we focused our efforts on conversational behaviors since conversations are the constituent behaviors of a forum. First, we have considered that two users hold the same role if they tend to participate in the same type of conversation. By taking into account their structural position in the motif of the conversation that embeds a post (a neighborhood), we aimed to capture the different ways users contribute to conversations where they participate. Indeed, we were able to show that we can detect different types of conversationalists.

Secondly, in order to take a more functional approach—in the sense of capturing the functions that drive user behaviors—we considered that two users hold the same role if their replying behavior—their criteria for choosing a post to reply—can be modeled by the same function. Or, more specifically, if their choices can be modeled by the same probability distribution. We showed that not all users behave similarly and that indeed modeling users as members of different subpopulations with different behavioral functions leads not only to a better fit of the data compared to a model that does not consider differences between the individuals—the model is more flexible—but also to better likelihoods in unobserved behaviors. Yet we also showed that this improvement is not enough to make better predictions, which leaves room for improvement of the model.

In an attempt to integrate the former two approaches, we proposed a third *dual-view* method that opens the door to the integration of features and behavioral functions. In particular, we modeled users features and behaviors as generated from a mixture model of roles, where users in the same component—the same role—have similar features and similar behaviors. We used synthetic data to show the properties of the model. In particular, we showed that a dual-view model can learn with fewer examples than models that use one single view (features as in our first definition, or behaviors as in our second one.).

6.2 Roles or not roles?

The aim of this thesis was to detect roles and to prove that these roles can be used as predictors of a user behavior. We showed that conversational behaviors, regarded from a merely structural point of view, are different for different groups of users. We stated that, in order to give these clusters the category of role, they needed to prove their predictive power. There is some evidence in favor of this—an increase in the likelihood

of unobserved behaviors—but the evidence is not strong enough and thus we do not dare to claim that we found *roles* from a grounded sociological basis.

Therefore, the question remains open. Are there *predictive* roles in online forums? There are certainly some users with a consistent extreme behavior, such as *trolls*, and specific methods have been proposed by some authors to detect them. Yet a finer-grained taxonomy might not be possible. One can blame the object—users are too random in online conversations, they have too much variance—or the measuring instrument—the method does not capture enough the non-random part. Or one can blame both.

As for the object, it might be good news that humans are not *always* that predictable—although, in many aspects, they are; we would have otherwise an army of jobless sociologists, statisticians and machine learners. As for the measure instrument, we propose some possible improvements in the following section.

6.3 Perspectives

As mentioned before, the role detection machinery can be improved in many ways. These are left as paths for future research. We divide them in methodological and technical.

6.3.1 Technical

Clustering based on count data

In Chapter 3, we chose hierarchical clustering to compare the clusters for each type of neighborhood and forum. The main object was a matrix of discrete distributions, where each distribution indicates the tendency of a user towards each type of conversation. We recall that, before obtaining that matrix, we had a matrix of counts —how many times a user was seen in each motif. As such, some specific methods can be used to deal with count data. For instance, we might model the vectors of counts as draws from a mixture of multinomial distribution. Then, we would infer clusters of users where users in the same cluster draw their vectors of counts from the same multinomial distribution. This model-based clustering can be easily used as the feature branch of our dual-view model.

A scalable dual-view model

In Chapter 5, we faced a serious problem of scalability in our dual-view model. The reason was two-fold. On the one hand, the need to compute the inverse of a matrix with as many rows and columns as users. This was required during the Gibbs Sampling to get the covariance matrix of the Normal distribution that generates the user coefficients (the matrix Λ'^{-1} to sample from $p(b_u|\cdot)$ in the Appendix A.3.1). This problem can be easily avoided if we choose to model individual behaviors instead of collective behaviors (the length of a thread), since the inference of a user parameters would be conditionally independent, given the clusters, of the other users. The behavioral model presented in 4 holds this condition. Unfortunately, its likelihood function has no conjugate priors and

thus we could not use Gibbs Sampling to sample the behavioral coefficients (α, β, τ) . Instead, we may use other sampling methods that do not require conjugacy.

On the other hand, Gibbs Sampling and Monte-Carlo methods are easily beaten in speed by Variational Inference methods. Replacing our Gibbs Sampler by a Variational Inference would likely boost the speed considerably. Although we would obtain approximations to the posterior—Variational Inference approximates the posterior with a function that is analytically tractable—the gain in scalability might be worth it.

6.3.2 Methodological

A dictionary of conversations

We showed in Chapter 3 that the different neighborhood definitions gave motifs that correspond, without much doubt, to the same type of conversation. Thus, some of them might be considered isomorphic. Even if we might apply a more aggressive pruning, we think that a more reasonable solution is to do this task manually so that we obtain a dictionary of meaningful conversational motifs. In other words, an orthogonal—as much as possible—base of conversations.

A better tree growth model

Our model in Chapter 4 considers that each group of users behaves following this behavioral function, borrowed from Gómez et al. (2012):

$$p(\pi_t = i | \boldsymbol{\pi}_{1:(t-1)}) \propto \alpha_{z(a(t))} d_i + \beta_{z(a(t))} r_i + \tau_{z(a(t))}^{l_i} \quad (6.1)$$

This 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. 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*. The likelihood would be:

$$p(\pi_t = i | \boldsymbol{\pi}_{1:(t-1)}) \propto \alpha_{z(a(t))} d_i + \beta_{z(a(t))} r_i + \tau_{z(a(t))}^{l_i} + \eta_{z(i)} \quad (6.2)$$

where γ_k would be the interestingness of users in cluster k . Note, however, that this model would increase in K (the number of clusters) the number of parameters.

Moreover, we might apply the idea of blockmodeling and consider that the interestingness depends on the groups of both users (the replier and the replied). For instance, everyone might be attracted by posts written by experts, and experts might only be attracted by posts from other experts. If we denote as $\eta_{k,k'}$ the interestingness of posts from group k' from the point of view of users in group k , we might express the likelihood as:

$$p(\pi_t = i | \boldsymbol{\pi}_{1:(t-1)}) \propto \alpha_{z(a(t))} d_i + \beta_{z(a(t))} r_i + \tau_{z(a(t))}^{l_i} + \eta_{z(a(t)), z(a(i))} \quad (6.3)$$

where the number of new parameters is $K \times K$.

Combining structure and text

In this thesis, we took into consideration only the structural aspect of conversations. We claimed that the subtleties of user language are —often— difficult to capture by current algorithms. Yet some textual analysis might be helpful if we combine it with structural analysis presented in this thesis.

We might, for instance, detect some basic textual features of the post and add them to the growth model of Chapter 4. An off-topic post, for instance, might be less attractive than a post that uses the key words of a particular forum. The length of a post might also be a good predictor. A basic sentiment analysis of posts to classify them into negative, neutral, and positive may give another feature. In Chapter 3, we might also add textual features to (carefully) increase our dictionary of motifs.

Bibliography

- Abbasnejad, E., S. Sanner, E. V. Bonilla, and P. Poupart (2013). Learning community-based preferences via Dirichlet process mixtures of Gaussian processes. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI '13*, pp. 1213–1219. AAAI Press.
- Adamic, L. A., J. Zhang, E. Bakshy, and M. S. Ackerman (2008). Knowledge sharing and yahoo answers. In *Proceeding of the 17th international conference on World Wide Web - WWW '08*, New York, New York, USA, pp. 665. ACM Press.
- 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.
- Airoldi, E. M., D. M. Blei, S. E. Fienberg, and E. P. Xing (2008, jun). Mixed Membership Stochastic Blockmodels. *Journal of Machine Learning Research* 9, 1981–2014.
- Anderson, E. (1935). The Irises of the Gaspé Peninsula. *Bulletin of the American Iris Society* 59, 2–5.
- 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.
- Anokhin, N., J. Lanagan, and J. Velcin (2012). Social Citation: Finding Roles in Social Networks. An Analysis of TV-Series Web Forums. In *Second International Workshop on Mining Communities and People Recommenders*, pp. 49–56.
- Backstrom, L., J. Kleinberg, L. Lee, and C. Danescu-Niculescu-Mizil (2013). Characterizing and Curating Conversation Threads: Expansion, Focus, Volume, Re-entry. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pp. 13–22.
- 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.
- Bickel, S. and T. Scheffer (2004). Multi-view clustering. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pp. 19–26.
- Biddle, B. (1986). Recent Developments in Role Theory. *Annual Review of Sociology* 12(1), 67–92.

- Biddle, B. J. (1979). *Role Theory: Expectations, Identities, and Behaviors*. New York: Academic Press.
- Birmelé, E. (2012). Detecting local network motifs. *Electronic Journal of Statistics* 6, 908–933.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc.
- Blondel, V. D., J.-L. Guillaume, R. Lambiotte, and E. Lefebvre (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* P10008(10), 1–12.
- Bonilla, E. V., S. Guo, and S. Sanner (2010). Gaussian process preference elicitation. In *Advances in Neural Information Processing Systems 23*, pp. 262–270. Curran Associates, Inc.
- Brown, M. P., W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America* 97(1), 262–267.
- 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.
- Cheng, Y., A. Agrawal, A. Choudhary, H. Liu, and T. Zhang (2014, dec). Social Role Identification via Dual Uncertainty Minimization Regularization. In *IEEE International Conference on Data Mining*, pp. 767–772. IEEE.
- Cheung, K. W., K. C. Tsui, and J. Liu (2004). Extended Latent Class Models for Collaborative Recommendation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*. 34(1), 143–148.
- Chou, B. and E. Suzuki (2010). Discovering community-oriented roles of nodes in a social network. In *Proceedings of the 12th international conference on Data warehousing and knowledge discovery*, pp. 52–64.

- Dahl, D. B. (2006). Model-based clustering for expression data via a Dirichlet process mixture model. In K.-A. Do, P. Müller, and M. Vannucci (Eds.), *Bayesian inference for gene expression and proteomics*, pp. 201–218. Cambridge University Press.
- Daudin, J. J., F. Picard, and S. Robin (2008). A mixture model for random graphs. *Statistics and Computing* 18(2), 173–183.
- Dorat, R., M. Latapy, B. Conein, and N. Auray (2007). Multi-level analysis of an interaction network between individuals in a mailing-list. *Annales des télécommunications* 62(3-4), 325–349.
- DuBois, C., C. T. Butts, and P. Smyth (2013). Stochastic blockmodeling of relational event dynamics. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, Volume 31, pp. 238–246.
- Duggan, M. (2015). Mobile Messaging and Social Media 2015. Technical Report August, Pew Research Center.
- Egghe, L. (2006). Theory and practise of the g-index. *Scientometrics* 69(1), 131–152.
- Eisen, M. B., P. T. Spellman, P. O. Brown, and D. Botstein (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the USA* 95(25), 14863–14868.
- Fisher, D., M. Smith, and H. T. H. Welser (2006, jan). You Are Who You Talk To: Detecting Roles in Usenet Newsgroups. In *HICSS'06: Proceedings of the 41st Hawaii International Conference on System Sciences*, pp. 59b–59b. IEEE.
- 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.
- Foucault, M. (2003). Abnormal. In *Lectures at the Collège de France 1974 - 1975*, London. Verso.
- Fraley, C., A. E. Raftery, T. B. Murphy, and L. Scrucca (2012). *mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation*. Department of Statistics, University of Washington.
- Fu, W., L. Song, and E. P. Xing (2009, jun). Dynamic mixed membership blockmodel for evolving networks. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, New York, New York, USA, pp. 1–8. ACM Press.
- Furtado, A., N. Andrade, N. Oliveira, and F. Brasileiro (2013). Contributor profiles, their dynamics, and their importance in five q&a sites. In *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW '13*, New York, New York, USA, pp. 1237. ACM Press.

- Gilks, W. R. and P. Wild (1992). Adaptive rejection sampling for {G}ibbs sampling. *Applied Statistics* 41(2), 337–348.
- Gleave, E., H. Welser, T. M. Lento, and M. A. Smith (2009). A Conceptual and Operational Definition of 'Social Role' in Online Community. In *2009 42nd Hawaii International Conference on System Sciences*, pp. 1–11. IEEE.
- Gliwa, B., A. Zygmunt, and J. Koźlak (2013). Analysis of roles and groups in blogosphere. *Proceedings of the 8th International Conference on Computer Recognition Systems 226*, 299–308.
- Goffman, E. (1959). *The presentation of self in everyday life* (4 ed.). London: Penguin Books.
- Golder, S. A. (2003). *A Typology of Social Roles in Usenet*. Ph. D. thesis, Harvard University.
- Golder, S. A. S. and J. Donath (2004). Social roles in electronic communities. In *Presented at the Association of Internet Researchers (AoIR)*, Volume 5, Brighton, pp. 1–25.
- Gómez, V., A. Kaltenbrunner, and V. López (2008, apr). Statistical analysis of the social network and discussion threads in slashdot. In *Proceeding of the 17th international conference on World Wide Web - WWW '08*, New York, New York, USA, pp. 645. ACM Press.
- 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.
- Görür, D. and C. E. Rasmussen (2010). Dirichlet process Gaussian mixture models: choice of the base distribution. *Journal of Computer Science and Technology* 25(July), 653–664.
- 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.
- Greene, D. and C. Pádraig (2009). Multi-View Clustering for Mining Heterogeneous Social Network Data. In *Workshop on Information Retrieval over Social Networks, 31st European Conference on Information Retrieval*.

-
- Henderson, K., B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li (2012). RolX: structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, New York, New York, USA, pp. 1231. ACM Press.
- Himmelboim, 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.
- Hirsch, J. E. (2005, nov). An index to quantify an individual’s scientific research output. *Proceedings of the National Academy of Sciences of the United States of America* 102(46), 16569–16572.
- Ho, Q., L. Song, and E. P. Xing (2011). Evolving Cluster Mixed-Membership Block-model for Time-Varying Networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS) 15*, 342–350.
- Holland, P. W., K. B. Laskey, and S. Leinhardt (1983). Stochastic blockmodels: First steps. *Social networks* 5, 109–137.
- Holland, P. W. P. and S. Leinhardt (1970). A method for detecting structure in sociometric data. *American Journal of Sociology* 76(3), 492–513.
- Kaltenbrunner, A., V. Gómez, and V. López (2007, oct). Description and Prediction of Slashdot Activity. In *2007 Latin American Web Conference (LA-WEB 2007)*, pp. 57–66. IEEE.
- Kan, A., J. Chan, C. Hayes, B. Hogan, J. Bailey, and C. Leckie (2013). A time decoupling approach for studying forum dynamics. *World Wide Web* 16(5-6), 595–620.
- Kemp, C., T. L. Griffiths, and J. B. Tenenbaum (2004). Discovering latent classes in relational data. Technical Report September, Massachusetts Institute of Technology.
- Kemp, C., J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda (2006). Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence*, Volume 1, pp. 381–388.
- Killick, R., P. Fearnhead, and I. A. Eckely (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association* 107, 1590–1598.
- Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data: Methods and Models*. New York: Springer.
- Kollock, P. (1998). *Communities in Cyberspace*. London: Routledge.

- Kumar, A., P. Rai, and H. Daume (2011). Co-regularized Multi-view Spectral Clustering. In *Advances in Neural Information Processing Systems*, pp. 1413–1421.
- 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.
- Labatut, V., N. Dugué, and A. Perez (2014). Identifying the community roles of social capitalists in the Twitter network. In *ASONAM 2014 - Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 371–374.
- Latouche, P., E. Birmele, and C. Ambroise (2012). Variational Bayesian Inference and Complexity Control for Stochastic Block Models. *Statistical Modelling* 2(1), 93–115.
- Linton, R. (1936). *The Study of Man*. New York: Appleton Century Crofts, Inc.
- Lorrain, F. F. and H. C. White (1971, jan). Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology* 1(1), 49–80.
- 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.
- Lumbreras, A., J. Lanagan, J. Velcin, B. Jouve, L. Eric, U. Lyon, and M. France (2013). Analyse des rôles dans les communautés virtuelles : définitions et premières expérimentations sur IMDb. In *Modèles et Analyses Réseau : Approches Mathématiques et Informatiques (MARAMI)*, pp. 1–12.
- Maia, M., J. Almeida, and V. V. Almeida (2008). Identifying user behavior in online social networks. In *Proceedings of the 1st Workshop on Social Network Systems*, New York, New York, USA, pp. 1–6. ACM Press.
- McCallum, A., X. Wang, and A. A. Corrada-Emmanuel (2007). Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research* 30(1), 249–272.
- McFarland, D. A., K. Lewis, and A. Goldberg (2016). Sociology in the Era of Big Data: The Ascent of Forensic Social Science. *American Sociologist* 47(1), 12–35.
- Mead, G. H. (1934). *Mind, self, and society*. USA: The University of Chicago Press.
- Milo, R., S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon (2002). Network motifs: simple building blocks of complex networks. *Science* 298(5594), 824–7.

-
- Mitchell, T. and A. Blum (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pp. 92–100.
- Nadel, S. (1957). *The theory of social structure*. Abingdon: Routledge.
- Nash, J. (1975). Bus riding: community on wheels. *Urban life and culture* 4, 99–124.
- Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics* 9(2), 249–265.
- Nelder, J., R. Mead, B. J. a. Nelder, and R. Mead (1965). A simplex method for function minimization. *Computer Journal* 7(4), 308–313.
- Niu, D., J. G. Dy, and Z. Ghahramani (2012). A nonparametric Bayesian model for multiple clustering with overlapping feature views. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pp. 814–822. JMLR.
- 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.
- Nowicki, K. and T. A. B. Snijders (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association* 96(455), 1077–1087.
- Parsons, T. (1951). *The Social System*. USA: The Free Press of Glencoe.
- Pavlidis, P., J. Weston, J. Cai, and W. S. Noble (2002). Learning gene functional classifications from multiple data types. *Journal of Computational Biology* 9(2), 401–411.
- Pellegrini, M., E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates (1999). Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proceedings of the National Academy of Sciences of the USA* 96(8), 4285–4288.
- Plummer, M., N. Best, K. Cowles, and K. Vines (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News* 6(1), 7–11.
- Rasmussen, C. E. (2000). The infinite Gaussian mixture model. In S. A. Solla, T. K. Leen, and K. Müller (Eds.), *Advances in Neural Information Processing Systems 12*, pp. 554–560. Cambridge, MA: MIT Press.
- Rossi, R. a., B. Gallagher, J. Neville, and K. Henderson (2013). Modeling dynamic behavior in large evolving graphs. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, New York, New York, USA, pp. 667–676. ACM Press.

- 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.
- Scripps, J., P.-N. Tan, and A.-H. Esfahanian (2007, oct). Exploration of Link Structure and Community-Based Node Roles in Network Analysis. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 649–654. IEEE.
- Tinati, R., S. Halford, L. Carr, and C. Pope (2014). Big Data: Methodological Challenges and Approaches for Sociological Analysis. *Sociology* 48(4), 663–681.
- Viegas, F. and M. Smith (2004). Newsgroup Crowds and AuthorLines: visualizing the activity of individuals in conversational cyberspaces. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*.
- 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.
- Wasserman, S. and K. Faust (1994). Social network analysis: Methods and applications. *Cambridge University Press* 1, 116.
- Welser, H. T., D. Cosley, G. Kossinets, A. Lin, F. Dokshin, G. Gay, and M. Smith (2011, feb). Finding social roles in Wikipedia. In *Proceedings of the 2011 iConference on - iConference '11*, New York, New York, USA, pp. 122–129. ACM Press.
- Welser, H. T., E. Gleave, D. Fisher, and M. A. Smith (2007). Visualizing the Signatures of Social Roles in Online Discussion Groups Finding Social Roles in Online Discussion. *Journal of Social Structure* 8(2), 1–32.
- Wernicke, S. and F. Rasche (2006, may). FANMOD: A tool for fast network motif detection. *Bioinformatics* 22(9), 1152–1153.
- 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.
- White, D. and K. Reitz (1983). Graph and semigroup homomorphisms on networks of relations. *Social Networks* 5, 193–234.
- White, H. C., S. A. Boorman, and R. L. Breiger (1976). Social structure from multiple networks. I. Blockmodels of roles and positions. *American journal of sociology* 81, 730–80.
- Whittaker, S., L. Terveen, W. Hill, and L. Cherny (1998). The dynamics of mass interaction. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pp. 257–264.

- Windham, M. P. and A. Cutler (1992). Information Ratios For Validating Mixture Analyses. *Journal of the American Statistical Association* 87(420), 1188–1192.
- Zhang, J., M. S. Ackerman, and L. Adamic (2007). Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*, New York, New York, USA, pp. 221. ACM Press.

A Conditional distributions for the dual-view mixture model

A.1 Chinese Restaurant Process

In this section we recall the derivation of a Chinese Restaurant Process. Such a process will be used as the prior over cluster assignments in the model. This prior will then be updated through the likelihoods of the observations through the different views.

Imagine that every user u belongs to one of K clusters. z_u is the cluster of user u and \mathbf{z} is a vector that indicates the cluster of every user. Let us assume that z_u is a random variable drawn from a multinomial distribution with probabilities $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$. Let us also assume that the vector $\boldsymbol{\pi}$ is a random variable drawn from a Dirichlet distribution with a symmetric concentration parameter $\boldsymbol{\alpha} = (\alpha/K, \dots, \alpha/K)$. We have:

$$\begin{aligned} z_u | \boldsymbol{\pi} &\sim \text{Multinomial}(\boldsymbol{\pi}) \\ \boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \end{aligned}$$

The marginal probability of the set of cluster assignments \mathbf{z} is:

$$\begin{aligned} p(\mathbf{z}) &= \int \prod_{u=1}^U p(z_u | \boldsymbol{\pi}) p(\boldsymbol{\pi} | \boldsymbol{\alpha}) d\boldsymbol{\pi} \\ &= \int \prod_{i=1}^K \pi_i^{n_i} \frac{1}{B(\boldsymbol{\alpha})} \prod_{j=1}^K \pi_j^{\alpha/K-1} d\boldsymbol{\pi} \\ &= \frac{1}{B(\boldsymbol{\alpha})} \int \prod_{i=1}^K \pi_i^{\alpha/K+n_i-1} d\boldsymbol{\pi} \end{aligned}$$

where n_i is the number of users in cluster i and B denotes the Beta function. Noticing that the integrated factor is a Dirichlet distribution with concentration parameter $\boldsymbol{\alpha} + \mathbf{n}$ but without its normalizing factor:

$$\begin{aligned} p(\mathbf{z}) &= \frac{B(\boldsymbol{\alpha} + \mathbf{n})}{B(\boldsymbol{\alpha})} \int \frac{1}{B(\boldsymbol{\alpha} + \mathbf{n})} \prod_{i=1}^K \pi_i^{\alpha/K+n_i-1} d\boldsymbol{\pi} \\ &= \frac{B(\boldsymbol{\alpha} + \mathbf{n})}{B(\boldsymbol{\alpha})} \end{aligned}$$

which expanding the definition of the Beta function becomes:

$$p(\mathbf{z}) = \frac{\prod_{i=1}^K \Gamma(\alpha/K + n_i)}{\Gamma(\sum_{i=1}^K \alpha/K + n_i)} \frac{\Gamma(\sum_{i=1}^K \alpha/K)}{\prod_{i=1}^K \Gamma(\alpha/K)} = \frac{\prod_{i=1}^K \Gamma(\alpha/K + n_i)}{\Gamma(\alpha + U)} \frac{\Gamma(\alpha)}{\prod_{i=1}^K \Gamma(\alpha/K)} \quad (\text{A.1})$$

where $U = \sum_{i=1}^K n_i$. Note that marginalizing out $\boldsymbol{\pi}$ we introduce dependencies between the individual clusters assignments under the form of the counts n_i . The conditional distribution of an individual assignment given the others is:

$$p(z_u = j | \mathbf{z}_{-u}) = \frac{p(\mathbf{z})}{p(\mathbf{z}_{-u})} \quad (\text{A.2})$$

To compute the denominator we assume cluster assignments are exchangeable, that is, the joint distribution $p(\mathbf{z})$ is the same regardless the order in which clusters are assigned. This allows us to assume that z_u is the last assignment, therefore obtaining $p(\mathbf{z}_{-u})$ by considering how Equation A.1 before z_u was assigned to cluster j .

$$p(\mathbf{z}_{-u}) = \frac{\Gamma(\alpha/K + n_j - 1) \prod_{i \neq j} \Gamma(\alpha/K + n_i)}{\Gamma(\alpha + U - 1)} \frac{\Gamma(\alpha)}{\prod_{i=1} \Gamma(\alpha/K)} \quad (\text{A.3})$$

And finally plugging Equations A.3 and A.1 into Equation A.2, and cancelling out the factors that do not depend on the cluster assignment z_u , and finally using the identity $a\Gamma(a) = \Gamma(a + 1)$ we get:

$$p(z_u = j | \mathbf{z}_{-u}) = \frac{\alpha/K + n_j - 1}{\alpha + U - 1} = \frac{\alpha/K + n_{-j}}{\alpha + U - 1}$$

where n_{-j} is the number of users in cluster j before the assignment of z_u .

The Chinese Restaurant Process is the consequence of considering $K \rightarrow \infty$. For clusters where $n_{-j} > 0$, we have:

$$p(z_u = j \text{ s.t } n_{-j} > 0 | \mathbf{z}_{-u}) = \frac{n_{-j}}{\alpha + U - 1}$$

and the probability of assigning z_u to any of the (infinite) empty clusters is:

$$p(z_u = j \text{ s.t } n_{-j} = 0 | \mathbf{z}_{-u}) = \lim_{K \rightarrow \infty} (K - p) \frac{\alpha/K}{\alpha + U - 1} = \frac{\alpha}{\alpha + U - 1}$$

where p is the number of non-empty components. It can be shown that the generative process composed of a Chinese Restaurant Process were every component j is associated to a probability distribution with parameters $\boldsymbol{\theta}_j$ is equivalent to a Dirichlet Process.

A.2 Conditionals for the feature view

In this appendix we provide the conditional distributions for the feature view to be plugged into the Gibbs sampler. Note that, except for $\beta_0^{(a)}$, conjugacy can be exploited in every case and therefore their derivations are straightforward and well known. The derivation for $\beta_0^{(a)}$ is left for another section:

A.2.1 Component parameters

Components means $p(\boldsymbol{\mu}_k^{(a)}|\cdot)$:

$$\begin{aligned} p(\boldsymbol{\mu}_k^{(a)}|\cdot) &\propto p\left(\boldsymbol{\mu}_k^{(a)}|\boldsymbol{\mu}_0^{(a)}, \left(\mathbf{R}_0^{(a)}\right)^{-1}\right) \prod_{u \in k} p\left(\mathbf{a}_u|\boldsymbol{\mu}_k^{(a)}, \mathbf{S}_k^{(a)}, \mathbf{z}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{\mu}_k^{(a)}|\boldsymbol{\mu}_0^{(a)}, \left(\mathbf{R}_0^{(a)}\right)^{-1}\right) \prod_{u \in k} \mathcal{N}\left(\mathbf{a}_u|\boldsymbol{\mu}_k^{(a)}, \mathbf{S}_k^{(a)}\right) \\ &= \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Lambda}') \end{aligned}$$

where:

$$\begin{aligned} \boldsymbol{\Lambda}' &= \mathbf{R}_0^{(a)} + n_k \mathbf{S}_k^{(a)} \\ \boldsymbol{\mu}' &= \boldsymbol{\Lambda}'^{-1} \left(\mathbf{R}_0^{(a)} \boldsymbol{\mu}_0^{(a)} + \mathbf{S}_k^{(a)} \sum_{u \in k} \mathbf{a}_u \right) \end{aligned}$$

Components precisions $p(\mathbf{S}_k^{(a)}|\cdot)$:

$$\begin{aligned} p(\mathbf{S}_k^{(a)}|\cdot) &\propto p\left(\mathbf{S}_k^{(a)}|\beta_0^{(a)}, \mathbf{W}_0^{(a)}\right) \prod_{u \in k} p\left(\mathbf{a}_u|\boldsymbol{\mu}_k^{(a)}, \mathbf{S}_k^{(a)}, \mathbf{z}\right) \\ &\propto \mathcal{W}\left(\mathbf{S}_k^{(a)}|\beta_0^{(a)}, (\beta_0^{(a)} \mathbf{W}_0^{(a)})^{-1}\right) \prod_{u \in k} \mathcal{N}\left(\mathbf{a}_u|\boldsymbol{\mu}_k^{(a)}, \mathbf{S}_k^{(a)}\right) \\ &= \mathcal{W}(\beta', \mathbf{W}') \end{aligned}$$

where:

$$\begin{aligned} \beta' &= \beta_0^{(a)} + n_k \\ \mathbf{W}' &= \left[\beta_0^{(a)} \mathbf{W}_0^{(a)} + \sum_{u \in k} (\mathbf{a}_u - \boldsymbol{\mu}_k^{(a)})(\mathbf{a}_u - \boldsymbol{\mu}_k^{(a)})^T \right]^{-1} \end{aligned}$$

A.2.2 Shared hyper-parameters

Shared base means $p(\boldsymbol{\mu}_0^{(a)}|\cdot)$:

$$\begin{aligned} p(\boldsymbol{\mu}_0^{(a)}|\cdot) &\propto p\left(\boldsymbol{\mu}_0^{(a)}|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a\right) \prod_{k=1}^K p\left(\boldsymbol{\mu}_k^{(a)}|\boldsymbol{\mu}_0^{(a)}, \mathbf{R}_0^{(a)}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{\mu}_0^{(a)}|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a\right) \prod_{k=1}^K \mathcal{N}\left(\boldsymbol{\mu}_k^{(a)}|\boldsymbol{\mu}_0^{(a)}, \left(\mathbf{R}_0^{(a)}\right)^{-1}\right) \\ &= \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Lambda}'^{-1}) \end{aligned}$$

where:

$$\begin{aligned}\Lambda' &= \Lambda_a + K\mathbf{R}_0^{(a)} \\ \boldsymbol{\mu}' &= \Lambda'^{-1} \left(\Lambda_a \boldsymbol{\mu}_a + K\mathbf{R}_0^{(a)} \overline{\boldsymbol{\mu}_k^{(a)}} \right)\end{aligned}$$

Shared base precisions $p(\mathbf{R}_0^{(a)}|\cdot)$:

$$\begin{aligned}p(\mathbf{R}_0^{(a)}|\cdot) &\propto p\left(\mathbf{R}_0^{(a)}|D, \Sigma_a^{-1}\right) \prod_{k=1}^K p\left(\boldsymbol{\mu}_k^{(a)}|\boldsymbol{\mu}_0^{(a)}, \mathbf{R}_0^{(a)}\right) \\ &\propto \mathcal{W}\left(\mathbf{R}_0^{(a)}|D, (D\Sigma_a)^{-1}\right) \prod_{k=1}^K \mathcal{N}\left(\boldsymbol{\mu}_k^{(a)}|\boldsymbol{\mu}_0^{(a)}, \left(\mathbf{R}_0^{(a)}\right)^{-1}\right) \\ &= \mathcal{W}(v', \Psi')\end{aligned}$$

where:

$$\begin{aligned}v' &= D + K \\ \Psi' &= \left[D\Sigma_a + \sum_k (\boldsymbol{\mu}_k^{(a)} - \boldsymbol{\mu}_0^{(a)})(\boldsymbol{\mu}_k^{(a)} - \boldsymbol{\mu}_0^{(a)})^T \right]^{-1}\end{aligned}$$

Shared base covariances $p(\mathbf{W}_0^{(a)}|\cdot)$:

$$\begin{aligned}p(\mathbf{W}_0^{(a)}|\cdot) &\propto p\left(\mathbf{W}_0^{(a)}|D, \frac{1}{D}\Sigma_a\right) \prod_{k=1}^K p\left(\mathbf{s}_k^{(a)}|\beta_0^{(a)}, \left(\mathbf{W}_0^{(a)}\right)^{-1}\right) \\ &\propto \mathcal{W}\left(\mathbf{W}_0^{(a)}|D, \frac{1}{D}\Sigma_a\right) \prod_{k=1}^K \mathcal{W}\left(\mathbf{s}_k^{(a)}|\beta_0^{(a)}, \left(\beta_0^{(a)}\mathbf{W}_0^{(a)}\right)^{-1}\right) \\ &= \mathcal{W}(v', \Psi')\end{aligned}$$

where:

$$\begin{aligned}v' &= D + K\beta_0^{(a)} \\ \Psi' &= \left[D\Sigma_a^{-1} + \beta_0^{(a)} \sum_{k=1}^K \mathbf{s}_k^{(a)} \right]^{-1}\end{aligned}$$

Shared base degrees of freedom $p(\beta_0^{(a)}|\cdot)$:

$$\begin{aligned} p(\beta_0^{(a)}|\cdot) &\propto p(\beta_0^{(a)}) \prod_{k=1}^K p(\mathbf{s}_k^{(a)}|\mathbf{w}_0^{(a)}, \beta_0^{(a)}) \\ &= p(\beta_0^{(a)}|1, \frac{1}{D}) \prod_{k=1}^K \mathcal{W}(\mathbf{s}_k^{(a)}|\mathbf{w}_0^{(a)}, \beta_0^{(a)}) \end{aligned}$$

where there is no conjugacy we can exploit. We may sample from this distribution with Adaptive Rejection Sampling.

A.3 Conditionals for the behavior view

In this appendix we provide the conditional distributions for the behavior view to be plugged into the Gibbs sampler. Except for β_{b_0} , conjugacy can be exploited in every case and therefore their derivations straightforward and well known. The derivation for β_{b_0} is left for another section:

A.3.1 Users parameters

Users latent coefficient $p(b_u|\cdot)$:

Let \mathbf{Z} be a $K \times U$ a binary matrix where $\mathbf{Z}_{k,u} = 1$ denotes whether user u is assigned to cluster k . Let $\mathbf{I}_{[T]}$ and $\mathbf{I}_{[U]}$ identity matrices of sizes T and U , respectively. Let $\boldsymbol{\mu}^{(f)} = (\mu_1^{(f)}, \dots, \mu_K^{(f)})$ and $\mathbf{s}^{(f)} = (s_1^{(f)}, \dots, s_K^{(f)})$ Then:

$$\begin{aligned} p(\mathbf{b}|\cdot) &\propto p(\mathbf{b}|\boldsymbol{\mu}^{(f)}, \mathbf{s}^{(f)}, \mathbf{Z})p(\mathbf{y}|\mathbf{P}, \mathbf{b}) \\ &\propto \mathcal{N}(\mathbf{b}|\mathbf{Z}^T \boldsymbol{\mu}^{(f)}, \mathbf{Z}^T \mathbf{s}^{(f)} \mathbf{I}_{[U]}) \mathcal{N}(\mathbf{y}|\mathbf{P}^T \mathbf{b}, \sigma_y \mathbf{I}_{[T]}) \\ &= \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Lambda}'^{-1}) \end{aligned}$$

where:

$$\begin{aligned} \boldsymbol{\Lambda}' &= \mathbf{Z}^T \mathbf{s}^{(f)} \mathbf{I}_{[U]} + \mathbf{P} \sigma_y^{-2} \mathbf{I}_{[T]} \mathbf{P}^T \\ \boldsymbol{\mu}' &= \boldsymbol{\Lambda}'^{-1} (\mathbf{Z}^T \mathbf{s}^{(f)} \mathbf{Z}^T \boldsymbol{\mu}^{(f)} + \mathbf{P} \sigma_y^{-2} \mathbf{I}_{[T]} \mathbf{y}) \end{aligned}$$

A.3.2 Component parameters

Components means $p(\mu_k^{(f)}|\cdot)$:

$$\begin{aligned} p(\mu_k^{(f)}|\cdot) &\propto p\left(\mu_k^{(f)}|\mu_0^{(f)}, \left(r_0^{(f)}\right)^{-1}\right) \prod_{u \in k} p(b_u|\mu_k^{(f)}, s_k^{(f)}, \mathbf{z}) \\ &\propto \mathcal{N}\left(\mu_k^{(f)}|\mu_0^{(f)}, \left(r_0^{(f)}\right)^{-1}\right) \prod_{u \in k} \mathcal{N}(b_u|\mu_k^{(f)}, s_k^{(f)}) \\ &= \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Lambda}'^{-1}) \end{aligned}$$

where:

$$\begin{aligned} \boldsymbol{\Lambda}' &= r_0^{(f)} + n_k s_k^{(f)} \\ \boldsymbol{\mu}' &= \boldsymbol{\Lambda}'^{-1} \left(r_0^{(f)} \mu_0^{(f)} + s_k^{(f)} \sum_{u \in k} b_u \right) \end{aligned}$$

Components precisions $p(s_k^{(f)}|\cdot)$:

$$\begin{aligned} p(s_k^{(f)}|\cdot) &\propto p(s_k^{(f)}|\beta_0^{(f)}, w_0^{(f)}) \prod_{u \in k} p(b_u|\mu_k^{(f)}, s_k^{(f)}, \mathbf{z}) \\ &\propto \mathcal{G}\left(s_k^{(f)}|\beta_0^{(f)}, \left(\beta_0^{(f)} w_0^{(f)}\right)^{-1}\right) \prod_{u \in k} \mathcal{N}(b_u|\mu_k^{(f)}, s_k^{(f)}) \\ &= \mathcal{G}(v', \psi') \end{aligned}$$

where:

$$\begin{aligned} v' &= \beta_0^{(f)} + n_k \\ \psi' &= \left[\beta_0^{(f)} w_0^{(f)} + \sum_{u \in k} \left(b_u - \mu_k^{(f)} \right)^2 \right]^{-1} \end{aligned}$$

A.3.3 Shared hyper-parameters

Shared base mean $p(\mu_0^{(f)}|\cdot)$:

$$\begin{aligned} p(\mu_0^{(f)}|\cdot) &\propto p(\mu_0^{(f)}|\mu_{\hat{b}}, \sigma_{\hat{b}}) \prod_{k=1}^K p(\mu_k^{(f)}|\mu_0^{(f)}, r_0^{(f)}) \\ &\propto \mathcal{N}(\mu_0^{(f)}|\mu_{\hat{b}}, \sigma_{\hat{b}}) \prod_{k=1}^K \mathcal{N}\left(\mu_k^{(f)}|\mu_0^{(f)}, \left(r_0^{(f)}\right)^{-1}\right) \\ &= \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\sigma}'^{-2}) \end{aligned}$$

where:

$$\begin{aligned}\sigma'^{-2} &= \sigma_{\hat{b}}^{-2} + Kr_0^{(f)} \\ \mu' &= \sigma_{\hat{b}}^{2'}(\sigma_{\hat{b}}^{-2}\mu_{\hat{b}} + Kr_0^{(f)}\overline{\mu_k^{(f)}})\end{aligned}$$

Shared base precision $p(r_0^{(f)}|\cdot)$

$$\begin{aligned}p(r_0^{(f)}|\cdot) &\propto p(r_0^{(f)}|1, \sigma_{\hat{b}}^{-2}) \prod_{k=1}^K p(\mu_k^{(f)}|\mu_0^{(f)}, r_0^{(f)}) \\ &\propto \mathcal{G}(r_0^{(f)}|1, \sigma_{\hat{b}}^{-2}) \prod_{k=1}^K \mathcal{N}\left(\mu_k^{(f)}|\mu_0^{(f)}, (r_0^{(f)})^{-1}\right) \\ &= \mathcal{G}(v', \psi')\end{aligned}$$

where:

$$\begin{aligned}v' &= 1 + K \\ \psi' &= \left[\sigma_{\hat{b}}^{-2} + \sum_{k=1}^K (\mu_k^{(f)} - \mu_0^{(f)})^2 \right]^{-1}\end{aligned}$$

Shared base variance $p(w_0^{(f)}|\cdot)$:

$$\begin{aligned}p(w_0^{(f)}|\cdot) &\propto p(w_0^{(f)}|1, \sigma_{\hat{b}}) \prod_{r=1}^K p(s_k^{(f)}|\beta_0^{(f)}, w_0^{(f)}) \\ &\propto \mathcal{G}(w_0^{(f)}|1, \sigma_{\hat{b}}) \prod_{k=1}^K \mathcal{G}\left(s_k^{(f)}|\beta_0^{(f)}, (\beta w_0^{(f)})^{-1}\right) \\ &= \mathcal{G}(v', \psi')\end{aligned}$$

$$v' = 1 + K\beta_0^{(f)}$$

$$\psi' = \left[\sigma_{\hat{b}}^{-2} + \beta_0^{(f)} \sum_{k=1}^K s_k^{(f)} \right]^{-1}$$

Shared base degrees of freedom $p(\beta_0^{(f)}|\cdot)$:

$$\begin{aligned}p(\beta_0^{(f)}|\cdot) &\propto p(\beta_0^{(f)}) \prod_{r=1}^K p(s_k^{(f)}|w_0^{(f)}, \beta_0^{(f)}) \\ &= p(\beta_0^{(f)}|1, 1) \prod_{r=1}^K \mathcal{G}\left(s_k^{(f)}|\beta_0^{(f)}, (\beta_0^{(f)}w_0^{(f)})^{-1}\right)\end{aligned}$$

where there is no conjugacy we can exploit. We will sample from this distribution with Adaptive Rejection Sampling.

A.3.4 Regression noise

Let the precision s_y be the inverse of the variance σ_y^2 . Then:

$$\begin{aligned}
p(s_y|\cdot) &\propto p(s_y|1, \sigma_0^{-2}) \prod_{t=1}^T p(y_t|\mathbf{p}^T \mathbf{b}, s_y) \\
&\propto \mathcal{G}(s_y|1, \sigma_0^{-2}) \prod_{t=1}^T \mathcal{N}(y_t|\mathbf{p}^T \mathbf{b}, s_y) \\
&= \mathcal{G}(v', \psi') \\
v' &= 1 + T \\
\psi' &= \left[\sigma_0^2 + \sum_{t=1}^T (y_t - \mathbf{p}^T \mathbf{b})^2 \right]^{-1}
\end{aligned}$$

A.4 Sampling $\beta_0^{(\mathbf{a})}$

For the feature view, if:

$$\frac{1}{\beta - D + 1} \sim \mathcal{G}\left(1, \frac{1}{D}\right)$$

we can get the prior distribution of β by variable transformation:

$$\begin{aligned}
p(\beta) &= \mathcal{G}\left(\frac{1}{\beta - D + 1}\right) \left| \frac{\partial}{\partial \beta} \frac{1}{\beta - D + 1} \right| \\
&\propto \left(\frac{1}{\beta - D + 1}\right)^{-1/2} \exp\left(-\frac{D}{2(\beta - D + 1)}\right) \frac{1}{(\beta - D + 1)^2} \\
&\propto \left(\frac{1}{\beta - D + 1}\right)^{3/2} \exp\left(-\frac{D}{2(\beta - D + 1)}\right)
\end{aligned}$$

Then:

$$p(\beta) \propto (\beta - D + 1)^{-3/2} \exp\left(-\frac{D}{2(\beta - D + 1)}\right)$$

The Wishart likelihood is:

$$\begin{aligned}
\mathcal{W}(\mathbf{S}_k|\beta, (\beta \mathbf{W})^{-1}) &= \frac{(|\mathbf{W}|(\beta/2)^D)^{\beta/2}}{\Gamma_D(\beta/2)} |\mathbf{S}_k|^{(\beta-D-1)/2} \exp\left(-\frac{\beta}{2} \text{Tr}(\mathbf{S}_k \mathbf{W})\right) \\
&= \frac{(|\mathbf{W}|(\beta/2)^D)^{\beta/2}}{\prod_{d=1}^D \Gamma(\frac{\beta+d-D}{2})} |\mathbf{S}_k|^{(\beta-D-1)/2} \exp\left(-\frac{\beta}{2} \text{Tr}(\mathbf{S}_k \mathbf{W})\right)
\end{aligned}$$

We multiply both equations, the Wishart likelihood (its K factors) and the prior, to get the posterior:

$$p(\beta|\cdot) = \left(\prod_{d=0}^D \Gamma\left(\frac{\beta}{2} + \frac{d-D}{2}\right) \right)^{-K} \exp\left(-\frac{D}{2(\beta-D+1)}\right) (\beta-D+1)^{-3/2} \\ \times \left(\frac{\beta}{2}\right)^{\frac{KD\beta}{2}} \prod_{k=1}^K (|\mathbf{S}_k||\mathbf{W}|)^{\beta/2} \exp\left(-\frac{\beta}{2}\text{Tr}(\mathbf{S}_k\mathbf{W})\right)$$

Then if $y = \ln \beta$:

$$p(y|\cdot) = e^y \left(\prod_{d=0}^D \Gamma\left(\frac{e^y}{2} + \frac{d-D}{2}\right) \right)^{-K} \exp\left(-\frac{D}{2(e^y-D+1)}\right) (e^y-D+1)^{-3/2} \\ \times \left(\frac{e^y}{2}\right)^{\frac{KD e^y}{2}} \prod_{k=1}^K (|\mathbf{S}_k||\mathbf{W}|)^{e^y/2} \exp\left(-\frac{e^y}{2}\text{Tr}(\mathbf{S}_k\mathbf{W})\right)$$

and its logarithm is:

$$\ln p(y|\cdot) = y - K \sum_{d=0}^D \ln \Gamma\left(\frac{e^y}{2} + \frac{d-D}{2}\right) - \frac{D}{2(e^y-D+1)} - \frac{3}{2} \ln(e^y-D+1) \\ + \frac{KD e^y}{2} (y - \ln 2) + \frac{e^y}{2} \sum_{k=1}^K (\ln(|\mathbf{S}_k||\mathbf{W}|) - \text{Tr}(\mathbf{S}_k\mathbf{W}))$$

which is a concave function and therefore we can use Adaptive Rejection Sampling (ARS). ARS sampling works with the derivative of the log function:

$$\frac{\partial}{\partial y} \ln p(y|\cdot) = 1 - K \frac{e^y}{2} \sum_{d=1}^D \Psi\left(\frac{e^y}{2} + \frac{d-D}{2}\right) + \frac{D e^y}{2(e^y-D+1)^2} - \frac{3}{2} \frac{e^y}{e^y-D+1} \\ + \frac{KD e^y}{2} (y - \ln 2) + \frac{KD e^y}{2} + \frac{e^y}{2} \sum_{k=1}^K (\ln(|\mathbf{S}_k||\mathbf{W}|) - \text{Tr}(\mathbf{S}_k\mathbf{W}))$$

where $\Psi(x)$ is the digamma function.

A.5 Sampling $\beta_0^{(f)}$

For the behavior view, if

$$\frac{1}{\beta} \sim \mathcal{G}(1, 1)$$

the posterior of β is:

$$p(\beta|\cdot) = \Gamma\left(\frac{\beta}{2}\right)^{-K} \exp\left(\frac{-1}{2\beta}\right) \left(\frac{\beta}{2}\right)^{(K\beta-3)/2} \prod_{k=1}^K (s_k w)^{\beta/2} \exp\left(-\frac{\beta s_k w}{2}\right)$$

Then if $y = \ln \beta$:

$$p(y|\cdot) = e^y \Gamma\left(\frac{e^y}{2}\right)^{-K} \exp\left(\frac{-1}{2e^y}\right) \left(\frac{e^y}{2}\right)^{(Ke^y-3)/2} \prod_{k=1}^K (s_k w)^{e^y/2} \exp\left(-\frac{e^y s_k w}{2}\right)$$

and its logarithm:

$$\ln p(y|\cdot) = y - K \ln \Gamma\left(\frac{e^y}{2}\right) + \left(\frac{-1}{2e^y}\right) + \frac{Ke^y - 3}{2} (y - \ln 2) + \frac{e^y}{2} \sum_{k=1}^K (\ln(s_k w) - s_k w)$$

which is a concave function and therefore we can use Adaptive Rejection Sampling. The derivative is:

$$\frac{\partial}{\partial y} \ln p(y|\cdot) = 1 - K \Psi\left(\frac{e^y}{2}\right) \frac{e^y}{2} + \left(\frac{1}{2e^y}\right) + \frac{Ke^y}{2} (y - \ln 2) + \frac{Ke^y - 3}{2} + \frac{e^y}{2} \sum_{k=1}^K (\ln(s_k w) - s_k w)$$

where $\Psi(x)$ is the digamma function.

A.6 Sampling α

Since the inverse of the concentration parameter α is given a Gamma prior

$$\frac{1}{\alpha} \sim \mathcal{G}(1, 1)$$

we can get the prior over α by variable transformation:

$$p(\alpha) \propto \alpha^{-3/2} \exp(-1/(2\alpha))$$

Multiplying the prior of α by its likelihood we get the posterior:

$$\begin{aligned} p(\alpha|\cdot) &\propto \alpha^{-3/2} \exp(-1/(2\alpha)) \times \frac{\Gamma(\alpha)}{\Gamma(\alpha + U)} \prod_{j=1}^K \frac{\Gamma(n_j + \alpha/K)}{\alpha/K} \\ &\propto \alpha^{-3/2} \exp(-1/(2\alpha)) \frac{\Gamma(\alpha)}{\Gamma(\alpha + U)} \alpha^K \\ &\propto \alpha^{K-3/2} \exp(-1/(2\alpha)) \frac{\Gamma(\alpha)}{\Gamma(\alpha + U)} \end{aligned}$$

Then if $y = \ln \alpha$:

$$p(y|\cdot) = e^{y(K-3/2)} \exp(-1/(2e^y)) \frac{\Gamma(e^y)}{\Gamma(e^y + U)}$$

and its logarithm is:

$$\ln p(y|\cdot) = y(K - 3/2) - 1/(2e^y) + \ln \Gamma(e^y) - \ln \Gamma(e^y + U)$$

which is a concave function and therefore we can use Adaptive Rejection Sampling. The derivative is:

$$\frac{\partial}{\partial y} \ln p(y|\cdot) = (K - 3/2) + 1/(2e^y) + e^y \Psi(e^y) - e^y \Psi(e^y + U)$$