User Preference Dynamics on Evolving Social Networks - Learning, Modeling and Prediction

Fabíola Souza Fernandes Pereira

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2018

Fabíola Souza Fernandes Pereira

User Preference Dynamics on Evolving Social Networks - Learning, Modeling and Prediction

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientadora: Gina Maira Barbosa de Oliveira
Coorientador: João Gama

Uberlândia
2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

P436 Pereira, Fabíola Souza Fernandes, 1987-
2018 User preference dynamics on evolving social networks [recurso eletrônico] : learning, modeling and prediction / Fabíola Souza Fernandes Pereira. - 2018.

Orientadora: Gina Maira Barbosa de Oliveira.
Coorientador: João Gama.
Tese (Doutorado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Ciência da Computação.
Disponível em: http://dx.doi.org/10.14393/ufu.te.2018.804
Inclui bibliografia.
Inclui ilustrações.

1. Computação. 2. Redes sociais - Análise. 3. Recuperação da informação. I. Oliveira, Gina Maira Barbosa de (Orient.). II. Gama, João (Coorient.). III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Maria Salete de Freitas Pinheiro - CRB6/1262

To my parents, Iron and Solimar.
To my beloved Diogo.

Acknowledgments

Firstly, I would like to express my deep gratitude to Prof. João Gama and Prof. Gina, my research supervisors. Prof. João Gama is one of the few fascinating people who we have the pleasure to meet in life. An exceptional professional and a humble human being. He offered me great opportunities. Encouraged me to submit works that I would never have imagined I could achieve. Prof. Gina is responsible for one of the most important achievements of my life, which was the finishing of this work. She enlightened every step of this journey with her personal and professional advices. I thank both for granting me the opportunity to grow as a researcher.

My gratitude also goes to my parents, Iron and Solimar, for their unconditional love and support. I also thank my beloved Diogo, whose embrace and caring words made everything seem easier. A special thanks to my sister Flávia and my little niece Bárbara. You all are my huge treasure. This work is yours.

I thank my friends and labmates who supported me in so many different ways. To Guilherme, Crícia and Klérisson for the moments in the LSI, talking, discussing and laughing. To Cláudio and Jean for the insightful talks and collaboration. To Fernanda, Myllene and Franciny for being my sisters and partners forever. You made the happy moments much more sweet. My sincere thanks also to my colleagues of LIAAD, especially I am grateful to Shazia for her friendship and collaboration in this work.

I acknowledge CAPES for the financial support.

Last but not least, I write, with great emotion, my principal thanks – to my eternal mentor Sandra de Amo, for everything she represented in my academic career. Unfortunately, fate did not allow us to continue working together. I miss you so much and there was not a day on this walk that I did not think about you. I hope, Sandra, to have succeeded in giving continuity to yours teachings that have been concretized in this thesis. I dedicate to you all my effort and conquest. Without you, I found myself in the middle of a doctorate. Without you, I lost my reference, my lawyer, my mentor. I miss you so much in UFU and in my career. I hope I have achieved OUR goal. My eternal gratitude. Receive the affection and deep respect of your last UFU doctoral student... Fabíola.

Resumo

Modelar as preferências e necessidades dos usuários é uma das tarefas de personalização mais importantes no domínio de recuperação de informações. Tais preferências são muito dinâmicas, já que os usuários tendem a explorar uma ampla variedade de itens e a modificar seus gostos de acordo com o tempo. Além disso, a todo momento, os usuários se deparam com opiniões dos outros e sofrem influência social. Em nossa pesquisa, investigamos a interação entre preferências do usuário e redes sociais ao longo do tempo. Definimos o que são dinâmicas de preferência do usuário e propusemos um modelo de preferência temporal capaz de descrever como as preferências do usuário evoluem ao longo do tempo por meio de alterações em seus perfis. Como solução para o problema, primeiro investigamos as redes temporais. Ao modelar uma amostra da rede do Twitter como uma rede social temporal, percebemos como os nós evoluem em função das métricas de centralidade e quão diferente é a evolução ao considerar redes estáticas versus temporais. Em seguida, exploramos a ideia de detecção de eventos de nó com base na centralidade de redes em evolução. O objetivo foi detectar em que pontos no tempo um nó altera significativamente seu comportamento. A proposta resultante foi um modelo de mineração de eventos de nó com duas estratégias diferentes para detectar pontos de mudança. Finalmente, juntamos nossas descobertas e propostas até então e realizamos uma avaliação experimental. A descoberta é que existe uma forte correlação entre eventos de mudança de preferência e eventos de nó baseados em centralidade, especialmente quando se considera redes temporais. Ao final, construímos uma solução completa para o problema de previsão de mudança de preferência, levando em consideração o uso da detecção de eventos de nós em redes em constante evolução, nas quais o tempo em que as arestas estão ativas é um elemento explícito da representação. Nosso modelo é capaz de prever mudanças nas preferências do usuário com níveis competitivos de precisão.

Palavras-chave: Redes temporais. Análise de redes sociais. Preferências do usuário. Redes sociais evolucionárias..

User Preference Dynamics on Evolving Social Networks - Learning, Modeling and Prediction

Fabíola Souza Fernandes Pereira

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA – UFU
FACULDADE DE COMPUTAÇÃO – FACOM
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO – PPGCO

The undersigned hereby certify they have read and recommend to the PPGCO for acceptance the thesis entitled **"User Preference Dynamics on Evolving Social Networks"** submitted by **"Fabíola Souza Fernandes Pereira"** as part of the requirements for obtaining the **Doctoral degree in Computer Science**.

Uberlândia, July 18, 2018.

Supervisor: _____
Prof. Dr. Gina Maira Barbosa de Oliveira
Federal University of Uberlândia

Cosupervisor: _____
Prof. Dr. João Gama
INESC TEC, University of Porto

Examining Committee Members:

_____
Prof. Dr. André C. P. L. F. de Carvalho
University of São Paulo

_____
Prof. Dr. Bruno Augusto Nassif Travençolo
Federal University of Uberlândia

_____
Prof. Dr. Elaine Ribeiro de Faria
Federal University of Uberlândia

_____
Prof. Dr. Fabrício Benevenuto
Federal University of Minas Gerais

Abstract

Modeling users' preferences and needs is one of the most important personalization tasks in information retrieval domain. User preferences are fairly dynamic, since users tend to exploit a wide range of items and modify their tastes accordingly over time. Moreover, all the time users are facing with others' opinions and suffering social influence. In our research, we investigate the *interplay of User Preferences and Social Networks over time*. We define what are user preference dynamics and propose a temporal preference model able to describe how user preferences evolve over time through changes on user profiles. As problem solution, we first investigate temporal networks. By modeling a sample of Twitter network as a temporal social network we perceive how nodes evolve in function of centrality metrics and how different is the evolution when considering static *vs.* temporal networks. Then, we explore the idea of centrality-based node event detection in evolving networks. The goal is to detect at what points in time a node change its behavior significantly. Our proposal is a node event mining model with two different strategies for detecting change points. Finally, we join our findings and proposals so far and perform an experimental evaluation using two datasets from different domains focused on our main goal: the interplay between user preferences and social networks over time. The discovery is that there is a strong correlation between preference change events and centrality-based node events, specially when considering temporal networks. Moreover, closeness centrality is more suitable when correlating preference changes and node events than betweenness. In the end, we build a complete solution for the preference change prediction problem, taking into account the use of node events detection in continuously evolving networks where the time when edges are active are an explicit element of the representation. Our prediction model is able to forecast changes on user preferences with competitive levels of accuracy.

Keywords: Temporal networks. Social network analysis. User preferences. Evolutionary network analysis.

List of Figures

List of Tables

List of Acronyms

BTG Better-than Graph

LDA Latent Dirichlet Allocation

MWA Moving Window Average

OSN Online Social Network

PH Page-Hinkley

TC Transitive Closure

TIMJ This Is My Jam

UPD User Preference Dynamics

WMWA Weighted Moving Window Average

Contents

I hereby certify that I have obtained all legal permissions from the owner(s) of each third-party copyrighted matter included in my thesis, and that their permissions allow availability such as being deposited in public digital libraries.

Fabiola Souza Fernandes Pereira

Chapter 1

Introduction

What drives people's preferences dynamics? Modeling users' preferences and needs is one of the most important personalization tasks in recommendation and information retrieval domains. User preferences are fairly dynamic, since users tend to exploit a wide range of items and modify their tastes accordingly over time. Moreover, all the time users are facing with others' opinions and suffering social influence. Online social networks, such as Facebook, Twitter, and music social networks, facilitate the building of social relations among people who share similar interests. Users can stay connected with each other and be informed of new trends, consumption preferences and opinions of social friends. According to Cartwright and Harary's theory (CARTWRIGHT; HARRARY, 1956), individual's opinion regarding another person, idea or product is influenced by those with whom he/she shares positive social ties.

The development of formalisms for preference specification and reasoning are essential tasks in literature since they can be used for sorting and selecting the objects that most fulfill user wishes. There are mining techniques for the automatic discovery of preferences and user profile building (AMO et al., 2015), and there also exists research in the development of powerful mechanisms for preference reasoning (WILSON, 2004). We are interested in user preference dynamics, i.e., the observation of how a user forms and evolves her preferences over time. A user preference is a specific type of opinion derived from comparative perception between two objects (HANSSON, 1995). For instance, when a user expresses "I don't like to read about politics. Sports are much better", we clearly identify her preference to sports news over politics.

Just as preferences change over time, new links and nodes are continuously created on a wide variety of social networks as new users join the network, and new friendships are created. This leads to a number of important analysis such as event and anomaly detection (AGGARWAL; SUBBIAN, 2014) in evolutionary networks analysis field. Indeed, key changes in the network structure often reflect individuals reaction to external events and trends (ARIAS et al., 2014). One can imagine that as the network evolves users evolve their social influence as well, which can directly result in changes to individual

preferences. Recent research has made considerable advances towards the understanding of fundamental structural properties (BOCCALETTI et al., 2006), community structure (OLIVEIRA et al., 2014), information diffusion (GUILLE et al., 2013), and social influence (SUN; TANG, 2011) on online social networks. However, the impact of the online social networks on user preferences remains elusive. For example, little is known about whether and to what extent node centralities on social networks are related to user tastes and behavior changes (ALTHOFF et al., 2017).

User preferences are largely applied in recommender systems (AGGARWAL, 2016). Content-based methods use a user's preference to find similar content, while collaborative methods use a user's preference to identify similar users and recommend popular content in the identified neighborhood. Efforts in modeling temporal recommendations have exploited the aspect of user choices by designing recommendation systems which systematically emphasize recency with good results (KOREN, 2009). The critical shortcoming of this formulation is that such a system merely reacts to preference changes rather than trying to predict them. The recommendations community lacks models which can predict changing preferences of users.

Given the above scenario, this work aims at investigating the interplay between user preferences and social networks over time for systems personalization. We hypothesize that the evolution of a user's preference is related to the evolution of her social network structure, specially when it comes to the detection of changes.

1.1 Motivation

After all, what exactly means to observe how user preferences evolve over time? The following example illustrates the preference dynamics problem we investigate in this work.

Let us consider a context concerning news that users like to read in everyday life. Suppose that analyzing the preferences of a given user A, we detect that on Aug 21^{st}, A prefers to read about *politics* and *economy* than other news categories such as *sports* or *health*. Then, in a second moment, A's preferences remain stable, just appearing a preference of *politics* over *economy* news. However, in a third moment, on Aug 30^{th}, we observe that A's preferences have changed and now, *economy* is preferred over *politics*. This situation is illustrated in the upper part of Figure 1, where preferences are represented by better-than graphs (a directed edge (u, v) indicates that u is preferred over v). In the lower part of Figure 1, snapshots of A's social network are represented. We notice that the network is also evolving with nodes appearing, disappearing, associating and disassociating with each other as time flies. In the network, nodes are Twitter users and a directed edge (x, y) means that y retweeted[1] x, i.e., the information flow. We conjecture

[1] Retweet is to share some content originally posted by someone else in Twitter.

that many aspects of A's social network can influence on A's preferences evolution. For instance:

- ❏ Around Aug 27^{th}, A was being influenced by users who also like *politics*.
- ❏ From 27^{th} to Aug 30^{th}, a new connection with an influential personality in *economy* may have appeared and influenced A.
- ❏ A is always in contact with people who like *sports*.

It is an essential point to detect and predict A's preferences evolution and changes over time. We show in this thesis that the temporal-topological social network structure of a given user is strongly correlated with her preference dynamics. According to our findings, by just observing A's social network evolution, we could increase the assertiveness of a news recommendation system for example, when recommending economy instead of politics news to A from Aug 30^{th}.

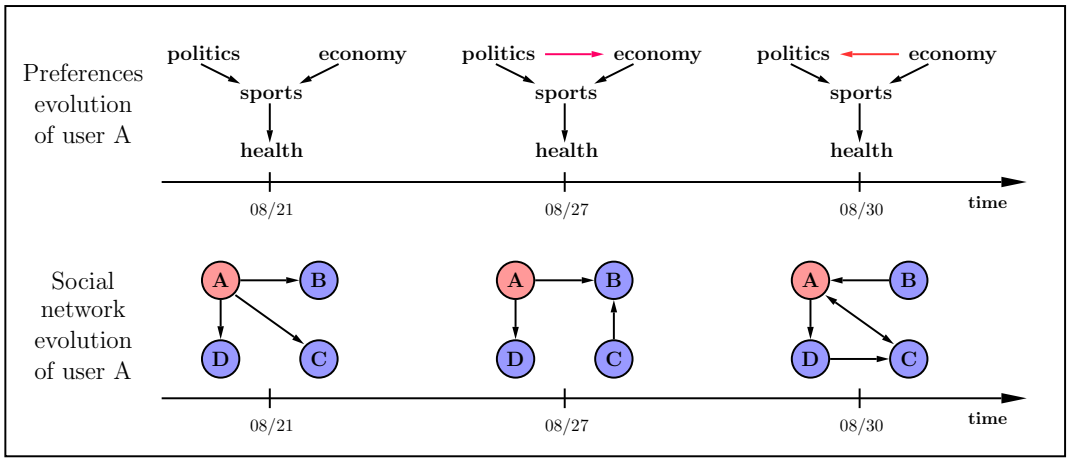

Figure 1 – Evolving perspective of A's temporal preferences (top) and A's temporal social network (bottom). Preferences are represented by better-than graphs where an edge (u, v) means that u is preferred over v. In the network, nodes are Twitter users and an edge (a, b) means that b retweeted a. On 08/30 there was a significant preference change. At the same time, A significantly changed her structural position (node centrality) on the network.

1.2 Goals and Research Challenges

In the previous sections we discussed about user preferences evolving over time, the importance of predicting evolution patterns like changes and novelty, the social influence that we suffer continuously and consequently, the good source of knowledge that social networks represent. In this context, the main purpose of this thesis is to:

investigate the interplay between User Preferences and Social Networks over time for systems personalization, through the online prediction of preference change on a continuously evolving network.

We decompose the main research goal into five specific goals, related with modeling, learning and predicting user preference changes. The first one focus on modeling user preference dynamics (UPD). Next, three specific goals compose the learning phase. We seek for insights on how to deal with the UPD problem on evolving networks. Our last specific goal is about predicting user preference changes. Such specific goals can be described as follows:

1. introduce and formalize the problem of UPD focusing on user preference changes;
2. investigate the suitable of temporal networks as foundation for the UPD problem solving;
3. propose, implement and evaluate an algorithm for centrality-based event detection in evolving networks;
4. investigate the correlation between user preference changes and centrality-based events in evolving networks and
5. propose, implement and evaluate a preference change prediction approach.

1.3 Hypothesis

We argue that social networks are good source of knowledge to investigate how user preferences evolve over time. In order to reach our goals and with this argument in mind, we elaborate three hypothesis that guided the topics explored in this thesis, especially our experimental settings.

- ❑ H1: There is a correlation between user preference changes and centrality-based node events in evolving social networks.
- ❑ H2: Temporal networks are more suitable than static networks to analyze user preference dynamics.
- ❑ H3: It is possible to predict, with high accuracy, user preference changes using centrality-based event detection over a continuously evolving network.

1.4 Contributions

In this thesis we link two research lines: dynamic network analysis and user preferences, being both of utmost importance for the field of discovery science. We make specific contributions to the fields of evolutionary network analysis, preference learning, social networks analysis and social computing. The contributions of this thesis are briefly described as follows.

- ❏ Introduction and formalization of the user preferences dynamics problem focusing on preference changes.

- ❏ An innovative analysis of users temporal centralities evolution over Twitter follower/followee network modeled as a temporal social network.

- ❏ A new approach for centrality-based events detection in evolving networks, consisting mainly of a node event mining model.

- ❏ A correlation of user preference changes and centrality-based events in temporal networks.

- ❏ A preference change prediction model. The model processes evolving network streams to detect change points based on node centrality values. We employ memory less and window based aggregating mechanisms for tracking the vacillation of these values.

1.5 Thesis Organization

We organize this thesis mainly focusing on the delimitation of the problem and positioning in relation to the state of art. The chapters order also reflects the order that we investigate respective issues. Lastly, our contributions are in Chapters 4, 5, 6, 7 and 8.

Chapter 2 [Background]. Presents the background we used to tackle our problem, namely user preferences, temporal networks and evolutionary networks analysis.

Chapter 3 [Related Work]. We organize the state of art according to three main topics: temporal dynamics of user preferences, event detection in evolving networks and the interplay between user preferences and social networks. In literature these topics are combined two by two, which brings the originality of our proposal by combining all of them.

Chapter 4 [User Preference Dynamics]. Here we formally define the problem of User Preference Dynamics (UPD). First, we propose a temporal preference model able to represent and reasoning with preferences over time. Then, we describe how to detect events of changes on temporal preferences.

Chapter 5 [Temporal Networks for UPD Analysis]. We propose the use of temporal social networks for UPD analysis. A rich experimental analysis in Twitter follower/followee dataset is presented concerning node centralities evolution to ratify our proposal.

Chapter 6 [Event Detection in Evolving Networks]. In this chapter we propose the notion of centrality-based event detection in evolving networks. We present the experimental evaluation over a Twitter interaction dataset to validate the ideas proposed.

Chapter 7 [Correlating Changes on User Preferences and Node Centrality in Evolving Networks]. In this chapter we use the insights gained in previous chapters and present the experimental results concerning the correlation between user preference changes and centrality-based events in evolving networks. The experiments were conducted over Twitter and the social music network This Is My Jam (TIMJ).

Chapter 8 [The Preference Change Prediction Model]. Finally, this chapter is focused on the preference change prediction model over evolving networks. We propose new strategies to detect node events, compatible with streaming networks, and then perform the preference change predictions also over Twitter.

Chapter 9 [Conclusion]. Concludes the thesis, summarizing main contributions, describing bibliographic production and pointing future directions of our research.

Chapter 2

Background

Preference learning refers to the problem of learning from observations which reveal, either explicitly or implicitly, information about the preferences of an individual. The acquisition of this kind of information can be supported by methods for preference mining. Generalizing beyond the training data given, the models learned are typically used for preference prediction, i.e., to predict the preferences of a new individual or the same individual in a new situation. We begin this chapter explaining the whole chain of this preference learning process, highlighting how it is applied into recommendation systems context.

Regarding networks evolving over time, we base our proposal on temporal networks which are network structures where the time when edges are active are an explicit element of the representation. Besides formally define a temporal network, we detail temporal measures, namely time-respecting paths and temporal centralities. These measures quantify the social influence of a node and need rethinking when considering the additional time dimension.

Last but not least, techniques of evolutionary network analysis also support our proposals. In this field the goal is to automatically learn temporal structural behaviors of nodes as time goes by and the edges (relationships) change in the network. We discuss about strategies for processing these continuously growing networks and essential properties that algorithms should satisfy.

Our goal in this chapter is to provide definitions of main concepts related to our proposal along with illustrative examples.

2.1 User Preferences

Given a pair of objects, predict which one is the most preferred. This is the basic goal of methods developed in user preferences field. In fact, a lot of work has been done concerning the topic of user preferences (AMO et al., 2015; WILSON, 2004). Methods for Preference Learning can be categorized following different criteria such as *Prefer-*

ence Elicitation (explicitly express preferences or mining implicit preferences), *Preference Specification* (user profiling), *Preference Semantics* (reasoning with preference models like pareto, conditional preference model) and *Application Domain* (document retrieval, query answering filtering, product rating, sentiment analysis, etc).

Elicitation of preferences consists basically in providing the user a way to inform his/her preferences on objects belonging to a dataset. The most efficient way to achieve it is by capturing implicit user's choices and applying preference mining algorithms. Concerning specification, in a qualitative approach, preferences are usually detailed by a compact set of preference rules from which a preference relation can be inferred. In (BöRZSöNYI et al., 2001) the authors model preferences as strict partial orders with Pareto semantics. Let us informally present these concepts through the following example.

Example 2.1.1 *John wants to personalize his timeline and has the following qualitative preferences about news: (1) I prefer news described by small texts. (2) Generally, I like news about politics. (3) During olympics games I am interested in sports news. These wishes are normally contradictory and a hard-constrained query over his preferences would result in an undesired empty result. Table 1 holds some sample news, their respective subjects and further features.*

Table 1 – News dataset.

Category	**Size**	**Timestamp**	**Subject**
sports	small	April	Neymar
politics	small	July	Impeachment
sports	large	August	RJ Olympics
health	large	May	Vitamin D
politics	medium	August	Corruption

According to Pareto preference semantics (BöRZSöNYI et al., 2001), two or more preferences are combined with all of them being equally important. Running a preference query over news dataset, just those news that are not worse than any other are returned. The result is depicted in Figure 2, where an edge from u to v in the better-than graph means that u is better than v according to Pareto semantics.

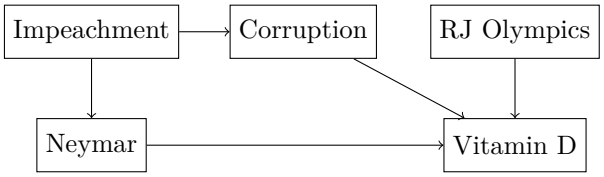

Figure 2 – News preference ordering.

Based on the three preferences expressed by John and on dataset domain, it is possible to mine that he would not like to read about Vitamin D and that Impeachment and RJ Olympics are the most preferred subjects.

2.2 Learning User Preferences in Recommender Systems

Methods for learning and predicting preferences in an automatic way are among the recent research topics in disciplines such as machine learning, knowledge discovery, and recommender systems. Approaches relevant to this area range from approximating of an as effective as possible question-answer process (preference elicitation) to collaborative filtering where a customer preferences are estimated from the preferences of other customers (FURNKRANZ; HULLERMEIER, 2010). In fact, problems of preference learning can be formalized within various settings, depending on the underlying type of preference model or the type of information provided as an input to the learning system. We explore the role of user preferences in recommender systems.

In a general way, to build an effective recommender system the following process is executed: first elicit patterns from feedback, which can be explicit (e.g. rating movies) or implicit (e.g. social data, visual perception, clicks, logs). The preference mining task consists in deriving a model from feedback able to infer a preference order between two given objects. This model is often referred to as prediction model. In some proposals, a preference mining task is not used, and there is just a user profiling module that seeks to represent preferences through feature vectors or tensors. In the end, given some items and a target user u, the goal is to predict a preference order or a ranking (a special case of total orders of a set of alternatives) of these items according to u's preferences. This process is depicted in Figure 3.

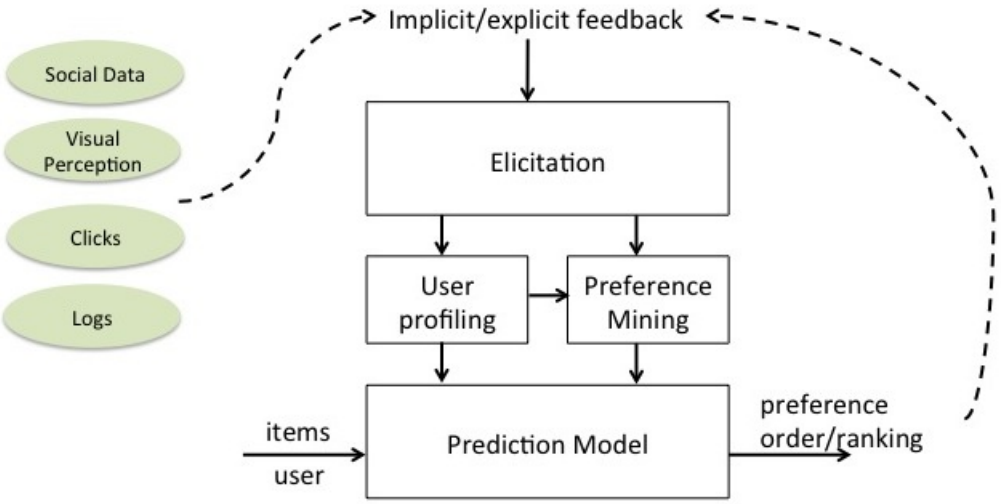

Figure 3 – Modelling and processing of a traditional recommendation schema.

2.3 Temporal Networks

In this thesis, we consider temporal networks, where the times *when* edges are active are an explicit element of the representation. A classical example of a temporal network application is on disease contagion through physical proximity (HOLME; SARAMAKI, 2012). Normally, the spreading of pathogenic organisms occurs through contact between two individuals and a temporal network is the best structure to represent this scenario. Social networks, our topic of interest, can also be represented as temporal networks, as they are increasingly ubiquitous and complex on their interactions (HOLME, 2014). Another examples of networks that can be represented as temporal networks are: face-to-face communications (CATTUTO et al., 2013), flights networks (WU et al., 2014) and phone calls (TABASSUM; GAMA, 2016).

There are many definitions in literature that formalize temporal networks (here, invariably also called temporal graphs) (WU et al., 2014; NICOSIA et al., 2013; KIM; ANDERSON, 2012; HOLME; SARAMAKI, 2012). Kim e Anderson (2012) defined the *time-ordered graphs* and Nicosia et al. (2013) call them *time-varying graphs*, but generally all definitions represent a set of time-edges among a set of nodes during an observation interval that takes into account their temporal ordering.

Definition 2.3.1 (Temporal network) *A temporal network $G = (V, E)$ is a set E of edges registered among a set of nodes V during an observation interval $[0, T]$. An edge between two nodes $u, v \in V$ is represented by a quadruplet $e = (u, v, t, \delta t)$, where $0 \leq t \leq T$ is the time at which the edge started and δt is its duration. The edges can also be called contacts.*

This definition is classical for representing flight graphs and phone calls networks, for example. But there are extensions to above definition. When contacts are instantaneous, $\delta t \to 0$, the temporal network is defined as a *contact sequence graph* (HOLME; SARAMAKI, 2012). These graphs are used to represent systems which the duration of the contact is less important (e-mails, sexual networks, likes in social networks). Another variation is to define temporal networks with edges that are not active over a set of times but rather over a set of intervals $e = (u, v, t_{init}, t_{end})$. These are the *interval graphs* (HOLME; SARAMAKI, 2012), good for modeling follow/unfollow relationships in Twitter network (PEREIRA et al., 2016a) or infrastructural systems like the Internet. In our proposal we use both *contact sequence* and *interval graphs*, varying according to the dataset and context that we apply. In fact, interval graphs can be transformed into contact sequence graphs and most of the network analysis techniques hold in both cases. More details will be discussed in Chapter 5.

Example 2.3.1 *Figure 4 illustrates two temporal networks. Let us consider the context of Twitter social network. In Figure 4(a) we have a contact sequence graph, representing*

mention interactions among users. The nodes are users and an edge (u, v, t) *indicates that* u *mentioned* v *in a tweet posted at* t[1]*. The times of the interactions are states next to the edges and the duration of the interactions are negligible. We can see that the users A and B interacted at times 3, 6 and 11, the users B and C interacted at 7 and 9 and so on.*

Now in the same Twitter context, we can consider the interval graph in Figure 4(b) where the edges represent follower/followee relationships and the intervals indicate that these relationships start at t_{init} *and finish at* t_{end}*. As example, E starts following F at 3 and unfollows at* 6*.*

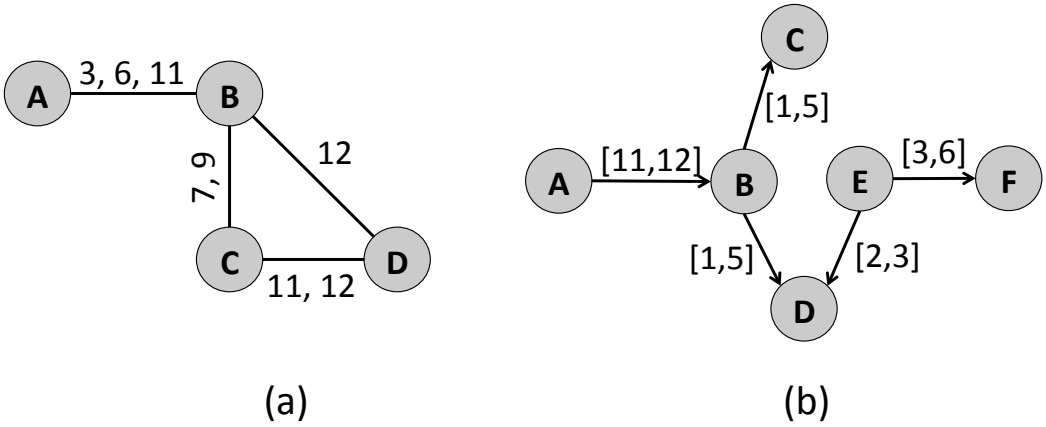

Figure 4 – Temporal networks represented as (a) contact sequence network and (b) interval graph.

2.4 Temporal Measures

The topological structure of static networks can be characterized by an abundance of measures (COSTA et al., 2007). In essence, such measures are based on connections between neighboring nodes (such as the degree or clustering coefficient), or between larger sets of nodes (such as path lengths, network diameter and centrality measures). When the additional dimension of *time* is included in the network picture, many of these measures need rethinking.

2.4.1 Time-respecting Paths

In a static graph, a path is simply a sequence of edges such that one edge ends at the node where the next edge of the path begins (such as A to B to C to D in Figure 4(a)). In a temporal graph, paths are defined as sequences of contacts with increasing times that connects sets of nodes – the *time-respecting paths* (KEMPE et al., 2000). As example, in

[1] Tweet is a slang term to describe what you post in Twitter. A mention is a Tweet that contains another user's @username anywhere in the body of the Tweet

Figure 4(a) there are time-respecting paths from A to C (e.g., <(A,B,6),(B,C,9)>), but none from D to A.

A difference between directed and temporal networks is that the paths are not transitive. The existence of time-respecting paths from i to j and j to k does not imply that there is a path from i to k, as seen in the example above – a path from i to k via j exists only if the first contact between j and k takes place after the last contact of i and j. This is related to a fundamental property of time-respecting paths: the existence of a time-respecting path that begins at i at time t' and leads to j does not guarantee that such a path between i and j exists for $t > t'$.

Thus, time-respecting paths define which nodes can be reached from which other nodes within some observation window $t \in [0, T]$. The set of nodes that can be reached by time-respecting paths from node i is called the *set of influence* of i. In the context of social networks, for example, this set of influence will be reached by opinions posted by i. Remarking the Figure 4(a), for $t \in [2, 7]$, we have that the set of influence of A is $\{B, C\}$.

The *duration* of a time-respecting path is the difference between the last and first contacts on the path (PAN; SARAMÄKI, 2011). Analogously to the shortest paths in static networks that define the geodesic distance, in temporal networks there are the *fastest time-respecting paths*, that indicate paths with lowest duration. In our example (Figure 4(a)) there are many time-respecting paths from A to D in the observation window $[3, 15]$: <(A,B,3),(B,D,12)>, <(A,B,3),(B,C,7),(C,D,11)>, etc. The fastest time-respecting path is: <(A,B,11),(B,D,12)> with duration 1.

2.4.2 Temporal Centrality Measures

In network theory, numerous centrality measures have been defined for identifying important nodes beyond the degree, e.g. with respect to their average distance to other nodes or importance for shortest paths connecting other nodes. Bringing to temporal networks scenario, a rather straightforward approach is to replace the role of paths in static networks by time-respecting paths, as proposed in (HOLME; SARAMAKI, 2012).

Latency. The concept of *latency* in a temporal network emerged from the need to keeping track of the age of information that a node has about other nodes. Consider the node i at time t in a temporal network over which information spreads. Let $\phi_{i,t}(j)$ be the latest time before t such that information from j have reached i. The *information latency* is defined as:

$$\lambda_{i,t}(j) = t - \phi_{i,t}(j) \tag{1}$$

and thus it is a measure of how old i's information coming from j is at time t. In an opinion propagation scenario, to detect whether a user has been influenced by her friendship network, is an important issue to quantify the information latency that she

has been receiving so far. For example, from Figure 4(a), let us compute the information latency in node C coming from A at time $t = 10$. We have that $\phi_{C,10}(A) = 9$ and $\lambda_{C,10}(A) = 10 - 9 = 1$. Supposing that the times in graph represent days, and today is $t = 10$, we conclude that C received information originated from A one day ago.

Temporal Closeness. The closeness centrality C_C (BOCCALETTI et al., 2006) for static networks is defined as

$$C_C(i) = \frac{N-1}{\sum_{j \neq i} d(i,j)} \tag{2}$$

where N is the number of nodes and $d(i,j)$ is the geodesic distance between i and j, i.e. the closeness centrality measures the inverse distance to all other nodes and is high for nodes who are close to all others. Similarly, for temporal networks, the idea is to measure how quickly a node may on average reach other nodes:

$$C_C(i,t) = \frac{N-1}{\sum_{j \neq i} \lambda_{i,t}(j)} \tag{3}$$

where $\lambda_{i,t}(j)$ is the latency between i and j.

Temporal Betweenness. The betweenness centrality C_B (BOCCALETTI et al., 2006) is another important centrality measure based on shortest paths, measuring the fraction of shortest paths passing through the focal node (or edge). For static networks, betweenness centrality is formally defined as

$$C_B(i) = \frac{\sum_{i \neq j \neq k} v_i(j,k)}{\sum_{i \neq j \neq k} v(j,k)} \tag{4}$$

where $v_i(j,k)$ is the number of shortest paths between j and k that pass i, and $v(j,k)$ is the total number of shortest paths between j and k. Thinking about temporal networks, we have:

$$C_B(i,t) = \frac{\sum_{i \neq j \neq k} w_{i,t}(j,k)}{\sum_{i \neq j \neq k} w_t(j,k)} \tag{5}$$

where $w_{i,t}(j,k)$ is the number of *fastest time-respecting paths* in the window of observation t between j and k that pass i and $w_t(j,k)$ is the total amount of *fastest time-respecting paths* in t.

2.5 Evolutionary Network Analysis

Consider a large dynamic (or streaming) network, how can we automatically learn the temporal structural behaviors of individual nodes and identify unusual activities? For instance, in a phone call network, we may want to learn the behavioral roles of individuals and monitor changes over time. This would allow us to detect when a person begins having unusual behavior (a potential fraud) or when he/she becomes an important client for the telecom company.

According to Aggarwal e Subbian (2014) there are two modes of analyzing evolving networks: *maintenance methods* and *analytical evolution analysis*. In the first, it is desirable to maintain the results of the data mining process continuously over time. For example, the resultant model of a preference mining method will evolve as the structure of the graph changes overtime. Therefore, this model will become stale over time, and the goal is to maintain the freshness of the model and, consequently, the final results. Another examples are the traditional social network mining tasks brought to the evolutionary context: evolutionary clustering methods (GUPTA et al., 2011; KIM; HAN, 2009), node classification where the nodes labels have to be predicted or adjusted as the network grows (AGGARWAL; LI, 2011), link prediction (AGGARWAL et al., 2012) and real-time estimation of network metrics (BAHMANI et al., 2010).

In the second, the idea is to directly quantify and understand the *changes* that have occurred in the underlying network. Such models are focused on modeling the change, rather than adjusting for the freshness in the results of data mining algorithms on networks. Graph evolution rules that simulate real-world networks behaviors in relation to degree distribution, clustering coefficient or average path length are examples of this analytical evolution analysis mode. Evolution laws such as preferential attachment (CHAKRABARTI et al., 2010; ALBERT; BARABÁSI, 2002), densification (LESKOVEC et al., 2005) and competition in evolving networks (BIANCONI; BARABÁSI, 2001) have been proposed. Also, here there are studies about nodes' role dynamics for understanding network evolution (ROSSI et al., 2012), influence analysis (AGGARWAL; SUBBIAN, 2012) and, specially, temporal outliers or anomaly detection (AKOGLU et al., 2015).

The goal in this thesis is to use *analytical evolution analysis* techniques in evolving networks for tracking user preference dynamics. In what follows we highlight a topic in this field, namely, strategies for processing evolutionary networks.

2.6 Strategies for Processing Evolutionary Networks

When we talk about ways to process networks evolving over time, we must first consider the evolutionary character of these networks. For instance, in email networks, links are added each minute, whereas in co-authorship networks, edges are added in a scale of

weeks or months. So, even considering the aspects of evolution and temporal information, we should perform offline or real-time algorithms, depending on the context.

2.6.1 Network Evolution

We summarize different strategies for processing evolving networks. These strategies are related with the network – or samples of it – that are considered during the analysis, impacting on the (i) model adjusting process, (ii) algorithms performance and (iii) semantic interpretation that we are interested in. We refer to *temporal ordering of edges* the incoming order of edges in the network, i.e., if a temporal network is considered in the analysis, as discussed in previous Section 2.3.

In Figure 5 we illustrate an edge stream scenario that will be used as input for exemplifying the strategies.

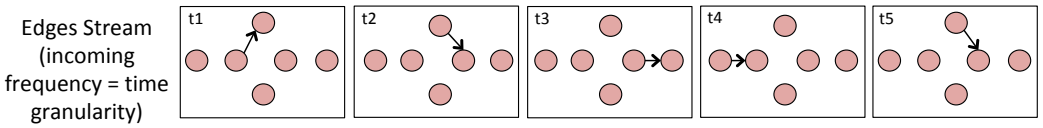

Figure 5 – Edges stream example. At each time step edges are incoming. For didactic reason, the incoming frequency is equal to the time granularity.

- *Slowly evolving networks.* Most of works proposed in the past decade consider these strategies when processing networks (GUHA; MCGREGOR, 2012). They are intuitive and straightforward, as shown in Figure 6.
  - *Batch processing.* The whole network is just batch processed considering the temporal order of the edges. Classical graph algorithms like Dijkstra are used in this scenario.
  - *Snapshots.* At each time $t1, t2, ...$ a network snapshot is considered. Here, the temporal order just make sense if time granularity of snapshots is greater than incoming order.

- *Streaming networks.* This scenario is far more challenging because of the computational requirements and the inability to hold the entire graph on the memory. What varies is the window approach and whether the temporal ordering of edges is also taken into account inside the window (Figure 7). Independently of the window strategy, in streaming networks the algorithms must use data structures that can be maintained incrementally.

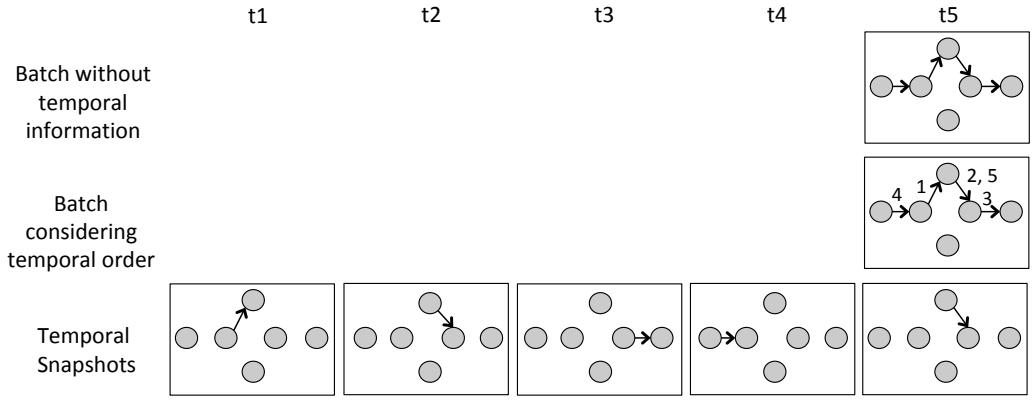

Figure 6 – Strategies for processing slowly evolving networks considering the edges stream from Figure 5.

- *Landmark window.* This strategy is good when we want to keep the history and the new incoming edges are processed considering the whole graph until then. Considering temporal order (temporal networks) is a very common scenario. For instance, face-to-face networks are processed with this strategy in a context of disease spreading (HOLME; SARAMAKI, 2012).
- *Sliding window.* In sliding window, the recent past of the network is sufficient. This is the foundation strategy for sampling networks with forgetting factor (AHMED et al., 2013). Not usually, the temporal order is considered inside the window.
- *Fixed observation window.* Some works process the network with a fixed observation window (WU et al., 2014), in which only a certain interval is interesting. Flight graphs are processed using this strategy, considering a temporal network inside the window.

Besides these window strategies, it is important to mention those based on *sampling the network* (AHMED et al., 2013; TABASSUM; GAMA, 2016). In this approach, algorithms typically need to select a (tractable) subset of the nodes and edges from which to make inferences about the full network.

2.6.2 Algorithms for Streaming Graphs

According to Zhang (2010) a streaming algorithm for massive graph is an algorithm that computes some function for the graph and has the following properties:

1. The input to the streaming algorithm is a graph stream.

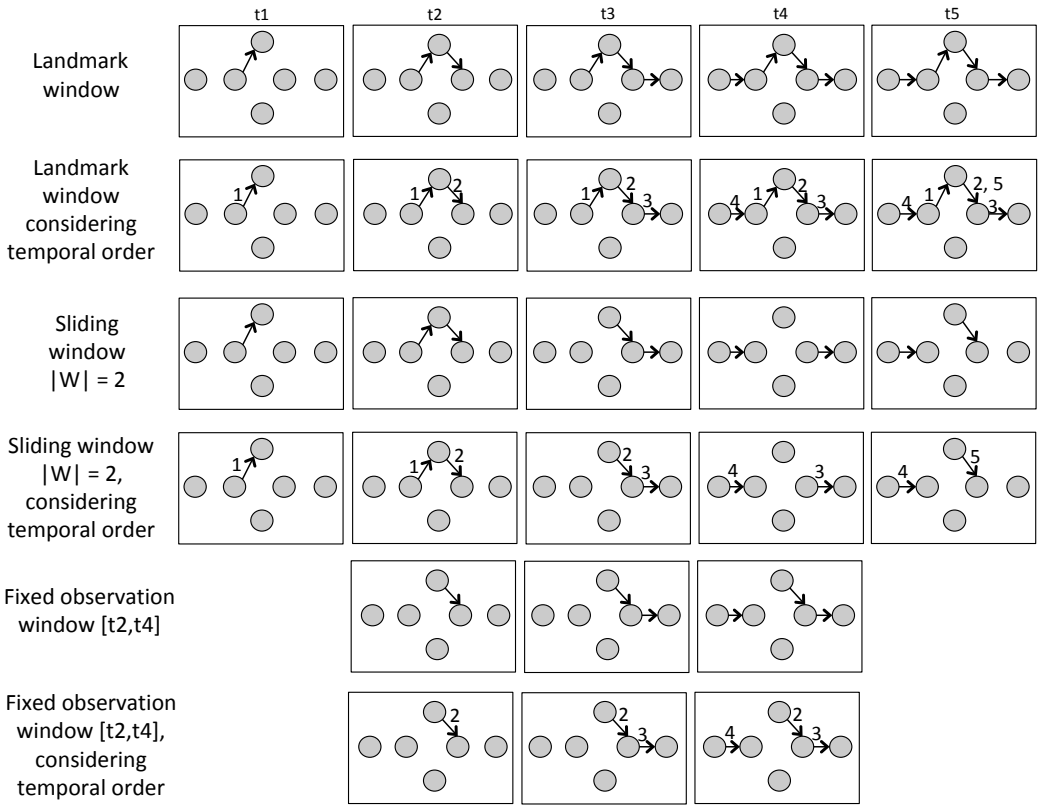

Figure 7 – Strategies for processing streaming networks considering the edges stream from Figure 5.

2. The streaming algorithm accesses the data elements (edges) in the stream in a sequential order. The order of the data elements in the stream is not controlled by the algorithm.

3. The algorithm uses a workspace that is much smaller in size than the input. It can perform unrestricted random access in the workspace. The amount of workspace required by the streaming algorithm is an important complexity measure of the algorithm.

4. As the input data stream by, the algorithm needs to process each data element quickly. The processing time of each data element in the stream is another important complexity measure of the algorithm.

5. The algorithms are restricted to access the input stream in a sequential fashion. However, they may go through the stream in multiple, but a small number, of passes. The third important complexity measure of the algorithm is the number of passes required.

In social network analysis, a common task is to compute measures for the network, often referred to sociometrics (ZAFARANI et al., 2014) – quantitative methods for measur-

ing social relationship. There are one-pass algorithms for node betweenness (KAS et al., 2013b; LEE et al., 2016), closeness (KAS et al., 2013a) and PageRank (BAHMANI et al., 2010). Indeed, there are open issues in this field, specially considering stream processing temporal metrics. We are preparing a survey about *Sociometrics in Streaming Graphs* as detailed in last Chapter 9. The survey will cover literature algorithms that compute sociometrics in graph streams (degree, density, modularity, betweenness, incremental closeness, for cite) and discuss drawbacks and shortcomings of each algorithm.

2.7 Leveraging Networks Terms

There is no consensus in literature in relation to the terms used to express networks that evolve over time. We summarize those most frequently employed.

- **Graph and network.** Graph refers to the data structure composed by nodes and edges. Networks are related to relations among agents connected by links. Throughout this work the terms networks and graphs are used interchangeably.
- **Evolving network or time-evolving network or dynamic network or time-varying graphs.** All these terms refer to networks that are changing with nodes appearing, disappearing, associating and disassociating with each other as time flies.
- **Temporal network.** As detailed in Section 2.3, temporal networks are networks whose order that edges appear and disappear is taken into account during the analysis.
- **Streaming network.** This term is related to the way that the network is observed and processed, specially in scenarios where we do not have the notion of begin and end of incoming data (see Section 2.6).

2.8 Final Considerations

In this chapter we have presented the preference learning process and how it is used in recommendation systems. User preferences can be roughly defined as strict partial orders among a set of objects.

We have also described the concepts underlying considering edges incoming order when analyzing temporal networks. In static networks, whether directed or not, if A is directly connected to B and B is directly connected to C, then A is indirectly connected to C via a path over B. However, in temporal networks, if the edge (A, B) is active only at a later point in time than the edge (B, C), then A and C are disconnected, as nothing can propagate from A via B to C. Thus, the time ordering can matter a lot. Temporal metrics have been employed in order to better quantify real social influence of nodes over

time. Consequently, temporal centralities like betweenness and closeness should consider time-respecting paths.

Lastly, we have discussed about networks streaming processing issues. Temporal scenarios require the design of specific strategies related with the network (or samples of it) that is considered during the analysis. We summarized them specially by means of the window strategy.

Chapter 3

Related Work

Our work is related to a number of areas of study, including preference dynamics, user preferences in social networks and event detection in evolving networks. In this chapter we identify, organize and discuss the state of art. Figure 8 is an illustrative schema showing how we explore related research. The originality of our proposal lies at the junction of these topics.

First, we present works exploring preferences evolution over time. Most of them focus on time-aware recommendation considering user profiles variation over time. Moreover, there are approaches similar to ours performing analytical evolution to describe and predict changes in user preferences, behavior, sentiments and actions. Then we focus on works combining preferences with social networks. We describe existent preference data modeling seeking to discover how these preferences are extracted from the network. Related work here varies a lot according to data domain. Finally, there are researches analyzing social networks over time. This evolving networks analysis can occur considering the network structure evolution or the network content evolution. We present related work on both directions focusing on event detection task, which is the foundation of our proposal.

Many approaches have been used the term *user preferences* for different purposes. In recommender systems, this term refers to user profiling, i.e., the way that users' tastes are *represented*, generally by means of a feature vector or a tensor. Considering the general Artificial Intelligence (AI) research, this user preferences term refers to the preference *order* over objects or ranking inferred by a preference model. Throughout our related work analysis, we will discriminate which *user preferences* are being addressed by respective approaches. In our work we refer to *user preferences dynamics* as well as in AI research line: dynamics of *preference order* induced over objects.

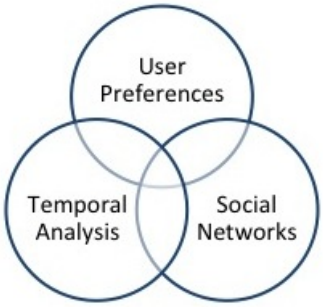

Figure 8 – Main related research topics of this thesis. The originality of this thesis lies at the junction of these three topics.

3.1 Temporal Dynamics of User Preferences

According to Liu (2015) modeling dynamics of preferences requires to address two challenges: (i) precise preference representation and user profile building and (ii) accurate preference evolution inference. Time-aware personalized recommendation systems generally represent preferences as concrete feature vectors and consider the historic of past profiles to predict preferences. A drawback in these approaches is the serendipity (over-specification) (DEBNATH et al., 2008). There are more aspects to be explored than the observation of the past. In another line of research the nature of preferences evolution is studied seeking to describe them qualitatively, quantitatively or predicting changes. The latter is closer to our proposal as we detail in what follows. Table 2 summarizes a comparative view among works.

3.1.1 Time-aware Personalized Recommendation

The topic of user preferences is largely used for personalized recommendation. In a recommendation system application, users have preferences for items. The data itself is represented as a utility matrix, i.e., a data structure that gives for each user-item pair, a value that represents the degree of preference of that user for that item. Values can be ratings, level of interaction, probabilities and so on (LESKOVEC et al., 2014).

In this context, since (KOREN, 2009) temporal dynamics have been incorporated in recommendation systems. In fact, the recommendation model developed by Koren (2009) was a landmark in the use of temporal information to improve recommendation quality. The model incorporated several time-sensitive user and item biases in the standard factor model. Gradual changes in user preferences over time were captured using a linear function. Their model showed that modeling temporal dynamics in user choices was essential for improving the performance of the recommender.

In the same direction, Xiang et al. (2010) deals with temporal recommendation using graph model. The proposal is to build a bipartite graph containing links of the type *<user,item>* and *<item,session>* (STG - Session-based Temporal Graph) which simulta-

neously models users' long-term and short-term preferences over time. A user preference p can be *injected* in the graph and, then, propagated. The datasets used for experimental evaluation are CiteULike and Delicious.

A tensor that has time as a dimension was used in (RAFAILIDIS; NANOPOULOS, 2014) to provide recommendations. In this approach, to account for the fact that user preferences are dynamic and change individually, the authors propose a measure of user preferences dynamics (UPD) that captures the rate with which the current preferences of each user have been shifted:

$$UDP_u = 1 - \frac{|I_{cur} \cap I_{prev}|}{|I_{cur} \cup I_{prev}|} \tag{6}$$

where I_{cur} denotes the set of items that user u has interacted at the current time slice and I_{prev} is the union set of the items that u has interacted at all the previous time slices. Low UPD_u values indicate that u preserved his preferences, whereas high ones correspond to users' u high tendency to change his preferences at the current time period. Recommendations are generated based on factorizing the tensor, by weighting the importance of past user preferences according to their UPD values. The empirical evaluation was done over music domain from *Last.fm* dataset.

More recently, Liu (2015) proposed to capture user's dynamic preference to provide timely personalized recommendation. The author proposes to apply a Gaussian Process (GP) regression to capture users' preference. First, LDA (Latent Dirichlet Allocation (BLEI et al., 2003)) is applied to extract k topics from text content. Then, each interaction between user u and item i is represented by a topic distribution and the user profile ρ_u records the topic distributions of u's past interactions and the corresponding timestamp. For user u the evolution of each topic is a time series. With GP is possible to predict u's future preference, i.e., topic distributions of the items that u is likely to interact in the next interactions. Once the future preference is inferred, the author computes the Jensen-Shannon divergence between u's future topic distribution and the topic distribution of each item candidate to provide top-N recommendation.

The work of Wu et al. (2016) also deals with temporal behavior of preferences in recommendation field. Specially, a social network is used to model interactions among users and items rather than a utility matrix. The social network is composed by two types of nodes: users and items. A user-user interaction is represented by a social link and a user-item interaction is a consumption behavior link or the users' preferences. The important point is that this work also deals with temporal behavior of preferences through social networks. Given the social network structure, the authors propose a probabilistic model to predict preferences, that are represented through a tensor $C \in \mathcal{R}^{N \times M \times T}$, where N is the number of users, M is the number of items and T the number of time steps. This model is founded on link prediction theory in social network analysis (ZAFARANI et al., 2014).

Finally, is important to mention the topic of stream recommender systems. Researchers here seek for incremental adjusting models for recommendation (CHANDRAMOULI et al., 2011; HUANG et al., 2015; VINAGRE et al., 2014; SIDDIQUI et al., 2014). As time goes by and streams of time-evolving preferences come, how to avoid an outdated prediction model? In most common approaches time-evolving preferences are simply ratings that can change over time (SIDDIQUI et al., 2014; PAPINI et al., 2014).

In general, recommendation models assume that user transitions are driven by a static transition matrix. At present, the recommendation community lacks models that predict changes in user preferences (KAPOOR et al., 2013).

3.1.2 Modeling Evolving Preferences

According to psychology field, humans choice probabilities are directly given by their past choices. For example, consumers were found to exhibit either a short term loyalty for their last purchased brand (inertia) or devaluation for the last purchased brand (variety seeking) (GLANZER, 1953).

In (SUN et al., 2008) the objective is to measure changes of user preferences. Items and user preferences both are modeled as a vector of features. Items are N-vectors where each position is a feature and contains the score of the feature i for the given item. A preference vector is a N-vector such that the value is positive if the user prefers higher score of the feature i, or negative otherwise. The values of preference vector come from click streams (implicit feedback) from CiteSeer database. To measure the changes of user preferences the authors define a metric based on the accuracy of predicting future user actions:

$$S = \frac{\bar{A}}{\sigma(A) + 1} \tag{7}$$

where $A = \{a(t_1), a(t_2), ..., a(t_T)\}$ is a time series and each $a(t)$ is the accuracy to predict the user actions in time t. $\bar{A}$ is the average prediction accuracy and $\sigma(A)$ is the standard deviation of the series. The intuition is: if the accuracy of predicting future actions is getting low then changes are occurring and thus the change metric score is high.

The evolving aspect of preferences was studied in (SCHLITTER; FALKOWSKI, 2009). This approach tracks communities of users according to their music tastes. A user is defined by a profile that describes his music preferences. The profile is a vector representing the user's music preferences. Each element in the vector profile represents an artist, defined by a set of genres. Communities of users are determined by clustering similar profiles. The method: (1) represent users by vector profiles; (2) calculate profiles similarity through cosine; (3) build a similarity graph G where nodes are profiles (the users) and the edges represent profiles with similarity higher than a threshold; (4) detect communities

in this graph; and (5) track communities evolution. Thus, the analysis here is over the evolving aspect of communities that represent users with similar preferences.

In (TAN et al., 2010) the authors discuss how to simultaneously model the social network structure, user attributes and user actions over time. A user's action at time t is generated by her latent state at t, which is influenced by her attributes, her own latent state at time $t-1$ and her neighbors' states at time t and $t-1$. Examples of actions are whether a user discusses the topic "Haiti Earthquake" in Twitter, whether a user adds a photo to his favorite list in Flickr or whether a researcher publishes a paper on a specific conference in Arnetminer.

Moore et al. (2013) have also studied user preference dynamics in music field. Preferences are represented by the user's playlists. A playlist is modeled using a first-order Markov model so that the probability of the next song in a playlist depends only on the current song and the user. Thus, preferences here are simply the common behavior, the sequence of songs. Starting from a static playlist the authors incorporate a temporal model under which the embedding can change over time, providing trajectories for users and songs through embedding space. Embeddings are models that learn positions for objects in a metric space. The resultant trajectories give insight into how the appeal of songs and artists changed over time. So, the changes are modeled as trajectories in a metric space.

A pioneer study focused on analyzing and modeling user sentiment dynamics in Twitter is (MACROPOL et al., 2013). The authors hypothesize that there is a strong relationship between users' activity acceleration and topic sentiment change. For instance, let us suppose that user John used to tweet about Obama. Analyzing his historical of posts, in a certain time t it is possible to verify an acceleration on his posting frequency, i.e., John posted a lot of tweets, higher than his average posting frequency. Thus, probably at time t there is a *sentiment change* of John in relation to Obama.

From a theoretical viewpoint, Thimm (2013) proposes a model for doing preference aggregation under preference changes scenario. It states that preference orders change over time as well as the motivation for our work. A framework for preference change that distinguishes between two atomic types of changes to a preference order, namely, *weakening* and *strengthening* of a specific domain element has been developed. A preference profile is composed by a set of preference orders, related with different attributes. For example, a preference profile can be over music genre, singer and music rhythm.

(KAPOOR et al., 2013; KAPOOR, 2014) are two of the most expressive works on predicting changes in user preferences. It is based on the idea that predicting temporal choices is not a trivial task just based on past behavior. It is necessary to predict the changes rather than just react to them. Their approach is founded on psychology theories that states the presence of both stickiness and devaluation effects in user preferences. The authors analyze user music listening behavior to extract signals of stickiness and

boredom. They demonstrate the use of hazard functions for measuring these phenomena. This work is orthogonal to ours. While in (KAPOOR, 2014) stickiness and boredom guided the preference change model, we seek to understand user preferences dynamics founded on social influence.

Existing research on temporal behavior prediction is also related with ours. In (ZHANG et al., 2014b; ZHANG et al., 2014a) a generative dynamic behavior model is proposed. The model considers the temporal item-adoption behaviors as joint effect of dynamic social influence and varying personal preference over continuous time. Through a dynamic preference probability space is possible to capture the dynamics of behaviors smooth variation over time. Our approach also uses social influence in an environment of continuous preferences but the focus is on preferences change prediction. User behaviors are different from user preferences. Behaviors are actions like *I play football* or *I buy a product* and preferences are order relations established between two objects, for instance *I prefer football than basketball*, *I prefer black cars than white cars*.

Table 2 – Comparing works in relation to temporal dynamics of user preferences. FV: feature vector; T: tensor; SFV: streams of feature vector; UP: user playlist; OPO: order over pairwise objects.

Work	Social Net. Data	Rec Sys	Analytical Evolution Evolution Aspect	Analytical Evolution Evolution Analysis	Preference Model
Time-aware recommendation					
Xiang et al. (2010)	✓	✓			FV
Rafailidis et al. (2014)		✓			T
Siddiqui et al. (2014)		✓			SFV
Liu (2015)		✓			FV
Wu et al. (2016)	✓	✓			T
Evolving preferences					
Sun et al. (2008)			pref change	quantitative	FV
Shlitter et al. (2009)			users' similarity	communities	FV
Tan et al. (2010)	✓		social action	predictive	-
Moore et al. (2013)			pref change	trajectories	UP
Macropol et al. (2013)	✓		sentim. change	predictive	-
Thimm (2013)			pref aggregation	qualitative	OPO
Kapoor (2014)			pref change	predictive	FV
Zhang et al. (2014b)	✓		user behavior	predictive	-
This work	✓		pref change	predictive	OPO

3.2 User Preferences and Social Networks

Works than join the topics of user preferences and social networks can be analyzed from three main perspectives: social recommendation, preferences propagation and mining

preferences from social networks. In what follows, we describe related work from these perspectives, highlighting *preference data*. After all, what are these preferences that come from social network? In Table 3 we present a comparative summary of related work from the point of view of preference data coming from the Online Social Network (OSN). Notice that works vary a lot in the way of representing users' profiles. Most of the time they are extracted from OSN data using some text mining method.

3.2.1 Opinion and Preferences Diffusion in OSN

When considering time dimension, researches go in the direction of opinion propagation and diffusion of preferences. Generally, preferences are modeled from information outside the network and then the network structure is used in cascading models and influence detection of these preferences. They are not mined from the network.

In (LOU et al., 2013) a preference profile p_v of an individual v is a k-dimensional vector that represents v's preference toward k different candidates. The jth element in p_v is an integer in $[1; k]$ indicating this individual preference for candidate j (smaller numbers denote higher ranks). The preference data comes from citations. In scientific research papers, citations implicitly reveal the research interests of authors. In other words, the authors consider that the acts such as citing or submitting to the journals or the conferences would reveal the authors' interests. By utilizing this fact, they infer the researchers' preference from their corresponding top frequently-cited conferences and journals. They chose the top 16 journals that possess most papers as the candidates to construct the preference lists and the preference criteria is based on the citation count of corresponding journals. The same strategy has been used in (ZHANG et al., 2011), from DBLP dataset.

3.2.2 Social Recommender Systems

Social recommender systems leverage users' friendship or interaction information to predict their preferences. Here, social information is used to improve prediction models (GUY, 2015) and address user cold start problems (FELÍCIO et al., 2016). Besides the works (WU et al., 2016; XIANG et al., 2010) previously presented in Section 3.1.1, we highlight the work of Wenzel e Kießling (2016) that proposes a database-driven recommendation approach using Online Social Networks. Besides social data (friends), activity, spatial and demographic information has also been used. The framework steps are: (1) select user preferences with a database preference query (`SELECT... FROM... PREFERRING...`). Then, (2) user models are item sets (one for each dimension) and (3) the similarity between two users is given by the comparison of one dimension at a time, resulting in a vector s_{u_a,u_b}. (4) Items consumed by most similar users are recommended.

3.2.3 Preference Mining

In this topic, researchers seek to answer *how can we extract and model preferences from social networks?* In particular, given personal preferences about some of the social media users, how can we infer the preferences of unobserved individuals in the same network?

(ABBASI et al., 2014) is an approach that infers users' missing attributes and preferences from networked data. Their method combines two characteristics: *weighted-vote relational neighbor (local prediction algorithms*) and *social dimensions (global prediction algorithms).* The method is called Local Social Dimensions. Basically, the paper is founded on *homophily* and *influence* properties from influence theory that states that a user's preferences are influenced by the influential users in her social circle. A Facebook fan pages dataset has been used and the preferences are labels indicating page's political view. Thus, it is a label ranking preference learning approach.

In (LI et al., 2014), user preferences are also considered in the context of social networks. How can we reason about user preferences given only weak sources of knowledge about users' attributes, preferences and social ties? The authors propose a text extraction system for Twitter to automatically extract structured profiles from the text of users' messages. Profiles are a set of logic rules of the type: `WORK-IN-IT-COMPANY(A)->LIKE-ELECTRONIC-DEVICE`, meaning that "people who work in IT companies like electronic devices." After building users profiles, using probabilistic logic is possible to infer preference order over a given set of objects.

We have proposed mining preferences from social media text using comparative sentences (PEREIRA, 2015; PEREIRA; AMO, 2015). A comparative opinion is a statement like *car X is much better than car Y* from which we can clearly extract a preference order: car X is preferred over car Y. Using a genetic algorithm we have mined comparative sentences from tweets about PlayStation, Wii and XBox video games. This approach is promising and can be adopted in the future in our work. It is necessary to propose a method that automatically converts comparative sentences into pairwise preferences over objects. Due to the lack of available data, in this thesis we do not use this preference mining technique.

The preference mining task is generally formulated as user profiling, specially in recommendation models. There is a lack of specific techniques for mining preferences (preference order relation) purely from social networks.

3.3 Event Detection in Evolving Networks

Our proposed solution to the issue of analyzing User Preference Dynamics is founded on event detection in social networks that are constantly evolving. According to Ranshous et al. (2015) the problem of *event detection in networks* can be stated as *given a fixed graph series G or graph stream $\mathcal{G}$, find a time point at which the graph exhibits behavior*

Table 3 – Comparative view of *preference data* extracted from online social networks (OSN).

Work	Source of Preference			User Profile	Dataset
	OSN Structure	OSN Data	Outside OSN		
Lou et al. (2013)		✓		feature vector	Citations
Abbasi et al. (2014)		✓		political view labels	Facebook
Li et al. (2014)	✓	✓		logical rules	Twitter
Pereira (2015)		✓		comparative sentences	Twitter
Wenzel et al. (2016)			✓	PreferenceSQL	Outdooractive
This work		✓		topics	Twitter/Music

sufficiently different from the others. In fact, detecting events is a common element in many methods that aims to further discoveries like *changes*, *anomalies* and *bursts* in graphs (AKOGLU et al., 2015; ZIGNANI et al., 2018). Detecting events means identifying that something happened out of the ordinary.

We highlight two main directions from this topic: Online Social Networks Event Detection and Event Detection over Dynamic Graphs. The former is related with social streams processing, where the messages' content being published in the network is analyzed. For example, a set of posts sharing the same topic and words within a short time. The latter focus on events in the network structure evolution, for instance an increasing number of new connections in the social graph. Our proposal concentrates on the latter category: discovering events in evolving networks.

At a high level, our goal is to identify events that occur in a given node considering a graph stream processing environment. Table 4 summarizes a comparative view of related work on this topic.

3.3.1 Event Detection in Social Streams

The work (AGGARWAL; SUBBIAN, 2012) incorporates network structure in event discovery over purely content-based methods. Each text message is associated with at least a pair of actors in the social network. The events detected are also related with topics evolving.

Burst events are generally related to topic evolving detection and tracking (CORDEIRO; GAMA, 2016). These works are looking for events like hot buzz words, what are the users' sentiments about a product release or how a specific topic is evolving.

In (BUNTAIN; LIN, 2016) the goal is to track interest profiles in real time by detecting bursts in Twitter's social media stream in real time using linear regression. Time-critical analysis of social media stream is also a related problem. Methods to rapidly categorize messages into a series of classes of interest and also to capture novel emerging categories

have been proposed (IMRAN et al., 2016; IMRAN et al., 2013). In (IMRAN et al., 2016) the goal is to categorize crisis-related messages on Twitter during natural or man-made disasters. Semi-supervised clustering techniques are used in this approach.

The difference of this line of work with our approach is that our method does not use text during the event detection task. We assume numerical measurements on the graph nodes. Though stream processing, these approaches focus on social content while ours concentrates on social network structure as we show in what follows.

3.3.2 Event Detection over Dynamic Graphs

Processing graphs as streams is an incoming problem. The work (BIFET et al., 2011) is one of the most complete when considering data mining in evolving graph streams. The focus, however, is on mining closed graphs, not on event detection. In (FAIRBANKS et al., 2013) a framework for processing graphs as streams is proposed for the link prediction task. This framework considers the cumulative grown of the graph, not addressing the space saving issue (GAMA, 2010).

In (ROZENSHTEIN et al., 2014) the authors consider the problem of mining activity networks to identify interesting events, such as a big concert in a city, or a trending keyword in a user community in a social network. The algorithms are founded in geo-spatial event detection information. Any stream processing strategy is addressed.

The most studied events in dynamic networks are anomalies and bursts (CORDEIRO; GAMA, 2016). Anomaly detection refers to the discovery of rare occurrences in datasets (AKOGLU et al., 2015; RANSHOUS et al., 2015). The most representative work in anomaly detection for dynamic graphs is (IDE; KASHIMA, 2004). It addresses the problem considering a time sequence of graphs (graph sequences). The focus is on faults occurring in the application layer of Web-based systems. First, they extract activity vectors from the principal eigenvector of dependency matrix. Next, via singular value decomposition, it is possible to find a typical activity pattern (in $t-1$) and the current activity vector (t). In the end, the angular variable between the vectors defines the anomaly metric. The network processing is through snapshots, not in a streaming fashion. Akoglu e Faloutsos (2010) used this Eigen Behavior based Event Detection (EBED) method to detect events in SMS interactions – a *who-texts-whom network*. The main difference in comparison to ours is that it detects events in a global perspective of the network, while ours is node-centric.

Recently, (EBERLE; HOLDER, 2016) proposed to discover anomalous subgraphs in graph streams using a change detection metric. Though the goal is to match patterns of anomalous subgraphs, the idea of detecting change in streams of graphs is very close to ours. In a high level, the authors compute graph properties GP as the graph evolves and then compare, using average and standard deviation, if there is an abrupt change in these GP. If yes, the change has been detected. The algorithm processes incoming edges

in batches using sliding window strategy. Our proposal focus on node properties tracking instead of global GP.

Table 4 – Event detection in evolving networks

Work	Network Content	Network Structure	Pattern/Task	Stream Processing Text	Stream Processing Graph
Processing social streams					
Aggarwal et al. (2012)	✓	✓	events	✓	
Buntain e Lin (2016)	✓		trends	✓	
Imran et al. (2016)	✓		classification	✓	
Processing dynamic graphs					
Ide et al. (2004)		✓	anomaly		
Akoglu et al. (2010)		✓	events		
Bifet et al. (2011)		✓	closed graphs		✓
Fairbanks et al. (2013)		✓	link prediction		✓ (partial)
Rozenshtein et al. (2014)		✓	geo-spatial events		
Eberle et al. (2016)		✓	anomaly		✓
This work		✓	events		✓

3.4 Final Considerations

We have organized the state of art according to three main topics related with our proposal: user preferences, social networks and temporal aspects of both. In literature these topics are combined two by two what leads to the originality of our proposal by combining all of them.

There are different ways of studying temporal dynamics of preferences. The works in the recommendation area consider as preferences user profiles, usually represented by a feature vector. So, analyzing preferences over time means to observe how these vectors were in the past to predict correct vectors in the future. In another line of research, preferences are considered as object-order relationships (pairwise preferences or rankings). In these works the task is not to predict the next ranking, but rather to describe how order evolves over time. Our proposal resembles these works when analyzing preferences changes through social networks.

The combination of preferences with social networks was analyzed under the optics of preference data extracted from the network. We have shown that there are few proposals that mine preferences directly from the social network, both as a user profile or as preference order relationships.

Finally, we discuss the state of the art in social network analysis over time, with a special focus on the task of detecting events, which is our proposal. A large collection of works search for events in social streams using techniques of text mining. Another

direction, aligned with our proposal, detects events in graphs that evolve their structure. We have highlighted works that process graphs as streams which is our scenario.

Chapter 4

User Preference Dynamics

The discussion about tastes over time is not new in literature. Some works reported this phenomena from psychological factors of familiarity, boredom and exploration (KAPOOR, 2014; KAPOOR et al., 2013). Others, aim to model categories of preference change: derivational, temporal models, consistency-preserving models and evolutionary models (YANOFF; HANSSON, 2009). In this chapter we come to the task of defining our problem. After all, what kind of user preferences we address and how do we handle temporal dynamics?

We distinguish preferences from opinions. Opinions are defined as a point of view, a belief, a sentiment or a judgment that a person may have about an object; preferences, as we have defined, involve an ordering on behalf of an agent and thus are relational and comparative (CADILHAC et al., 2015), often referred as pairwise preferences.

There are different ways to perceive how preferences of a given user vary over time. One can consider *novelty*, for instance the emergence of a new accessory in fashion domain or new car models. Another way is considering *selectivity*, where the user becomes more or less restrictive in her preferences. And finally, we can also observe *changes*, i.e., whether in the past the user preferred some item and nowadays not anymore. Our approach focus on the latter: *preferences changes*.

The chapter begins with the proposal of a temporal preference model able to represent and reasoning with preferences on time. Then, we describe how to detect events of *changes* on temporal preferences. We conclude presenting the problem formalization that we address in this thesis.

4.1 Temporal Preference Model

There is no consensus on the definition of preferences dynamics (LIU, 2011). We adopt the following definition:

Definition 4.1.1 (User Preference Dynamics (UPD)) *UPD refers to the observa-*

tion of how a user evolves his/her preferences over time.

A *preference* is an *order relation* between two objects. For example, when a user says: *"I prefer sports to politics"*, if we order *sports* and *politics* in a ranking, we can clearly identify that *sports* will be in the top position.

Definition 4.1.2 (Temporal Preference Relation $\succ_t$) *A temporal preference relation (or temporal preference, for short) on a finite set of objects $A = \{a_1, a_2, ..., a_n\}$ is a strict partial order over A inferred at time t, i.e., a binary relation $R \subseteq A \times A$ satisfying the irreflexivity and transitivity properties at t. Typically, a strict partial order is represented by the symbol $\succ$. Considering $\succ_t$ as a temporal preference relation, we denote by $a_1 \succ_t a_2$ the fact that a_1 is preferred to a_2 at t.*

Definition 4.1.3 (User Temporal Profile Γ_t^u) *User temporal profile Γ_t^u is the transitive closure (TC) of all temporal preferences of user u at t.*

Example 4.1.1 *Let $A = \{sport, tv, religion, music\}$ be the set of objects in our running domain representing themes of interest of user A. Figure 9 illustrates the temporal preferences of A at days 1, 4 and 9 through better-than graphs. Remark that an edge (a_1, a_2) indicates that a_1 is preferred to a_2 and edges inferred by transitivity are not represented. We have: $\Gamma_1^A = \{sport \succ_1 tv, tv \succ_1 religion, sport \succ_1 religion, sport \succ_1 music\}$, $\Gamma_4^A = \{sport \succ_4 tv, tv \succ_4 religion, sport \succ_4 religion, tv \succ_4 music, sport \succ_4 music\}$ and $\Gamma_9^A = \{sport \succ_9 tv, tv \succ_9 religion, sport \succ_9 religion, music \succ_9 tv, music \succ_9 religion\}$.*

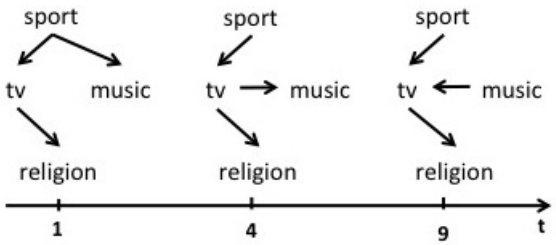

Figure 9 – Better-than graphs representing temporal preferences at days 1, 4 and 9. Edges inferred by transivity are not depicted for better visualization.

4.2 Detecting Changes on Temporal Preferences

A key property of temporal preferences is the irreflexivity. We say that a temporal profile Γ_t^u is *inconsistent* when there is a preference $a_1 \succ_t a_1 \in \Gamma_t^u$. It would means that *"I prefer X better than X!"*, which does not hold for a strict partial order.

Our proposal for detecting preference change is based on the *consistency* of user temporal profiles. The idea is to compute the union of user profiles collected over time, infer

temporal preferences by transitivity considering all timestamps and verify if there is any inconsistency in the resulted set of preferences. If so, we detect an event of preference change. These concepts are formalized in what follows.

Definition 4.2.1 (Temporal Profile Union Ω_t^u) *Two temporal preferences of the type $a_1 \succ_{t-1} a_2$ and $a_2 \succ_t a_3$, can unite to infer a third temporal preference $a_1 \succ_t a_3$, once considering transitivity of both, temporal preference relation and timestamp order. A temporal profile union Ω_t^u is the transitive closure (TC) of all irreflexive relations given by $\Gamma_{t-1}^u \cup \Gamma_t^u$.*

Definition 4.2.2 (Preference Change δ_t^u) *If there is a temporal preference inconsistency in Ω_t^u a preference change has been detected at time t for user u. In other words, a preference change δ_t^u is defined as:*

$$\delta_t^u = \begin{cases} 1 & \textit{if there is a temporal preference inconsistency in } \Omega_t^u \\ 0 & \textit{otherwise} \end{cases} \tag{8}$$

Remarking on Example 4.1.1, the temporal profile union $\Omega_9^A = \{..., tv \succ_4 music, music \succ_9 tv, tv \succ_9 tv, ...\}$ contains the inconsistency $tv \succ_9 tv$. Thus, a preference change has been detected at time 9 ($\delta_9^A = 1$). Intuitively, we have that at day 1, for example, A prefers to read/post/share on her social network news about *sport*, but between *tv* and *religion* she is in the mood for *tv*. On the following days, A's preferences practically do not change, just appearing a preference of *tv* over *music*. However, at day 9, A's presented a preference change, as *music* became preferred over *tv*.

4.3 *PrefChangeDetection* Algorithm

In order to formalize the detection of changes in our temporal preference model, we propose the *PrefChangeDetection* algorithm. The intuition of this algorithm is to analyze *better-than graphs* (BTG) of a user during some observation period T. For each $t \in T$ we compute $BTG_t^u \cup BTG_{union}^u$, where BTG_t^u is the current BTG derived from the temporal profile Γ_t^u, and BTG_{union}^u refers to the accumulated preferences relative to the period of $[t-|W|, t-1]$, for W being a time window. If the resulting $BTG_t^u \cup BTG_{union}^u$ (a temporal profile union Ω_t^u) has at least one cycle (meaning an inconsistency) we have detected a preference change at t. Algorithm 1 formalizes this idea.

In line 4, the union of two better-than graphs corresponds to Ω_t^u computation. The preference revision operation (line 8) consists in transforming BTG_{union}^u in acyclic by removing the oldest edges. We implemented the strategy proposed in (CADILHAC et al., 2015) to obtain a consistent and updated set of preference relations. According to Cadilhac et al. (2015) a *preference revision* is a sequence of two operations: *downdating*

the existing preferences to a maximal subset that is consistent with the new preference, followed by adding the new preference to the result. Therefore, new preferences take priority over old ones.

The size of the observation period determines if we are tracking short-term or long-term preference events. As example of real events, we can cite new product releases and special personal occasions such as birthdays (XIANG et al., 2010). The window W adjusts this feature. In line 11, updating BTG^u_{union} means forget preferences inferred before $t - |W|$.

Algorithm 1 PrefChangeDetection

Input: User u, window W, a vector Γ^u of size $|T|$ containing u's temporal profiles for each $t \in T$
Output: A vector δ^u of size $|T|$ containing u's preference changes for each $t \in T$

1: $BTG^u_{union} \leftarrow \emptyset$ //BTG^u_{union} is the accumulated better-than graph of u
2: **for all** $t \in T$ **do**
3: build better-than graph BTG^u_t from $\Gamma^u[t]$
4: $BTG^u_{union} \leftarrow BTG^u_{union} \cup BTG^u_t$
5: **if** BTG^u_{union} is not acyclic **then**
6: *// change detected*
7: $\delta^u[t] = 1$
8: revise BTG^u_{union} //remove cycle maintaining more recent preferences
9: **else**
10: $\delta^u[t] = 0$
11: update BTG^u_{union} according to W //remove from BTG^u_{union} all $\succ_{t'}$ s.t. $t' < t - |W|$
12: **return** δ^u

Remarking on *PrefChangeDetection* complexity analysis, the time to build a better-than graph (line 3) is $O(P)$ where P is the number of temporal preference relations in Γ^u_t, which in the worst case is the combination $C_{|A|,2}$, for A being the finite set of objects in the domain (the nodes). In line 4, unifying two graphs costs $2O(|A|+|P|)$ where A and P are the set of nodes and edges, respectively. The time to detect if a directed graph is acyclic (line 5) is $O(|A|+|P|)$. Preference revision (line 8) takes $O(|A|)$ which is the time to compute a maximal independent set in graphs. The last operation is to do a graph update (line 11) which in the worst case is $O(|A|+|P|)$. Hence, *PrefChangeDetection*, in the worst case, has complexity of $O(|T|\times 5(|A|+|P|)$.

4.4 Problem Definition

The hypothesis of this work is founded on the idea that social networks are good source of knowledge to investigate how user preferences evolve over time. We have delimited in previous sections that our approach focus on preferences changes. With these concepts in mind, we formulate our problem as:

Given (i) a user u, (ii) an evolving social network $\mathcal{N}$ and (iii) the set of objects in preferences domain A, predict preference changes events $\delta^u_{t'}$, for any $t' > t$.

Remark that in our work we focus on predicting preference changes *events*, i.e., if there will be changes based on the observation of u's evolving network. Another problem that arises from the observation of user preferences dynamics is the prediction of *which* preferences there will be in the temporal profile after changes. This problem of predicting preferences is not addressed in our work.

4.5 Final Considerations

In this chapter, we have introduced the problem of user preferences dynamics from change detection perspective. Basically, the preferences that we handle in this work are order relations – pairwise preferences, and a change occurs whether in the past the user preferred some item not preferred anymore. We have defined a temporal preference model able to handle temporal preferences. A preference change event occurs when there is a temporal preference inconsistency in user profile which could means a *"I prefer X better than X"* situation. We have also proposed an algorithm to detect preference changes events.

The concepts described in this chapter turn feasible our problem definition which consists in *predicting preference change events* given a user, her evolving social network and the set of objects in her preferences domain. In the next chapters we will present several analysis and proposals over the evolving social network $\mathcal{N}$ having in mind the problem: how could we predict preferences changes events just observing the network evolution? In the end, Chapter 7, we finally show an experimental evaluation describing the strategy based on topic modeling for extracting preferences from the social network and further findings.

Chapter 5

Temporal Networks for UPD Analysis

We propose the use of *temporal networks* as network modeling strategy to analyze user preferences dynamics (UPD). According to Holme e Saramaki (2012), temporal networks take into account *when* links appear and disappear. The intuition of using this approach as foundation of our problem solving is that analyzing evolving social networks considering the temporal order that links are formed and disappear impacts on paths taken by the information in the network. We believe this can reproduce more realistically users relations and consequent social influence.

In this chapter, we present an empirical analysis about how centrality values evolve in temporal graphs applied to Twitter social network. We published this investigation in (PEREIRA et al., 2016a). Our goal was to get some insights related to the evolution behavior of a real social network modeled as a temporal graph.

5.1 Social Temporal Networks

Our main goal here is to represent social networks, specially Twitter, founded on temporal graphs theory. As presented on Chapter 2.3, according to Holme e Saramaki (2012), temporal networks can be divided into two classes corresponding to the types of representations: *contact sequences* and *interval graphs*. While in contact sequences, the edges are active over a set of times, in interval graphs, they are active over a set of intervals. In this chapter, we are interested in representing Twitter social network as an interval graph.

Formally, let $G = (V, E)$ be a directed temporal graph, where V is the set of nodes and E is the set of edges of G. A directed edge $e \in E$ is a quadruple $(u, v, t_{init}, t_{end})$, where $u, v \in V$, the direction is from u to v and the time interval $[t_{init}, t_{end}]$ corresponds to the existence period of e in G.

For example, the temporal information of Twitter has the following meaning: the nodes are users and an edge $(u, v, t_{init}, t_{end})$ indicates that v starts following u at t_{init} and unfollows u at t_{end+1} (v follows u during $[t_{init}, t_{end}]$). As we are dealing with a real

dynamic social network, a user can follow/unfollow another user at any time. This is the most interesting aspect that we are investigating: how *following* relationships on Twitter evolve over time and impact on users behaviors?

There are three global variables that must be defined for we start reasoning with temporal graphs: W, R and T. $W = [n, N]$ is the window time of observation of G. R is the retention time of nodes, i.e., the time between information arrival in the node and the instant from which it can be forwarded. T is the edge traversal time. In our Twitter temporal graph representation, we adopted $R = 1$ day and $T = 0$, as tweets are published instantaneously and the average interaction time for posts is one day (WU et al., 2014). Notice that in our context, T and R are defined as global variables. For flight graphs, for example, each edge (flight traversal) and node (airports) must be defined with their respective traversal and retention times. Figure 10 is an example of Twitter as a temporal network – specifically, an interval graph.

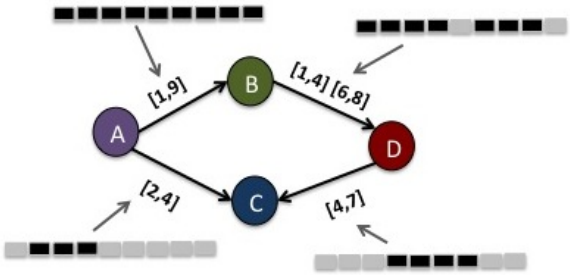

Figure 10 – Twitter as an interval graph. Nodes are Twitter users and an edge $(u, v, t_{init}, t_{end})$ indicates that v starts following u at t_{init} and unfollows u at t_{end+1} (v follows u during $[t_{init}, t_{end}]$). For instance, A followed B from instant 1 until instant 9.

5.1.1 Temporal Paths

The definitions here are grounded on the works (WU et al., 2014; TANG. et al., 2010; NICOSIA et al., 2013) with soft adaptations. The topological structure of static networks can be characterized by an abundance of measures (COSTA et al., 2007). When the additional dimension of time is included in the network picture, many of these measures need rethinking. So that the concept of geodesic distance cannot be limited to the number of hops separating two nodes but should also take into account the temporal ordering of links.

A *temporal path* $P_{u,v}$ in a temporal graph G is a sequence $P_{u,v} =< (v_1, v_2, t_1), (v_2, v_3, t_2), ..., (v_{k-1}, v_k, t_{k-1}) >$, where

- $(v_i, v_{i+1}, t_{init}, t_{end}) \in E$ is the i-th temporal edge on $P_{u,v}$,
- $1 \leq i \leq k$,

– $t_i + R + T \leq t_{i+1}$,

– $t_{init} \leq t_i \leq t_{end}$,

– $n \leq t_1$ and $t_{k-1} \leq N$,

– $u = v_1$ and $v = v_k$.

The *temporal length* or *duration* $d_{P_{u,v}}$ of a temporal path $P_{u,v}$, is the number of snapshots from t_1 to t_{k-1}, i.e., $d_{P_{u,v}} = t_{k-1} - t_1$. Finally, the *temporal shortest-path* or *fastest path* is the minimum temporal length between two nodes u and v, defined as $min(d_{P_{u,v}})$.

Example 5.1.1 *Considering the temporal network of Figure 10 and the parameters* $W = [1, 9]$, $T = 0$ *and* $R = 1$, *we can cite some examples of temporal paths:*

Table 5 – Examples of temporal paths

Temporal Path	Duration	Is fastest path?
$P_{A,D} =< (A, B, 1), (B, D, 2) >$	1	yes
$P_{A,D} =< (A, B, 2), (B, D, 6) >$	4	no
$P_{B,C} =< (B, D, 1), (D, C, 4) >$	3	no
$P_{A,C} =< (A, C, 2) >$	0	yes

5.1.2 Temporal Centralities

Once defined temporal paths concepts, we revisit two important centrality metrics: *closeness* and *betweenness*. As nodes represent users in our context, we chose these local metrics to conduct our analysis in this chapter.

Remembering from Chapter 2.3, the *closeness centrality* of a node is used to measure how close it is from the others nodes in the graph. For example, people with high closeness in a social network are in an excellent position to monitor the information flow – they have the best visibility into what is happening in the network. The closeness centrality of a node v in G is defined as:

$$closeness(v) = \sum_{u \in V \setminus \{v\}} \frac{1}{min(d_{P_{v,u}})} \tag{9}$$

where $min(d_{P_{v,u}})$ is the *duration* of the *fastest-path* from v to u. If there is not any path $P_{v,u}$ then the summation term is 0. If $min(d_{P_{v,u}}) = 0$ (when traversal time $T = 0$), then the summation term is 2.

The *betweenness centrality* of a focal node is the fraction of fastest paths passing through it. In a social network, people with high betweenness have great influence over what flows, and not in the network:

$$betweenness(v) = \sum_{v \neq j \neq k} \frac{w_v(j,k)}{w(j,k)} \tag{10}$$

where $w_v(j,k)$ is the number of fastest paths between j and k that pass through v and $w(j,k)$ is the total amount of fastest paths between j and k.

5.2 Twitter Celebrities Dataset

Although there are many Twitter datasets available in literature (TANG. et al., 2010; VISWANATH et al., 2009; KWAK et al., 2011), we need one containing the information of *when* relationships start and end in the network. Only then we will have a complete temporal network of follower-followee relationships. Moreover, we chose Twitter due to rich information available that allow us to correlate with users behaviors. Datasets like (CATTUTO et al., 2013) are limited on structural information. To the best of our knowledge, there is no available Twitter dataset with temporal network structure.

We developed TCraw[1], our Twitter data collector. The architecture is composed by two crawlers that use Twitter Rest APIs[2] to get data. The first one, Data Crawler is responsible to collect our observation network. This is the name used to refer to nodes and edges that we choose to track. The observation network crawling is done according to the following steps:

1. Choose s users seeds on Twitter. This is the level 0 ($current\_level = 0$).
2. While $current\_level < MAX\_LEVEL$ do
   a) For each node n in $current\_level$, get m followers of n
   b) Set $current\_level = current\_level + 1$

Once stored entire observation network in file format, the Update Crawler is started. Its function is to update structural node information based on a given time interval U. For example, for $U = 24hs$, the temporal network is built with one day granularity. Since structural information is not available in Twitter (Twitter API does not provide historical information about when a user starts/end following other), our dataset only makes sense from the moment Update Crawler is started. So, the observation network built from Data Crawler is our initial state.

[1] Available at: <http://lsi.facom.ufu.br/~fabiola/evolving-networks>
[2] <https://dev.twitter.com/rest/public>

Time-changing Characteristics of Data. Table 6 details statistics of our dataset[3]. It is important to notice that the dataset does not track nodes evolution, just edges activation/inactivation for the 144,975 nodes. The total number of temporal edges has been computed considering that two edges of the type $(u, v, t_{init}, t_{end})$ and $(u, v, t_{init'}, t_{end'})$ are different.

Table 6 – Twitter dataset statistics.

Observation window W	[08/28/2015, 12/15/2015]
Update window (granularity) U	1 day
Max fanout (m)	10,000
`MAX_LEVEL`	1
# nodes	144,975
# edges in first day	837,961
# total temporal edges	1,222,118
Avg # new follows/day	3,492
Avg # unfollows/day	3,657
# seeds (s)	27
# themes	9 (3 seeds each)
Themes related to seeds	politics, sport, news, religion, music, humor, fashion, health, TV

As illustrated in Figures 11 and 12, the dataset is fairly dynamic, specially on October 2015. These observations endorse that the dataset has a time-changing characteristic when considering edges.

Limitations. The strategy we have used for collecting the observation network is a limitation in our dataset. The idea of starting from seeds celebrities resulted in a network extremely unbalanced, with weak connections between users that are not seeds. Another limitation is the difficulty on tracking updates daily in our network. Twitter API has hard restriction policies, which resulted in a relatively small Twitter sample. The longer nodes or edges are considered, the higher should be the granularity (update window).

Even with these limitations, in what follows we show how interesting is the analysis of evolving centralities in our dataset. It is certainly a time-changing network structured data.

5.3 Network Evolution

First of all, in order to compute temporal centralities, in this section we present the baseline algorithm we have implemented able to scan a temporal graph from evolving centralities perspective. Although there are algorithms addressing this evolving centrality problem in literature (WU et al., 2014; TANG et al., 2009; SANTORO et al., 2011) none of

[3] Available at: <http://lsi.facom.ufu.br/~fabiola/evolving-networks>

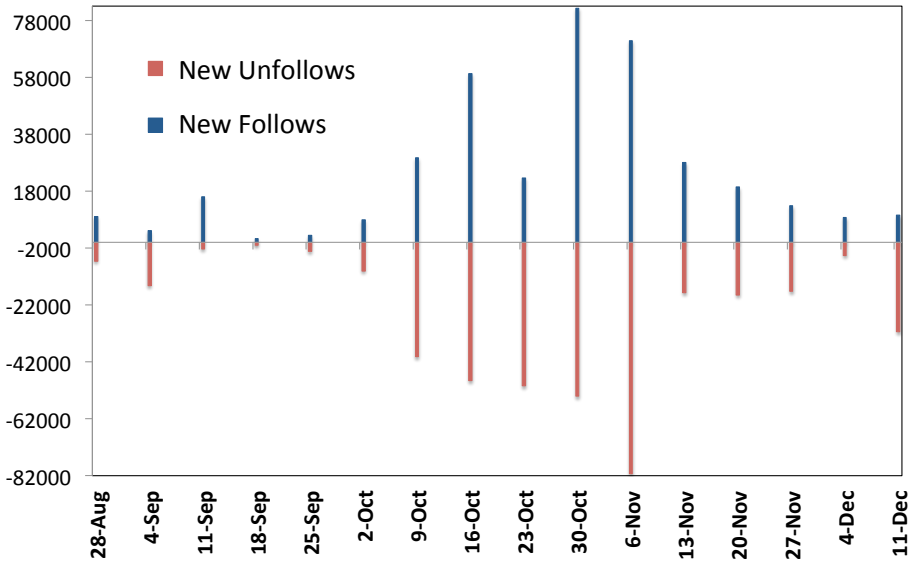

Figure 11 – Number of new follows/unfollows aggregated per week. October is the most activity month.

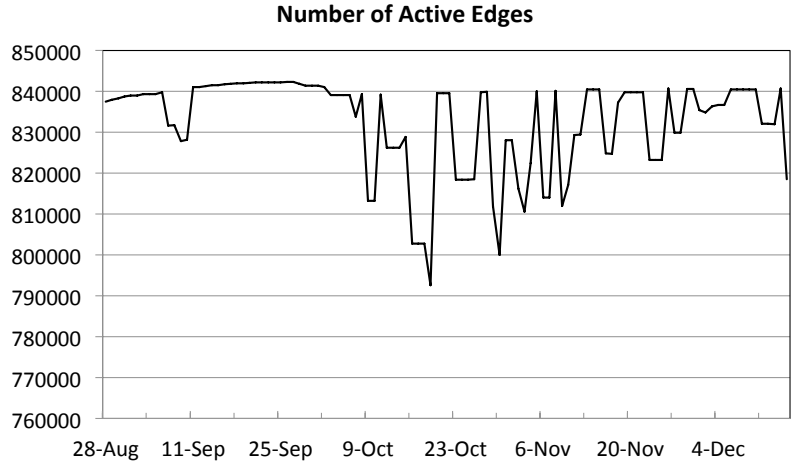

Figure 12 – Number of active edges, i.e., edges present in the given time step.

them addresses exactly our definition of temporal paths (see Section 5.1.1). The idea is to analyze temporal edges as streams.

5.3.1 Stream Representation of an Interval Graph

An interval graph can be represented as data stream. Intuitively, a stream is a sequence of all edges in G that come in order of the time each edge is created. When analyzing interval graphs, the existence interval of an edge $e = (u, v, t_{init}, t_{end})$ means that at any

time inside this interval $[t_{init}, t_{end}]$, there may be a communication between the nodes u and v. For that reason, we first define the stream representation of e as a set of contacts.

Definition 5.3.1 (Contact) *An edge $e = (u, v, t_{init}, t_{end})$ can be represented by a set of contacts $C_e = \{(u, v, t_{init}), (u, v, t_{init+1}), ..., (u, v, t_{end})\}$. A contact $c_{e,t} \in C_e$ is a triple (u, v, t) indicating that there may be communication between u and v at t.*

The data stream representation of an interval graph is a sequence of contacts from all edges in G, ordered by the time of each possible contact. For example, if G has the following edges:

$\{(v_1, v_2, 4, 6), (v_1, v_3, 1, 3), (v_2, v_4, 5, 6), (v_4, v_5, 6, 8)\}$,

the corresponding edge stream is:

$\{(v_1, v_3, 1), (v_1, v_3, 2), (v_1, v_3, 3), (v_1, v_2, 4), (v_1, v_2, 5),$
$(v_2, v_4, 5), (v_1, v_2, 6), (v_2, v_4, 6), (v_4, v_5, 6), (v_4, v_5, 7),$
$(v_4, v_5, 8)\}$.

This definition is different from the one adopted in (WU et al., 2014). In our proposal we convert the edges represented by means of time-varying intervals into set of contacts. The edge stream representation is a sequence of *all possible contacts* from all edges in G ordered by respective contact times. On the other hand, in (WU et al., 2014), the stream is a sequence of all edges in G, ordered according to their *starting time*.

We argue that this conversion is necessary considering the meaning we want to represent: information in Twitter can be diffused as a result of a sequence of contacts among users.

5.3.2 Temporal Centrality Analysis

Before calculating exact closeness or betweenness centrality, we need first to compute all pairs fastest paths, which is too expensive for a large graph. Algorithm 2 is a baseline that we describe for this task. The idea is to process the edge stream and store all temporal paths for each node. In a second phase (from line 14) the algorithm scans all nodes removing paths that are not the fastest ones.

Remarking on complexity analysis of Algorithm 2, we have that for each incoming contact, the computational cost to update the values is $O(V \times P)$, for P being the average number of paths between two nodes. In the end, there is the additional cost of removing cycles and paths that are not the fastest ones, corresponding to $O(V \times P)$ (from line 14). If we consider the total number of elements in the stream as C, we have a final complexity of $O(C \times 2(V \times P))$.

It is important to mention that the algorithm has a high spatial cost. To be able to calculate the exactly (not approximated) values of fastest paths, our algorithm keeps

Algorithm 2 All pairs fastest paths detection

Input: A temporal graph $G = (V, E)$ in its data stream representation, $W = [n, N], R, T$
Output: For each node $v \in V$, a set L_v containing all fastest paths from v

```
1: for each incoming contact c = (u, v, t) do
2:     if t ≥ n and t ≤ N then
3:         // P_{u,v} is a temporal path
4:         P_{u,v} ← {c}
5:         // L_u is a set of all temporal paths from u, initially ∅
6:         L_u ← L_u ∪ {P_{u,v}}
7:         for each x ∈ V do
8:             for each P_{x,z} =< (x, v_1, t_1), ..., (v_k, z, t_k) >∈ L_x do
9:                 if z = u then
10:                    if t ≥ t_k + R + T then
11:                        P_{x,v} ← P_{x,z} ∪ {c}
12:                        L_x ← L_x ∪ {P_{x,v}}
13: // Removing cycle paths and paths that are not the fastest ones
14: for each x ∈ V do
15:     for each P_{x,z} ∈ L_x do
16:         if x = z then
17:             L_x ← L_x − {P_{x,z}}
18:         else
19:             for each P'_{x,z} ∈ L_x do
20:                 if d_{P_{x,z}} > d_{P'_{x,z}} then
21:                     L_x ← L_x − {P_{x,z}}
22: return L_x for each x ∈ V
```

all paths stored while processing the stream. In what follows, we explore the aspects of memory size vs. stream size vs. high velocity processing temporal paths.

5.4 Experimental Analysis

All experiments were performed with 32GB of main memory available. The values for retention and traversal time are $R = 1$ day and $T = 0$, respectively. The Twitter dataset granularity is of 1 day.

5.4.1 Varying Time Intervals

For computing the all pairs fastest paths, the input observation window W can affect several aspects, for cite: overall running time, stream size and duration of fastest paths. We define different intervals to perform the analysis. As Twitter dataset has been collected from Aug 28th, 2015 to Dec 15th, 2015, in Table 7 we summarize the adopted observation windows.

We measure the execution time of Algorithm 2 for different observation windows. Figure 13(a) illustrates these results. The values for WE, FO and MO correspond to the average execution time in each group of windows. We can observe an exponential

Table 7 – Description of observation windows used in experiments.

Period	# of intervals	Values
Weekly (WE)	15	$WE_1 = [09/01, 09/07], WE_2 = [09/08, 09/14], ..., WE_{15} = [12/08, 12/14]$
Fortnightly (FO)	7	$FO_1 = [09/01, 09/15], FO_2 = [09/16, 09/30], ..., FO_7 = [11/30, 12/14]$
Monthly (MO)	3	$MO_1 = [09/01, 09/30], MO_2 = [10/01, 10/31], MO_3 = [11/01, 11/30]$
Total (TO)	1	$TO = [08/28, 12/15]$

behavior compatible with the increasing number of incoming contacts C (Figure 13(b)). Our algorithm is dependent on the size of the observation window.

In Figure 13(c) we show how different the fastest paths values can be just varying observation windows. This endorses the time-varying aspect of Twitter network.

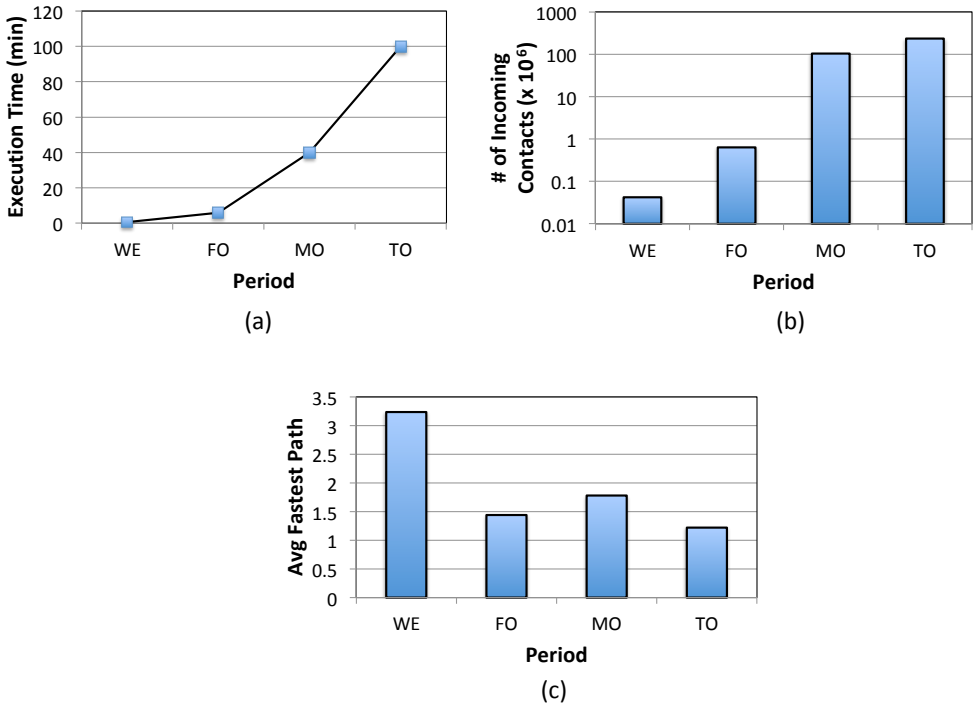

Figure 13 – Results varying the size of observation window when running all pairs fastest paths algorithm. (**a**) The execution time, (**b**) the number of incoming contacts and (**c**) the fastest path duration averaged over all nodes.

5.4.2 Evolving Centralities

The closeness centrality can be easily calculated from the return of Algorithm 2. With all pairs fastest paths and their respective duration, the $closeness(v)$ is a straight sum of these values (see Eq. 9). In Figure 14(a) we can see the closeness value averaged across all users for different observation windows. The observation during short or large amount

of time does not influence on closeness. These values depend mainly of the network behavior: as Twitter network is diversified and extremely dynamic, nodes' closeness vary accordingly.

Another interesting analysis is illustrated in Figure 14(b). Three users u_1, u_2 and u_3 were randomly selected and their closeness analyzed over fortnightly intervals. Remark that these users are not seeds. The graph shows that users are always changing their closeness.

As well as closeness, the betweenness centrality $betweenness(v)$ is a straight calculus from the fastest paths returned by Algorithm 2 (see Eq. 10). Furthermore, the nodes have the same behavior in varying their centrality values. In Figure 14(a) we have the betweenness averaged across all users for different intervals. And in 14(c) the variation for users u_1, u_2 and u_3.

Finally, we rank all nodes according to their temporal betweenness centrality values (first positions for higher centralities). For this analysis, we consider a sequence of incremental observation windows of the type $I_1 = [day1, day15], I_2 = [day16, day30], ..., I_7 = [day91, day105]$. The values in Table 8 suggest that nodes centralities are fairly dynamic and from one observation to the next, the node may have become more or less important.

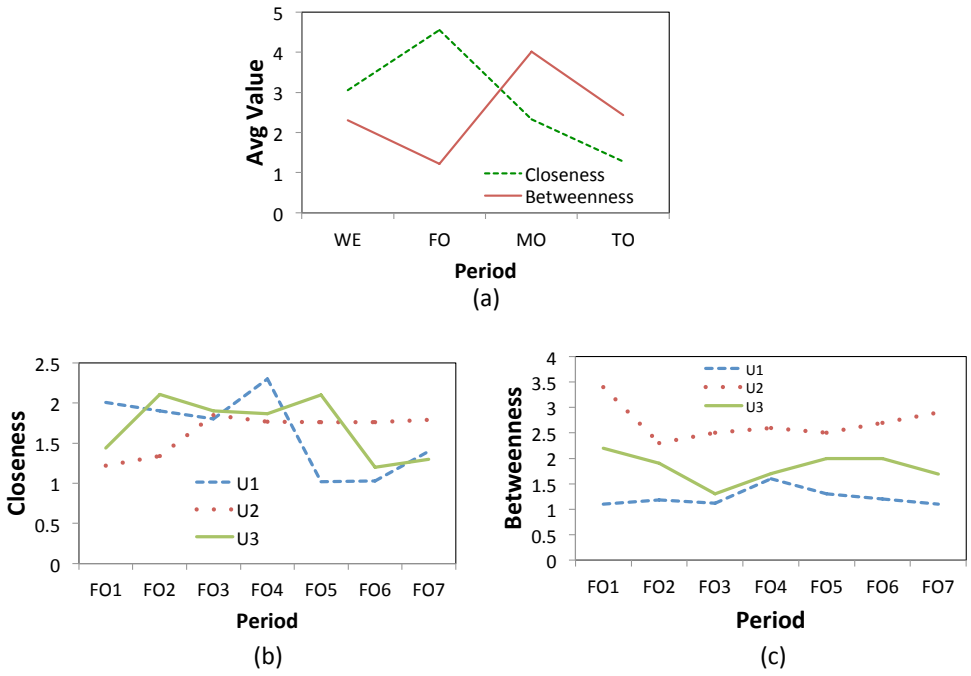

Figure 14 – Evolving centralities observations for (**a**) closeness and betweenness averaged over all nodes, and for (**b**) closeness and (**c**) betweenness of three randomly selected users.

Table 8 – Ranking variation of users U_1, U_2 and U_3 from instants FO_1 to FO_7 considering betweenness centrality.

User	FO_1	FO_2	FO_3	FO_4	FO_5	FO_6	FO_7
U_1	653	644	600	589	592	615	617
U_2	122	123	100	224	220	235	249
U_3	544	530	533	530	544	580	600

5.5 Discussion

The strategy we used in this chapter for computing all pairs fastest paths and then temporal closeness and betweenness centralities is not incremental or real-time processing. Algorithm 2 is extremely sensitive to the size of the stream. Its high velocity processing depends on the size of available memory, which does not characterize the large-scale learning from data streams problem (GAMA, 2010). On the other hand, our proposal is a first look in temporal networks over a real dataset from evolving centralities perspective. Despite of some works investigate the evolving centrality problem, they mainly use approximation and sampling methods (WU et al., 2014; TANG et al., 2009; SANTORO et al., 2011; TABASSUM; GAMA, 2016). To the best of our knowledge, there is not a solution for exactly temporal centrality computing when considering an interval graph, specially for evolving *betweenness*.

5.6 Applications

Analyzing evolving centralities in networks can be applied in diverse real applications. We highlight here two of them. First, the problem of contagion (information, influence and disease) where the interest is in observing not exactly people getting infected, but who infected them (YANG; LESKOVEC, 2010). As contagion is a dynamic process, temporal networks can be applied in this problem and tracking evolving centralities can help in increasing the sales in marketing field, for example.

A second emerging application is on the analysis of user preferences and behaviors (LI et al., 2014). The analysis of evolving centralities can reveal patterns of influence and communications in social networks. For instance, these patterns help to understand how users' preferences evolve over time for more accurate recommendation systems.

5.7 Final Considerations

Our purpose in this chapter was to analyze Twitter from evolving network structure viewpoint. We have modeled Twitter as a temporal network and revisited the concept of shortest path considering the time dimension. We have shown how to compute closeness and betweenness centralities using fastest paths through an algorithm based on all pairs

fastest paths detection. The analysis has been performed over Twitter follower/followee network and our findings have shown that analyzing Twitter as a temporal graph models the behavior of real applications and is different from just considering static analysis.

Chapter 6

Event Detection in Evolving Networks

Nodes structural behavioral dynamics are non-stationary, that is, they change or fluctuate over time. For instance, the structure induced by emails for a given user may change during the work hours. Perhaps this user serves as a coordinator at work and therefore during the day her email activity represents structural behaviors such as the center of a star (node with large number of incoming or outgoing edges) or a bridge that connects multiple communities (or departments in this case).

In this chapter, we explore the idea of event detection in evolving networks. Track evolution analysis is closely related to the problem of outlier detection in temporal networks because temporal outliers are often defined as (abrupt) change points. Our contributions are two-fold. In a first step, we discuss a novel idea of detecting node events based on centrality measures. Then, we address the problem of processing the evolving network in order to detect these events. We glimpse the need of an online algorithm, efficient in terms of online maintenance. Ideas and empirical analysis here presented were published in (PEREIRA et al., 2016).

6.1 Motivation

We are proposing to spot change-points in an evolving network at which one node deviate from its normal behavior. In literature there are many algorithms exploring text-based events in social streams (AGGARWAL; SUBBIAN, 2012; CORDEIRO; GAMA, 2016) or event-detection in the whole network (AKOGLU et al., 2015; AKOGLU; FALOUTSOS, 2010). Thus, our novelty is on focusing in the node behavior: at what points in time a node in an evolving network change its behavior significantly? These are events that reflect some user change behavior. We enumerate some real world examples.

1. **Twitter follower/followee network.** A detected event for user u can indicate that u became a celebrity (increasing number of followers). Or, we can detect that

u now is interested in data streams research field (new representative followees in the field).

2. **Twitter interaction network.** This network consists on representing mentions and retweets among users. In this context, we can detect that u is in the mood for politics (heavily tweeting/retweeting about politics), or that u does not care anymore about religion discussions or even that u earned some influential position related to her career (and got a lot of mentions and retweets).

3. **Facebook friendship network.** This scenario is similar to the Twitter follower/followee network. We could detect that the user u started a new graduation course, so that new friendships are course colleagues.

4. **Email network.** A node event here can represent that u is responsible for some bug in a critical system or that u got a promotion.

5. **Call/SMS graphs.** u's birthday is the most intuitive node event able to be detected. Another examples are u's new baby, u's graduation etc.

Thus, detecting a node event is the action of detecting some remarkable fact or occurrence in someone's life.

6.2 Node Event Mining: The Model

Detecting a node event means to detect the moment that this node played a different role in the network. A node role is directly related to its centrality measures. For instance, a role of influence in the whole network, a bridge role or a role of influence over some community. The idea of our model is to process the evolving network as a stream. Given a target node, as stream evolves, we fire alarms at instant times that node events are detected. In order to detect node events, we propose 2 different strategies, all of them founded on centrality metrics analysis. In what follows, we first describe these strategies and then formally define our *node event mining* problem.

6.2.1 Detecting Nodes Change-points

At what point in time a node in an evolving network change its behavior significantly? Our proposal is based on change-points detection. We introduce *change-point scoring functions* which take values between 0 and 1 where a higher value indicates a change-point. For all functions, we denote $C_t^m(v)$ the centrality metric m of a node v at time t, for m being any centrality measure like closeness, betweenness, degree, Katz, PageRank etc (ZAFARANI et al., 2014). We also consider a window W containing past summarized

centrality values. In Section 6.3 we will discuss more details about processing the evolving network as well as window strategy.

6.2.1.1 Average Score

Given a node v, we compare the current centrality value $C_t^m(v)$ with the arithmetic mean of the $|W|$ past centrality values inside the window W. Formally, we define the average score as

$$\Gamma_t(v) = \frac{|C_{past}^m(v) - C_t^m(v)|}{max(C_{past}^m(v), C_t^m(v))} \tag{11}$$

where $C_{past}^m(v)$ is the average of previous centrality values stored in W, defined as

$$C_{past}^m(v) = avg(C_{t-|W|}^m(v), ..., C_{t-1}^m(v)) \tag{12}$$

The denominator factor from Eq. 11 is responsible to normalize the average score. This is a baseline score very common in non-stationary analysis. Detecting events with the average score simply means that node v changed its role in the network in relation to its own previous behavior, but no additional information like the whole network or v's neighbors is considered.

6.2.1.2 Ranking Score

This approach is founded on change-point in rankings (WEI; CARLEY, 2015). The idea is to maintain a ranking R_t containing all the nodes in the network ordered according to their centrality metrics values for each time instant t. Based on the variations of these metrics values and, consequently, ranking positions from recent past (past positions stored in window W) to current time, we detect changes.

Formally, given the current set of nodes V, we consider a ranking R_t containing all nodes in V ranked in descending order according to their centrality values at time t. We define $pos_t(v)$ as the position of node v in R_t, i.e., $C_t^m(u) > C_t^m(v)$ iff $pos_t(u) > pos_t(v)$, for $u, v \in V$. The *ranking score* $\Lambda_t(v)$ is the acceleration of node v in the ranking position from the past to current instant time t:

$$\Lambda_t(v) = \frac{|pos_t(v) - pos_{past}(v)|}{max(pos_t(v), pos_{past}(v))} \tag{13}$$

where $pos_{past}(v)$ is the average of previous positions of v in rankings $R_{t-|W|}, ..., R_{t-1}$.

Here we consider the target node v centrality evolution in relation to the evolution of the others nodes in the network. With this score we can detect changes that are specific for v, evidencing its changing behavior in contrast to the continuous behavior of the remaining nodes. This is important to distinct cases of bursts, for example, where the

whole network is impacted and not necessary we have a specific node change-point. So, the *ranking score* remains stable.

6.2.2 Problem Definition

We proposed two different *change-point scores*, each one more appropriate for a specific scenario. Finally, we are ready to define a node event.

Definition 6.2.1 (Node event) *Given an evolving network $\mathcal{N} = (V, E)$ and a target node $v \in V$, a node event $\varepsilon_t(v)$ for v at time t is said to be occurred if the score for change-point detection is greater than the threshold θ. In other words, we have:*

$$\varepsilon_t(v) = \begin{cases} 1, \Theta_t(v) > \theta \\ 0, otherwise \end{cases} \tag{14}$$

for Θ assuming any of the change-point scores: Γ (average) or Λ (ranking). For the average and ranking scores θ is usually defined by the standard deviation of $C^m_{past}(v)$ and $pos_{past}(v)$, respectively.

Our node event mining task is: given an evolving network $\mathcal{N}$ and a target node v, return all node events $\varepsilon_t(v)$.

6.3 Node Event Detection Algorithm

The most common way of processing evolving networks is assuming them as edges streams. For our node event mining task we propose a sliding window based algorithm that is able to detect node events as the network evolves. The idea is to work with a sliding window strategy based on time instants. Thus, as time goes by, the oldest edge stream objects are forgotten. According to our model, the node event detection is based on centralities functions. Only the most current objects are considered for updating centrality values. Data summarization strategy varies according to the change-point scoring function choice. In what follows, we formally describe this process.

Definition 6.3.1 (Time domain) *Time domain $\mathbb{T}$ is an ordered, infinite set of discrete time instants $t \in \mathbb{T}$.*

Definition 6.3.2 (Edge stream) *An edge stream is a continuous and temporal sequence of objects $S = E_1...E_r...$, such that each object $E_i = (u, v, t)$ corresponds to an interaction (or a contact) from node u to node v at t, for $t \in \mathbb{T}$.*

We assume that an evolving network $\mathcal{N}$ is an edge stream. Remark that a temporal network (see def. 2.3.1) can naturally be processed as an evolving network (edge stream). For each edge $e = (u, v, t, \delta t) \in E$ and for each time instant t' inside the interval $[t, t + \delta t]$, we consider an edge stream object $E_i = (u, v, t')$. For example, given a temporal network with the following edges: $\{(v_1, v_2, 4, 2), (v_1, v_3, 1, 2), (v_2, v_4, 5, 1), (v_4, v_5, 6, 2)\}$, the corresponding edge stream is: $\{(v_1, v_3, 1), (v_1, v_3, 2), (v_1, v_3, 3), (v_1, v_2, 4), (v_1, v_2, 5), (v_2, v_4, 5), (v_1, v_2, 6), (v_2, v_4, 6), (v_4, v_5, 6), (v_4, v_5, 7), (v_4, v_5, 8)\}$.

The Algorithm 3 is a sketch for processing the *node event mining* task.

Algorithm 3 Node Event Detection

Input: Edge stream $E_1 ... E_r ...$, target node v
Output: Detected events for v

$V \leftarrow \emptyset, E \leftarrow \emptyset, \mathcal{N} = (V, E)$
$t_{current} \leftarrow t$ //t is the current time instant
for each incoming edge stream object $E_i = (u, z, t)$ **do**
    $E \leftarrow E \cup \{E_i\}$
    $V \leftarrow V \cup \{u, z\}$
    update $C_t^m(v)$ or $pos_t(v)$ according to the change-point scoring function
    compute summary values for v at t
    if $t > t_{current}$ **then**
        $t_{current} \leftarrow t$
        if $\varepsilon_t(v) = 1$ **then**
            raise node event detected alarm
        slides W
        $E \leftarrow E - \{(a, b, t') | t' < t - |W|\}$ //refresh $\mathcal{N}$
        refresh summary values
return a binary vector signaling all alarms detected for v

6.3.1 The Window Strategy

We adopted a sliding time-based window of temporal extent $|W|$ and progression step of 1 time instant $t \in \mathbb{T}$. According to our definition, for the same discrete time instant t the edge stream can have many edge stream objects. For example, in a Twitter interaction network, considering 1-day time instants, we can receive several edge stream objects per day.

This window strategy is a good choice because allows detecting node events (i) without much processing effort, (ii) taking advantage of scoring functions semantics and (iii) in an up-to-date way as network evolves.

The window slides over two structures: edge stream objects and summary values. The stream objects are nothing more than the network evolving over time. Thus, having a sliding window over such objects means that centrality metrics used for event detection will always be calculated on an upgraded network, where old edges are discarded. In the same way, values summarized in memory during stream processing are being forgotten as

they become older and leave the window cover. As we will present, the summarization is then done in function of time instants.

6.3.2 Computing Centrality Values

In line 6 we have to update node centralities values in function of new incoming edge. This is our most important and costly task. How to online compute centralities in an evolving network?

Some proposals recently arose in this direction (KAS et al., 2013b; KAS et al., 2013a; LEE et al., 2016). The goal is to effectively update betweenness and closeness centralities, respectively, of nodes in dynamic social networks while avoiding re-computations by exploiting information from earlier computations. However these approaches are not based on online strategies. In this thesis we do not advance in this direction, leaving as future work the proposal of online algorithms for closeness and betweenness processing. We follow the greedy strategy described on previous Chapter 5 when computing temporal centralities. Thus, line 6 has been addressed as a call to Algorithm 2 and further trivial strategies required to update $C_t^m(v)$ or $pos_t(v)$.

6.3.3 Summarizing Values

Each change-point scoring function requires different statistics summarized in memory. But the idea is the same: maintain for each node $|W|+1$ values according to the scoring function. For average score we maintain centralities values C_t^m and $|W|$ past values and for ranking score ranking positions *pos*. In this way, line 7 calls a computation referent to current values (at t) and line 14 refreshes values by forgetting old statistics out of the sliding window and computing average *past* values.

6.4 The Evolving Network

Taking into account these new proposals for node event detection task, we now present some experiments we have performed in order to smooth out the rough edges and put in practice our novel ideas. First we describe the interaction Twitter dataset used in experiments. It is a different dataset from that used in previous chapter precisely to overcome its limitations (see Section 5.2).

6.4.1 Dataset

Folha de São Paulo (or Folha, for short) is one of the most influential newspapers in Brazil. Taking advantage of the fact that Twitter is widespread in the country, we performed our analysis over the news domain in Twitter social network. We collected a

large body of tweets from Folha over the course of 3 weeks, starting in June 24, 2016. Our data collection strategy was as follows.

First, we used Twitter's streaming API to collect all tweets related to the newspaper (user @folha). Thus, our dataset consist of tweets about the news tweeted by Folha newspaper, the retweets and all inherent information mentioning these news. Next, we built the following interaction network: nodes are Twitter users. One edge from user u_1 to u_2 means that u_2 retweed at t some text originally posted by u_1, i.e. edges represent the information flow. The edges are temporal and just exist during the interaction (time t), then they disappear. Figure 15 illustrates the evolving aspect of our network. Colored edges represent topics that are being discussed in the network (in Chapter 7 we give more details about this topic modeling strategy).

In all, we collected 200,806 tweets, 78,944 nodes (users) and 108,133 distinct edges considering the 3 weeks of observation period. An important characteristic of our network is that it has a low average path length. This is a consequence of the fact that in Twitter a retweet always comes from the original post, not mattering from where the user read that post – from the user who originally posted it or from an intermediate user who already retweeted it. On average, the path length is 1.033.

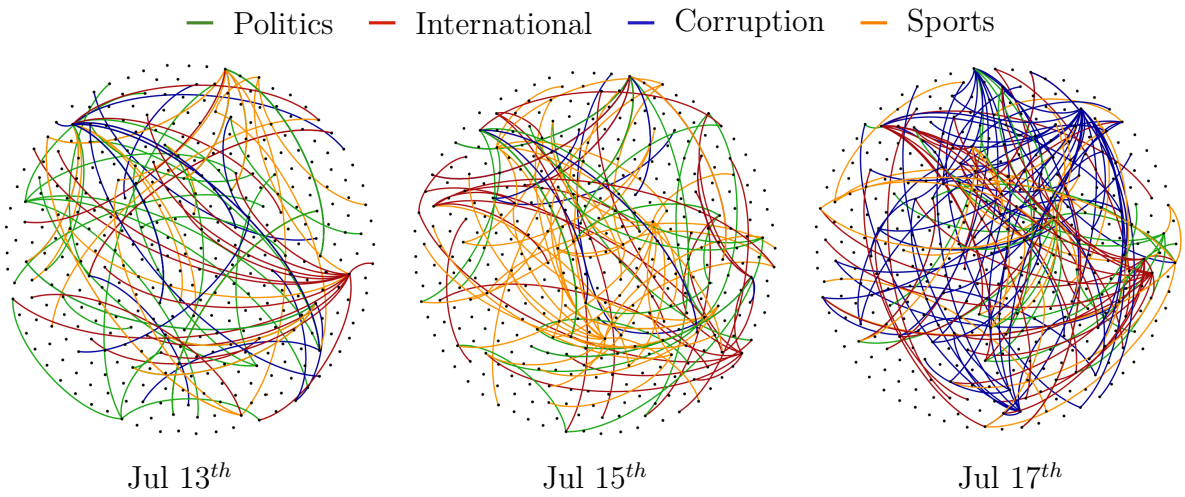

Figure 15 – Snapshots of samples of the evolving interaction network (Jul 13^{th}, Jul 15^{th}, Jul 17^{th}). Nodes are Twitter users. One tie from user u_1 to u_2 means that u_2 retweed at t some text originally posted by u_1. The colors represent topics that users are talking about at t. The samples were built by filtering nodes with degree between 50-22000 and edges representing the 4 most popular topics. Each snapshot corresponds to 1 day time-interval. This figure highlights the *edges* evolving aspect. Nodes are not evolving for better visualization.

6.4.2 Network Semantics

Let us illustrate the semantics of our dataset considering the closeness centrality measure (ZAFARANI et al., 2014). Closeness is related to the visibility of a node in the

network. It is the capacity of a node to reach others in a fast way. Thus, a high closeness value means a good information spreading capacity.

In our context, we identify three types of user: *consumers*, *producers* and *consumers &producers*. *Consumers* are the users who most often just retweet, not publishing any new content. Generally, they have low closeness values. *Producers* are always publishing popular tweets and have a medium closeness value. Finally, the *consumers&producers* have a high activity in the network, tweeting and retweeting all the time. These users have the highest closeness values.

As example, let us consider the scenario illustrated in Figure 16. Events can occur with any type of user, meaning that their usual role changed at that moment. User 3 has a typical *consumer* behavior until time t_6. Just retweeting or even with no activity in the network. From time t_7 user 3 presents a different behavior, which can be a persistent change or an ephemeral behavior. Thus, an event occurred around t_7 and t_8. User 1 is clearly a *producer* from t_1 to t_3. And users 2 and 4 are *consumers&producers* during the whole observation period.

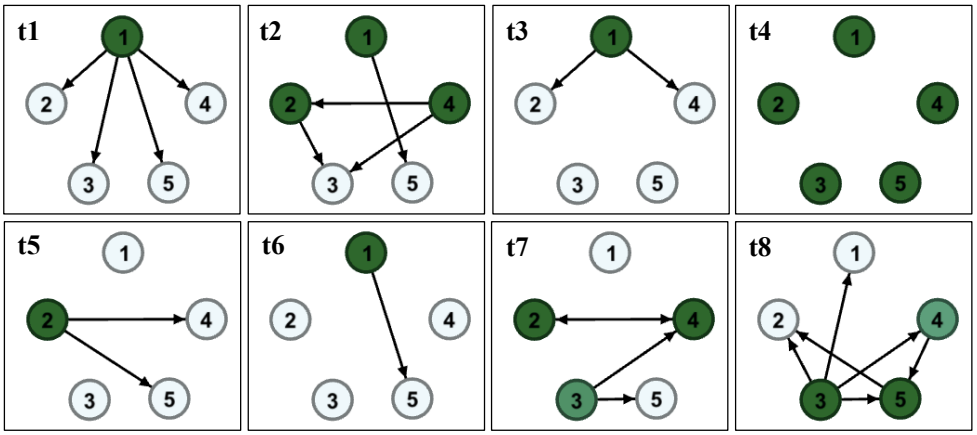

Figure 16 – Example of evolving behavior with closeness centrality. The darker nodes, the greater the centrality.

6.5 Empirical Analysis: Detecting Events with Closeness Centrality

We analyzed the evolving behavior of each node in the network considering closeness centrality measure. In Figure 17 we show the centrality evolving behavior for three different users, one of each type (*consumer*, *producer* and *consumer&producer*). It is possible to distinguish that, generally, the types of users are related with their closeness centrality.

6.5.1 Influence of Parameters Setting

The balance between window size $|W|$ and threshold θ determines what we call of *nature of the event* that we are detecting.

θ adjusts the intensity of the events, varying from smooth to drastic events. In Figure 18 we present an analysis for user $u4$, with θ assuming $0.1, 0.2$ and 0.5 and $|W|= 4$. We chose $u4$ due to its high activity level in the network and $|W|= 4$ as an intermediate value according to our observation period.

As expected, when considering smooth variations more events were detected. Drastic events indicate that the user changed drastically his role in the network. Around day 15 the events indicate that $u4$ leaves a central position as a *consumer&producer* to assume a *consumer* role.

Now, analyzing the impact of the window size $|W|$, in Figure 19 there are the events detected for $|W|$ assuming 2, 4 and 10, $\theta = 0.2$ and user $u4$. Varying $|W|$ means that we are considering the recent past for low values (short-term events) or a big historic for high values (long-term events). As our dataset is relatively short, in these experiments the window size variation did not result in interesting findings. Short-term events are not interesting in our context due to the high variation of centrality values in the network. Considering a mid-term period ($|W|= 4$) reflected better the evolving user role.

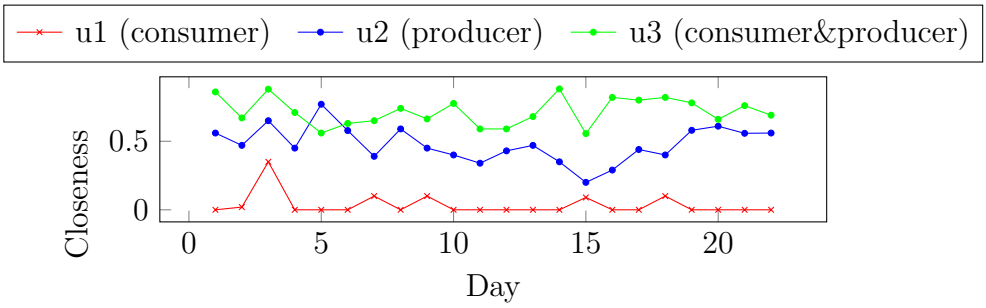

Figure 17 – Closeness evolving for three different types of user.

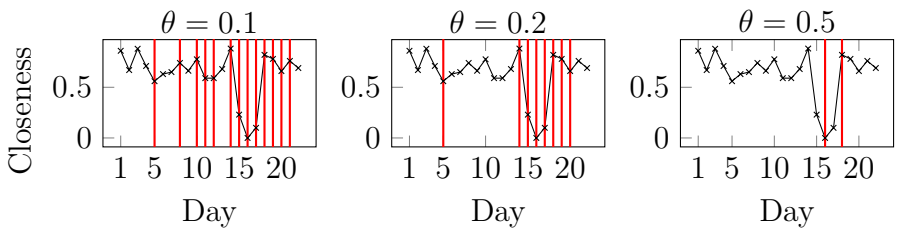

Figure 18 – Impact of θ (intensity of the events) for a high activity user u_4. Detected events are highlighted in red lines.

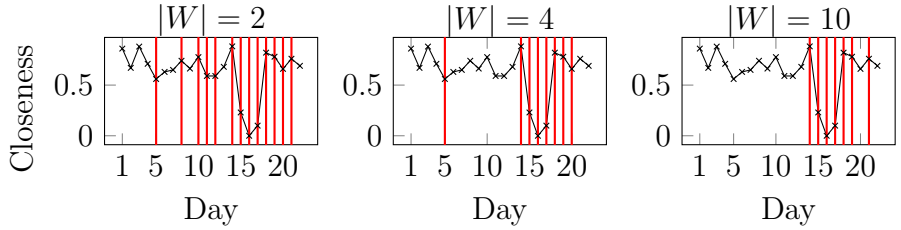

Figure 19 – Impact of $|W|$ (window size) for a high activity user u_4. Detected events are highlighted in red lines.

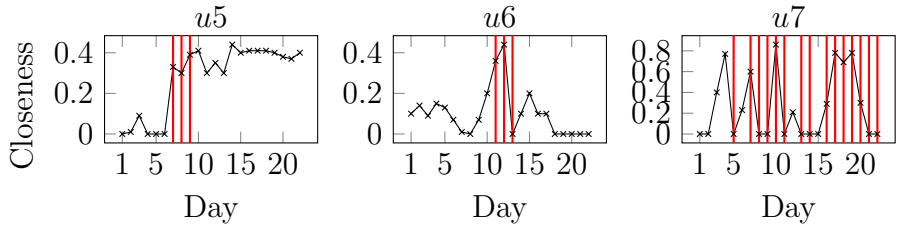

Figure 20 – Detected events highlighted in red lines for three different users (u_5, u_6, u_7). $\theta = 0.2$ and $|W| = 4$.

6.5.2 Detected Events Analysis

From the previous analysis, we consider $\theta = 0.2$ and $|W| = 4$ as the default values. Thus, here we are interested in smooth mid-term events. In Figure 20 we present the events detected for three users $u5$, $u6$ and $u7$. For users $u5$ and $u6$ it is possible to distinguish that the sequence of detected events reflects the moment they change their roles. In case of user $u5$ the change is permanent and for $u6$ the change is just ephemeral. However, in the case of $u7$, a lot of events were detected sequentially, reflecting a behavior of intermittent activities in the network. In fact, a weak point in our event detection method is this: just observing the events, we are not able to distinguish if the events are persistent, if they reflect a burst in the network or if they are ephemeral.

6.6 Final Considerations

We have discussed about node event detection in evolving networks and how this task can be used in our main UPD problem. We have proposed centrality-based strategies for node event detection. A sketch of how detecting these events incrementally in evolving networks has also been presented.

We have performed an empirical analysis considering a more dynamic dataset than the one used in previous chapter. The aim of experiments was to get some feeling about the dataset and the node event detection task. In the next Chapter 7 we will deeply explore through experiments, the proposed node event mining model, both in terms of different strategies for calculating change-point scores, and in terms of a more robust dataset.

Chapter 7

Correlating Changes on User Preferences and Node Centrality in Evolving Networks

After defining our problem and proposing innovative ideas followed by simple experiments to verify these ideas, we are prepared to move now one more step towards validating our hypothesis H1 and H2 stated in the beginning of this thesis (see Chapter 1).

Here we come to the task of correlating changes on user preferences and node events in evolving networks. Our focus is to present a complete analysis comparing the behavior of static versus temporal social networks (Chapter 5) in this correlation context. In order to compute preference changes, we consider the strategy based on consistency formalized in Chapter 4. For node events detection we consider the different change-points scores discussed on Chapter 6. The results presented here do not consider online centralities computing as well as done in previous chapter. Finally, the contributions presented in this chapter are published in (PEREIRA et al., 2016b; PEREIRA, 2017; PEREIRA et al., 2018).

7.1 Dataset

In order to correlate user preferences changes and node events in evolving social networks we need a dataset (1) containing the information of when links begin and end in the network (temporal network topology) and (2) some semantic information from which is possible to extract users' preferences (network content).

We used the same strategy described in Chapter 6, but crawled a new bigger sample of the Twitter Brazilian news dataset. In short, we built a Twitter interaction network, where nodes are Twitter users and an edge (u_1, u_2, t) represents that u_2 retweeted at t

some text originally posted by u_1[1]. As network links are retweets, we assume that contacts are instantaneous considering a minimum time granularity of 1 day and thus the edges duration is not taken into account. In all, we collected 1,771,435 tweets, 150,822 of which were retweeted at least once. Table 9 gives some basic statistics of the crawled network[2] that was used for studying UPD and node events.

Table 9 – Summary of networks statistics

	Twitter network	Jam network
Domain	Brazilian news in Twitter	social music network
Time span	08/08/2016 - 11/09/2016	08/26/2011 - 09/26/2015
# nodes	292,310	54,393
# temporal edges	1,392,841 (retweets)	1,667,335 (likes)
Avg static path length	12.31	7.63
Avg temporal path length	5 (day granularity)	2 (week granularity)

7.2 Extracting Preferences

Probabilistic topic models such as LDA have been applied to extract and represent users' profile in different application scenarios, e.g., Web search and recommendation (AGARWAL; CHEN, 2010; LIU, 2015; CHRISTIDIS et al., 2010). In this work we follow this trend to profile users by applying LDA as we do not have explicit preferences elicited in our dataset. Thus, in order to discover what users are talking about on the network we performed topic modeling with the LDA algorithm (BLEI et al., 2003).

Every interaction (or retweet) between two users is associated with a textual content. We treat each such tweet (textual information) as a document, and the aggregation of all users' interactions considering the entire observation period forms a text corpus. Based on this corpus we perform LDA to extract 50 topics such that each document (tweet) is represented by a topic distribution. According to (WALLACH et al., 2009) choosing a larger k for LDA does not significantly affect the quality of the generated topics. The extra topics can be considered noise. However, choosing a small k may not separate the information precisely. Thus, we varied k from 20 to 80 and from empirical observations we selected $k = 50$ topics.

We analyzed the interpretability of the topics and manually assigned a keyword describing each topic. On Table 10 there are some examples of mined topics and their respective assigned keywords. Following this, we manually grouped these 50 keywords into 10 more general topics, as detailed on Table 11. The reason to group topics into more general ones is to provide better interpretability as these final 10 topics are the domain of preferences. Thus, $A = \{politics, international, corruption, sports, security,$

[1] We consider retweets and quote-status that are retweets with comments

[2] Dataset available at <http://www.lsi.facom.ufu.br/~fabiola/evolving-networks>

$education, entertainment, economy, religion, others\}$ is the set of objects in the domain on which we extract user preferences and each tweet is labeled with one object $o \in A$.

Table 10 – Examples of some topics identified by LDA from Twitter data and respective keywords manually assigned to them for better interpretability.

Keyword	Topic (top-5 words)
Olympic Games	rio, olimpiada, brasil, jogos, metro
Lava Jato	moro, lula, cunha, juiz, sergio
USA Elections	trump, hillary, eua, midia, federais
Lower house speaker	cunha, camara, maia, presidente, governo
Odebrecht	via, folha_com, odebrecht, caixa, milhoes

Table 11 – Manually grouping topic keywords into 10 more general topics.

General topic	Keywords describing a topic
Politics	pro-PT day, coup, lower house speaker, Dilma, Marina Silva, elections, Doria, Temer, INSS, PEC, strike
International	Venezuela, USA elections
Corruption	Moro, Lula, Odebrecht, lava jato, triplex, delação
Sports	Olympic Games, Football
Security	violence, policy, popular manifestations
Education	high school, ENEM
Entertainment	youtuber, book, show
Economy	Petrobras, inflation rate
Religion	pope, Universal
Others	press, journalism, curses

To extract pairwise preferences for each user we use the following strategy: if user u tweets (or retweets) about o at time t, then u has more interest in o over the remaining topics in domain at that moment. We also considered a weight $w_t^u(o)$ based on the number of tweets posted at the same time on a particular topic o. In this case, the top posted topic is preferred over others, the second top posted topic is preferred over the remaining ones and so on. Formally, we have: $\Gamma_t^u = \{o \succ_t^u o' \mid w_t^u(o) > w_t^u(o') \text{ and } o, o' \in A\}$. Noteworthy here is that the time t being considered depends on the time granularity in question, which can be of 1 day or 1 month, for instance. Therefore, a user can post many tweets at the same t.

Example 7.2.1 *Let us suppose that John posts 4 times about corruption, 3 times about sports, 2 times about politics and 1 time about international on time* 3*. The temporal preferences of John on* 3 *are:* $\Gamma_3^{John} = \{corruption \succ_3^{John} sports, sports \succ_3^{John} politics, politics \succ_3^{John} international, international \succ_3^{John} security, international \succ_3^{John} education, international \succ_3^{John} entertainment, international \succ_3^{John} economy, international \succ_3^{John} religion, international \succ_3^{John} others\}$*, besides those temporal preferences obtained from transitive closure of* Γ_3^{John} *omitted for better presentation.*

Figure 21 illustrates samples of the evolving network. As we presented in Chapter 3 there are different strategies to extract user preferences from social networks. We chose the use of topic modeling in order to handle network content and then correlate the evolution patterns of these preferences with evolution patterns of centrality metrics. In (PEREIRA et al., 2016b) we used a different technique to extract preferences mostly based on network topology (number of followers/followees). By considering topics, we improve the impact of our findings as extracting preferences from topics is based on network content.

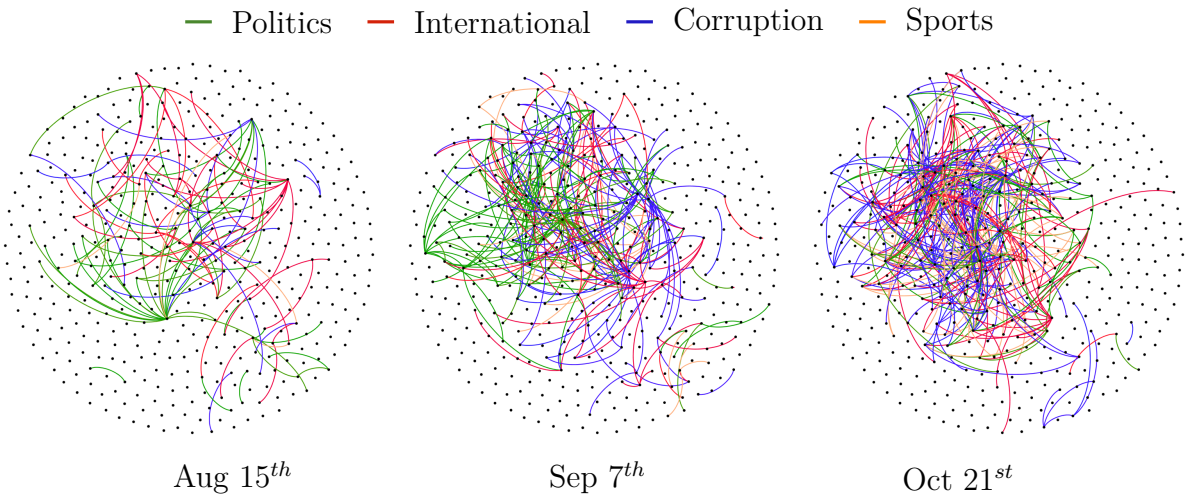

Figure 21 – Snapshots of samples of the evolving interaction network (Aug 15^{th}, Sep 7^{th}, Oct 21^{st}). Nodes are Twitter users. One tie from user u_1 to u_2 means that u_2 retweed at t some text originally posted by u_1. The colors represent topics that users are talking about at t. The samples were built by filtering nodes with degree between 50-22000 and edges representing the 4 most popular topics. Each snapshot corresponds to 1 day time-interval. This figure highlights the *edges* evolving aspect. Nodes are not evolving for better visualization.

7.2.1 This Is My Jam Dataset

This Is My Jam (TIMJ) was an online social music network where users could share their favorite songs with their followers. Only one song could be shared at a time – the current *jam*, which lasted for up to one week in users' status. Furthermore, as a social network, users could like each other's jam. TIMJ dataset was released by Jansson et al. (2015). We built a temporal network based on users' likes, where nodes are Jam users and an edge (u_1, u_2, t) means that u_2 liked u_1's jam posted at t. In this way, the directed edges represent the music influence flow. Jam network features are summarized in Table 9.

7.2.1.1 Extracting Preferences

User preferences were extracted based on music genres. Originally, the TIMJ dataset does not contain jams genre annotations. (JANSSON et al., 2015) mapped the TIMJ dataset to the Million Song Dataset (MSD) (BERTIN-MAHIEUX et al., 2011) – a million popular collection of music tracks and their metadata. From these music tracks, we considered the ground truth CD2 from (SCHREIBER, 2015) to obtain song-level genre annotations. Only the songs presented in the ground truth were taken into account in our analysis. As result, the final set of preference domain is composed by 15 elements: $A = \{rock, pop, country, electronic, reggae, rnb, metal, jazz, punk, folk, latin, world, rap, blues, newage\}$ and we got 528,787 jams annotated with the underlying genre $o \in A$.

The pairwise preferences for each user are extracted from the current jam genre. If user u posted a jam annotated with genre o at time t, then u clearly prefers o over the remaining genres in the domain at that moment. As in Twitter dataset, we considered the weight $w_t^u(o)$ based on the number of times the same genre appeared in u's status during the time granularity being taken into account.

Example 7.2.2 *As example, let us suppose that Mary posted 3 rock jams and 2 jams of pop on time $t = 15$. The temporal preferences of Mary at t are: $\Gamma_{15}^{Mary} = \{rock \succ_{15}^{Mary} pop, pop \succ_{15}^{Mary} country, pop \succ_{15}^{Mary} electronic, ..., pop \succ_{15}^{Mary} newage\}$, besides those temporal preferences obtained from transitive closure of Γ_{15}^{Mary}.*

7.2.2 Discussion

Though our analysis is limited to the Twitter news and social music domains due to the availability of public datasets, we expect our results to generalize to other items like movies, videos, books, vacation packages, shopping etc., which are fairly susceptible to social influence effects. In both domains, the user preferences were extracted based on the content being shared by the users whereas the temporal networks were built based on the interaction of the users with their friends. Moreover, our proposed method behavior will not be affected if users' preferences are estimated from completely independent external sources, as social networks invariably model users behaviors.

7.3 Experimental Evaluation

The main goal of experiments is to investigate the correlation between preference changes δ (Def. 4.2.2) and node events ε (Def. 6.2.1) on Twitter and Jam temporal networks[3]. All algorithms were implemented in Java language using Gephi API[4] as foun-

[3] Source codes available at <http://www.lsi.facom.ufu.br/~fabiola/evolving-networks>
[4] https://gephi.org/toolkit/ (BASTIAN et al., 2009)

dation. All the experiments run over a server equipped with Intel(R) Xeon(R) CPU @ 2.40GHz on 140GB RAM, twenty cores and Linux Ubuntu operating system.

7.3.1 Experimental Environment

Centrality Metrics. We consider two centrality measures: *betweenness* and *closeness*. These measures have different meanings and our objective is to stress to what extent their evolution correlate with preference changes.

According to Zafarani et al. (2014), considering closeness centrality, the intuition is that the more central nodes are, the more quickly they can reach other nodes. Formally, these nodes should have a small average shortest path length to other nodes. The smaller the average shortest path length, the higher the centrality for the node. The betweenness centrality characterizes how important nodes are in connecting other nodes. For a node v, compute the number of shortest paths between other nodes that pass through v.

Change-point scores and Preference Changes. For node events detection we consider three different scores: the proposed approaches (1) average score Γ and (2) ranking score Λ, and (3) the baseline approach of Akoglu e Faloutsos (2010) which we call Z score. In this baseline approach, authors also propose to spot change-points on a time-varying graph from which many nodes deviate from their common behavior. It is the work more related to ours due to two aspects: (i) the change-point based approach and (ii) the temporal dynamics of the network. The idea is to characterize a node with several features so that it becomes a multi-dimensional point. Z score is computed in function of the dot-product between the current feature-vector $\mathbf{v}$ and a typical feature-behavior $\mathbf{r}$, which is the average of past feature-vectors.

For preference change detection we implement our proposed approach δ described in Chapter 4.

Social Network Modeling. We compare *static networks* with *temporal networks*. The difference is that in the temporal scenario we consider temporal paths (fastest paths, as discussed in Chapter 5) when computing centrality metrics, while in the static scenario we consider shortest paths. In temporal networks, the temporal order is taken into account, while in static networks it is not.

Datasets. We vary the time granularity of the social temporal networks Twitter and Jam. In Jam network, time granularities are month, semester and year. In Twitter network, we consider day, week and month. Thus, in all we have six social networks related with news and music domains.

Window size $|W|$. The solutions we propose for the problem of preference change and node events detection are highly sensitive to the size of the observation window W. We vary the window size with values of 2, 4 and 7 time units. This size is related to the desired

semantics we wish to analyze. If we are interested in tracking short-term events, then short sizes fit better. For instance, preferences over the domains of news or restaurants have a high rate of change. On the other hand, long sizes are more appropriate when the events are not frequent, for example preferences about musics and movies. Twitter-month does not vary for values 4 and 7 because it does not contain more than 3 months. The same occur with jam-year because it is limited to 4 time steps (4 years).

Threshold θ. Adjusts the intensity of node events we are looking for, varying from smooth to drastic events. In our experiments we explore how this intensity impacts on correlations with preference changes. From some observations in our data, we detected that Z score has lower levels of θ in comparison to the other scores. Thus, we consider different ranges according to scores. To setup Z values, we varied from 0.01 to 0.05 in a 0.001 granularity in order to observe the amount of detected events for the default features described above. After, we chose the following values to conduct the remainder of the experiments based on diversity: 0.01, 0.015 and 0.04. For ranking and average, the procedure was the same, varying from 0.1 to 0.5, and the final values are: 0.1, 0.2 and 0.5.

Table 12 summarizes values considered in our experiments.

Table 12 – Experimental environment.

Feature	Variation	Default
Dataset	jam-month, jam-semester, jam-year twitter-day, twitter-week, twitter-month	jam-month twitter-week
$\|W\|$	2, 4, 7	2
θ (Γ and Π)	0.1, 0.2, 0.5	0.2
θ (Z)	0.01, 0.015, 0.04	0.015
Centrality metric	betweenness, closeness	closeness
Change-point score Θ	ranking Λ, average Π, Z	average Π
SN Modeling	static, temporal	temporal

7.3.2 Performance Evaluation

The results in Figure 22 correspond to runtimes of the Algorithm 1 for all datasets and different window sizes $|W|$. According to Algorithm 1 complexity analysis, detecting preference changes costs $O(|T||P|)$, which is related to the number of temporal preference relations P and the time interval T being analyzed – the longer T, the more costly the algorithm will be. In Figure 22 we refer to the runtime accumulated for all users in the datasets. Twitter contains more users than Jam. Twitter-day network contains the largest interval $T = 94$ and jam-year has a low number of users as well as a short time interval $T = 4$. The window size is related to P. As $|W|$ increases, more temporal preference relations can be extracted, impacting on the runtime.

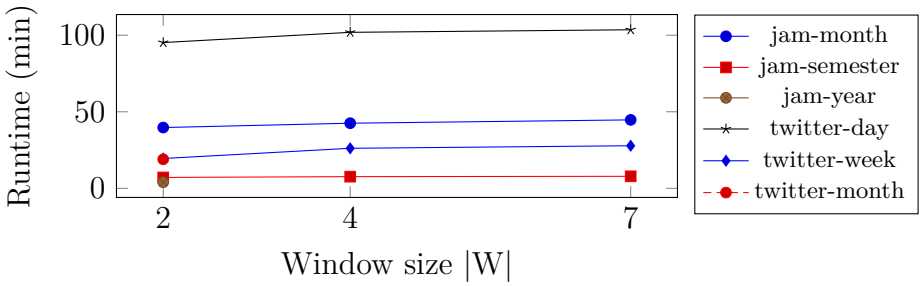

Figure 22 – Performance evaluation of the algorithm *PrefChangeDetection*. Runtimes refer to the time elapsed to process all users of the corresponding dataset.

Remarking on Algorithm 3, the runtimes to detect node events are depicted in Figure 23. For the sake of simplicity, we do not present ranking score runtime information. Ranking and average scores have the same computational complexity. *NodeEventDetection* performance is directly related with network size, which means that the more nodes and edges in a network, more paths between nodes can be detected. In all scenarios, temporal networks are more time-consuming than static networks counterpart. In fact, when considering temporal order, there are more paths than when time is not taken into account. Comparing centrality runtime behavior, we conclude that computing closeness centrality is faster than computing betweenness centrality (BRANDES, 2001). This difference also impacts on the high runtime elapsed by Z score, which covers both centralities.

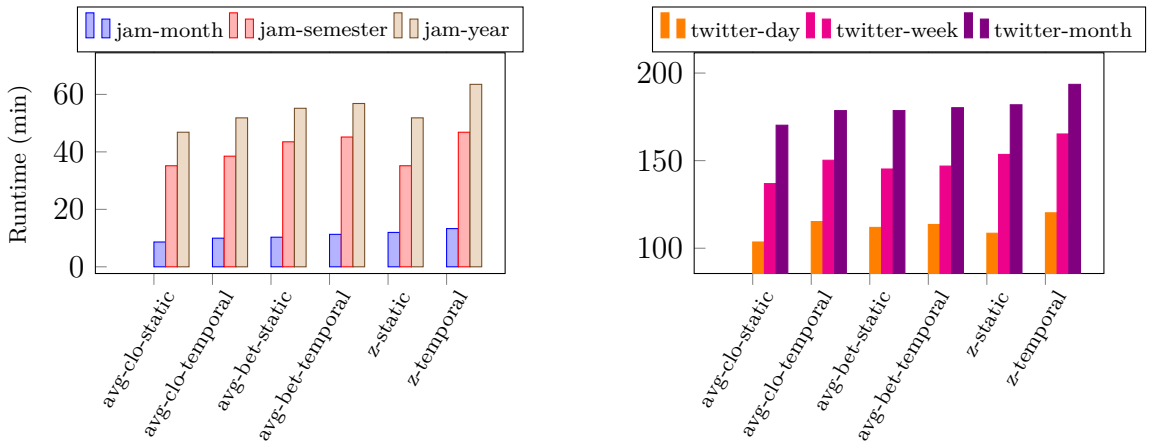

Figure 23 – Performance evaluation of the algorithm *NodeEventDetection*. Runtimes refer to the time elapsed to process all users of the corresponding dataset.

7.3.3 Analyzing Network and Preference Evolution

Taking into account the set of parameters and possible scenarios to stress, we first perform observations taken from both specific nodes/users and the whole network evolving behavior. In the following, we detail important evidences extracted from these observations.

7.3.3.1 Evolving Networks

In the first analysis we compare, quantitatively, all change-points scores averaged over all users for each time step, also varying centrality metrics. Default setup was considered for the remaining features. The results are presented in Figures 24 and 25, for Twitter and Jam networks, respectively. This experiment reflects networks' global behavior. The most important observation is that values for *ranking* and *average* are high in contrast to Z indicating that we should consider different values for θ when detecting events, otherwise *ranking* and *average* scores will detect much more node events than Z, not reflecting the reality. This behavior can be explained by the fact that Z score is more complex and consider a set of centrality measures (closeness and betweenness in this case) to describe a node while *ranking* and *average* are computed with respect to only one centrality measure. Concerning centrality metrics, we vary *average* and *ranking* for closeness and betweenness. Quantitatively, change-point scores values remain in the same range independent to the centrality metric. The number of events detected is different for each centrality metric which is expected, as they have different meanings and thus vary according to different changes in the structure of the network.

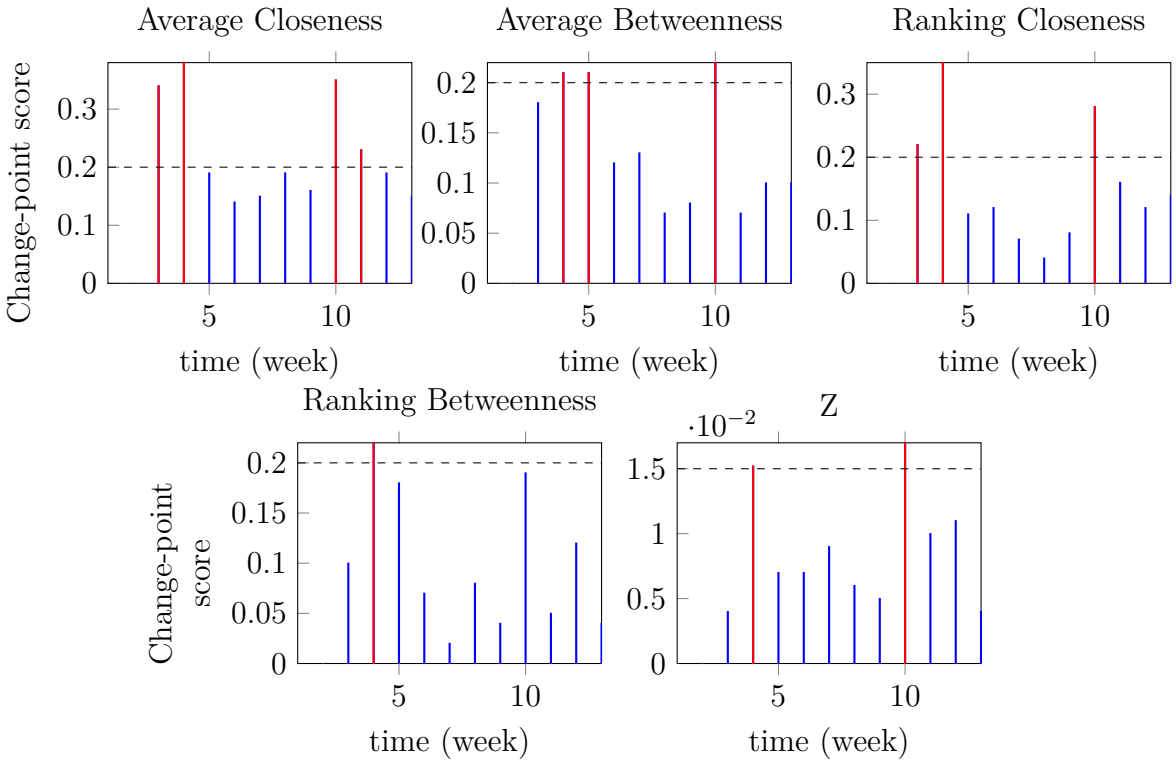

Figure 24 – Change-point scores averaged over all users in Twitter-week network, for $|W|= 2$ and temporal modeling.

Qualitatively speaking, the change-points detected occurred on similar moments for *average*, *ranking* and Z in both datasets. These observations give us confidence in terms of the time instants the events occurred, independent to the centrality metric and the

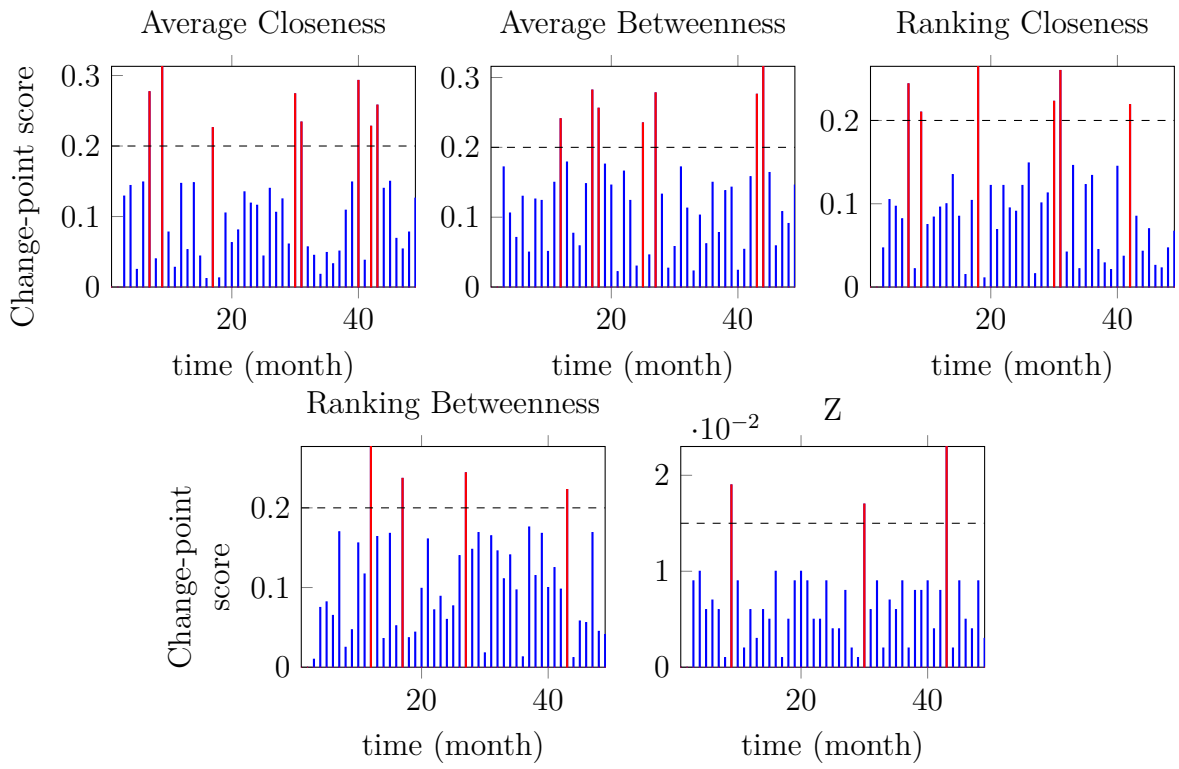

Figure 25 – Change-point scores averaged over all users, for $|W|= 2$ and temporal modeling the Jam-month network.

change-point score strategy. It is an open question to define which change-point score fits better in a given scenario. The difficulty is related to the lack of a ground truth when analyzing social media data as users are scattered all across the globe (ZAFARANI; LIU, 2015). We glimpse that the variety of scenarios we propose for detecting node events can be further stressed and used to define evaluation metrics. Remark that here we just perform an analytical comparison among the events as our focus is on correlating the detected change-points with preference changes, not on defining the highest accuracy for the node event detection task.

7.3.3.2 Preference Dynamics

We analyze how preferences evolve in both networks considering a global perspective. Results are depicted in Figure 26. In twitter-week network the topics *sports, corruption* and *politics* are the most preferred during the whole period. Comparing weeks 3 and 4, the number of users preferring *sports* over the others have increased. The same behavior can be observed for weeks 9 and 10 regarding *politics*, and 12 and 13 for *economy*. These change points occur around the same time instants detected on previous experiments, specially considering average closeness setup (Figure 24).

Jam-month network users mostly prefer *rock* and *pop*. A pattern deviation can be observed on months 9, 30, 33, 34 and 43 when users mostly prefer genres different from

rock and *pop*. Again, we can establish a comparison between these time steps and those detected on Figure 25.

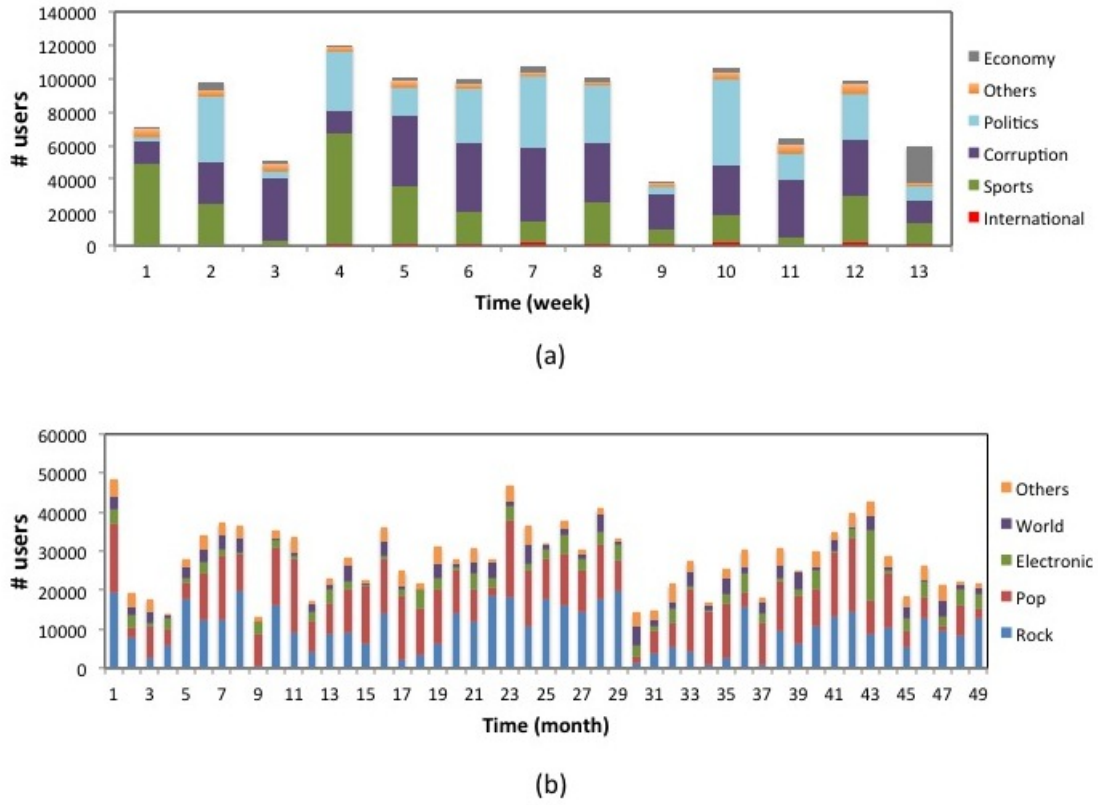

Figure 26 – Most preferred topics over time considering all users in the (a) twitter-week and (b) jam-month networks.

In order to illustrate a local perspective of preference change process, in Figure 27 we show a given user u's better-than graphs (BTGs) in two different moments of twitter-day network (u id = 58488491). On Aug 21^{st} u preferences were *corruption* and *politics* over *sports* and then *sports* over the remaining topics. On Aug 22^{nd}, new preferences $sports \succ^{u}_{Aug22} politics$ and $sports \succ^{u}_{Aug22} corruption$ appeared, causing a preference change event. After the revision, the resulting acyclic BTG represents u preferences on Aug 22^{nd}. Considering that Aug 21^{st} was the end date of Olympic games in Rio de Janeiro, probably u had been influenced by this trending topic on the network.

7.3.3.3 User Preferences and Network Evolution

In the last analysis we observe the relationship between change behaviors considering all nodes/users. Figure 28 depicts comparisons among all scoring strategies in relation to the percentage of nodes that change their behavior from twitter-week and jam-semester temporal networks. The first observation is that for all scores the percentages maintain a pattern with low deviation. This indicates coherency on scoring strategies. We can also observe that Z score detected fewer changes than *average* and *ranking*. Moreover, betweenness centrality detected a higher number of changes than closeness. From these

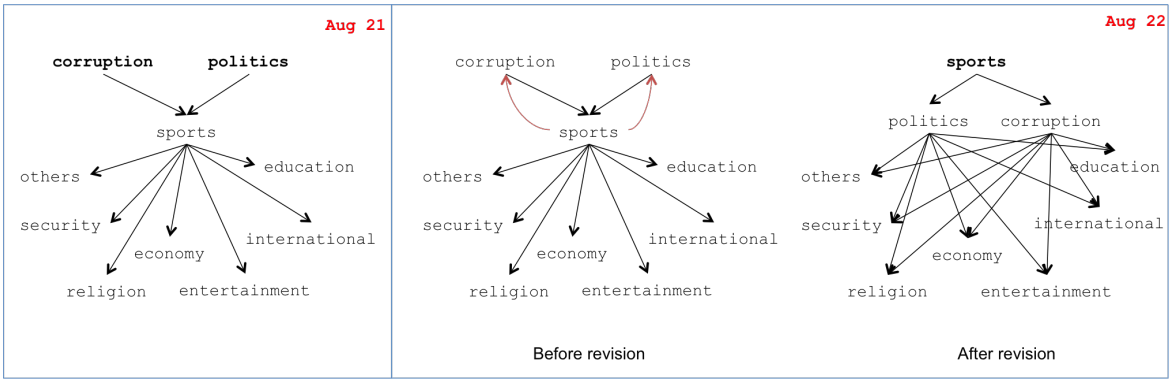

Figure 27 – Better-than graphs representing u_3 preferences on days Aug 21 and Aug 22. Aug 21 was the end date of Olympic Games in Rio de Janeiro.

observations, we were able to ascertain high levels of confidence concerning change-point scores and centrality metrics.

From the preference evolution viewpoint, the percentage of users that change their preferences is very similar to the percentage of nodes change-points previously discussed. On average, 36% and 27% of users changed their preferences on a weekly and semiannually basis, respectively.

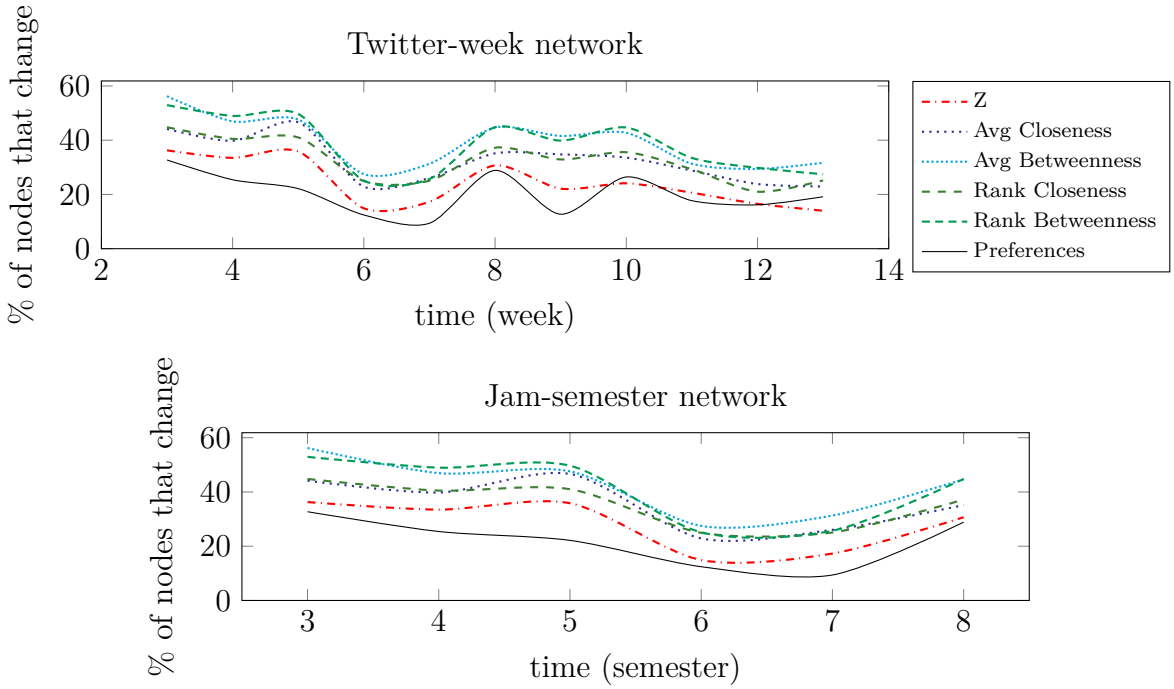

Figure 28 – Percentage of nodes/users that change centrality/preference at each time step in (top) Twitter-week and (bottom) Jam-semester temporal networks.

7.3.4 Relating preference changes and node events

There are many directions to explore from the evidences presented in the previous section: (i) to what extent evolving networks are related with user preference dynamics

(UPD)? (ii) Which centrality metrics should be used in order to analyze UPD? (iii) Which change-point scores should be considered and (iv) what is the best network modeling for analyzing UPD: static or temporal? To address these points we formulated the following research questions:

Q1: Is there a relationship between user preference changes and centrality-based node events in evolving social networks?

We use Pearson Correlation Coefficient (ρ) to evaluate if there is a linear correlation between δ and ε and the strength of this correlation. For each user u of our observation period, we compute $\rho(\delta^u, \varepsilon^u)$ considering a population of the whole observation period (94 twitter-day, 13 twitter-week, 3 twitter-month and 49 jam-month, 8 jam-semester, 4 jam-year). Then we averaged these correlation values $\rho_{avg}(\delta, \varepsilon)$ over all users.

We explore several scenarios for each of the six social networks – *twitter-day, twitter-week, twitter-month, jam-month, jam-semester, jam-year*, in order to stress the time granularity effect. We also vary the parameter θ according to respective scores range (see Table 12). This parameter indicates that the closer to 1 more significant are the centrality changes that are being considered. Then, we vary the window size $|W|$ to explore long-term and short-term impact of the events on the correlation strength of variables. When considering smooth variations (low θ) more events were detected.

Each scenario compares ρ values in relation to betweenness and closeness centralities for average and ranking scores, and in relation to Z score with betweenness and closeness being used to describe a node. Figures 29 and 30 illustrate our results highlighting the comparison between static and temporal networks correlation strengths.

In all scenarios δ and ε associate significantly (as compared to the corresponding critical values – in all scenarios critical values are lower than 0.1). Our null hypothesis H_0 is that there is no linear correlation between δ and ε, i.e. $\rho = 0$. Two random variables (with no correlation) would have a 90% probability of p-value greater than a critical value. We observe a strong correlation between change events in user preferences and in centrality metrics in most scenarios.

ρ values for Jam networks are higher than Twitter networks comparing similar scenarios. This can be explained by the preference extraction strategies and inherent noise. In fact, the Jam preference semantic based on music genres is more accurate than the topic modeling strategy used for preference extraction from Twitter. Moreover, besides users mostly retweet their preferences, they can retweet due to other reasons (METAXAS et al., 2015), while in general users listen what they prefer (MOORE et al., 2013).

Q2: Are temporal networks more suitable than static networks for analyzing user preference dynamics?

Across all scenarios investigated here, the decision of modeling our network with temporal information made difference. The more time instants, the greater the difference of

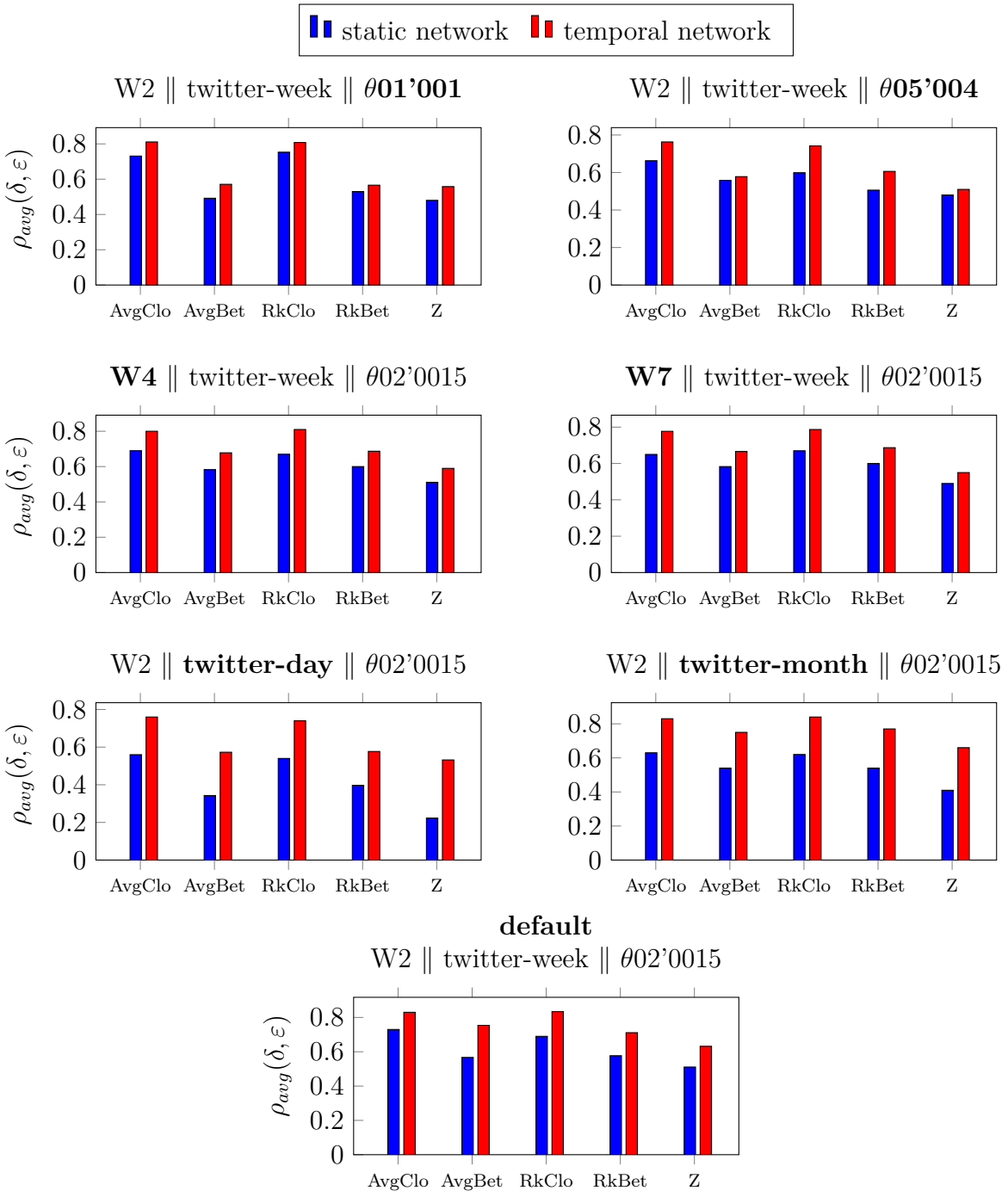

Figure 29 – ρ between preference changes and centrality-based node events for Twitter dataset.

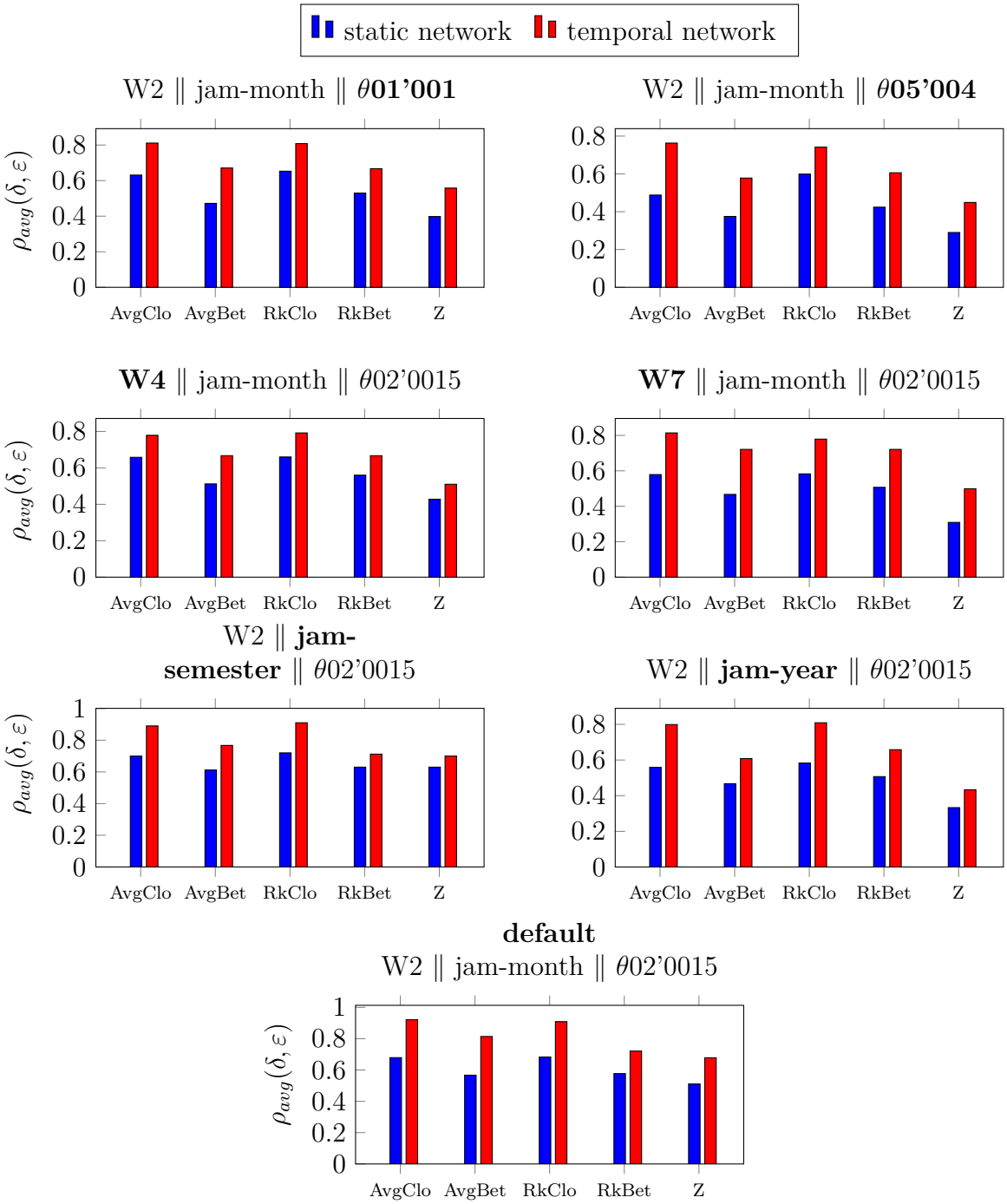

Figure 30 – ρ between preference changes and centrality-based node events for This Is My Jam dataset.

ρ values in temporal networks against static networks. For instance, in Twitter default scenario the higher ρ when using the static network is 0.71 while the same value corresponds to the lower ρ considering the temporal network. Thus, temporal networks are statistically more suitable than static networks for analyzing UPD.

The results obtained so far can be explained by the phenomena of information propagation and inherent consequences of homophily and influence. The main difference between temporal and static networks is that temporal networks take into account the contact sequence between nodes to compute paths (PEREIRA et al., 2016a) and this has an impact on different centrality measures. The related work (GUILLE; HACID, 2012) discusses about relation among preferences and information propagation in social networks. The aspects described in our motivating example (Chapter 1) could illustrate that preferences are directed by information flow on the social network. Finally, temporal networks represent information flow more realistically.

Q3: Considering closeness and betweenness, what change-point score and respective centrality metrics should be used when analyzing UPD?

If we analyze correlation values comparing change-point scores, we find that *average* and *ranking* are more correlated than *Z*. However, there is no consensus in relation to the best change-point score. For instance, considering Jam default settings *average* has stronger correlations than *ranking*, but in Twitter default the behavior is the opposite. Despite *Z* score is induced by structural measures as in *average* and *ranking*, the combination of betweenness and closeness measures to describe a node did not result in stronger correlations than considering them separately. The centralities are conceptually different and not necessarily when one is highly correlated the other will be, decreasing Z score performance.

Now observing centrality metrics we conclude that closeness is more suitable when correlating UPD and node events, considering average and ranking. The closeness centrality measures the inverse total distance to all other nodes and is high for nodes that are close to all others. Similarly, for temporal networks, the idea is to measure how quickly a node may on average reach other nodes. In this work, we hypothesize that user preference dynamics are related to her social network evolution, given the aspect of social influence and, consequently, network structural changes. Thus, we conclude that this relationship of a node quickly reaching others is the most important aspect we should consider when addressing the UPD problem.

7.4 Final Considerations

In this chapter, we have shown experiments demonstrating that *temporal networks* are suitable to analyze user preferences dynamics (UPD). We built two social networks, one based on retweets about Brazilian news and the second based on likes of a music

community, and found that structural changes in networks' topology of a node (user) is directly related with changes on preferences of that user.

User preferences have been extracted by means of topic modeling based on network content. Centrality-based node events have been detected considering changes on structural position of nodes in network topology. By correlating changes on preferences and on centralities we move towards understanding how *content* and *topology* evolve in a network.

Remark that all the results presented are based on snapshots processing, considering or not the temporal order of outcomes. Computing change-points with online algorithms over evolving networks is the approach we will present in the next Chapter 8.

Chapter 8

The Preference Change Prediction Model

Social networks evolve at a fast rate of edge arrival. Due to this characteristic processing and analyzing such networks is challenging. Usually, the most feasible approach to process them is to consider network streams and perform analysis under online approaches and one-pass constraints (FAIRBANKS et al., 2013). When considering such evolving aspect, analysis can be performed from the perspective of changes on the structure of the underlying network.

According to Gama et al. (2014), change detection refers to techniques and mechanisms for explicit drift detection characterized by the identification of change points or small time intervals during which changes occur. In evolving networks context, these changes can be detected observing the whole network, for instance communities (CORDEIRO; GAMA, 2016) and motifis (BIFET et al., 2011) evolution; or the changes can be analyzed in a node-centric way, where nodes centrality and roles are observed during network evolution (PEREIRA et al., 2016). In this work, our focus is on node-centric network evolving analysis.

Semantically, social networks structures model users' relationships and are able to reveal their behaviors and preferences. Considering an evolving environment, all the time users are facing others' opinions and being socially influenced, making their preferences fairly dynamic. This scenario dispatches several research efforts to investigate the interplay between user preferences and social networks (LI et al., 2014; ABBASI et al., 2014).

In this chapter we propose a model for processing evolving network streams to calculate node centric centrality scores and further to detect change points based on these centrality values. Essentially, our model consists in processing edges' stream, calculating nodes centrality and employing aggregating mechanisms for tracking the evolution of the centrality measures for a given user u. We show that this evolution can be used to predict u's preference changes, validating our hypothesis H3 stated in the beginning of this thesis (Chapter 1).

Massive networks are difficult to be stored in memory as a total aggregated network and processing them as aggregate makes no sense with evolution. Our model uses a memory less Page-Hinkley test (PAGE, 1954) and two window based mechanisms with only window size of memory required for centrality measures of corresponding nodes. For updating centralities such as betweenness and closeness we only need to store the network of current time step T as far as for updating degree centrality we do not even need to store any network. These proposals are an evolution of those discussed in Chapter 6, when considering online processing strategies to handle the evolving network. The contributions presented in this chapter are published in (PEREIRA et al., 2019).

8.1 Preference Change Detection

In this section we present our model for processing evolving network streams to calculate node-centric centrality scores and further employing aggregating mechanisms such as moving window average, weighted moving window average and Page-Hinkley (PH) test for tracking the vacillation of preferences by users based on their temporal streams of centrality scores. Additionally, as in previous chapters, we use a change point scoring function and change point detection threshold to quantify and qualify the change points obtained. Some assumptions for the above model such as handling of insertions and deletions are briefed in subsection 8.1.8. Lastly in this section we detail the evaluation methodology for gauging our results.

8.1.1 Processing Streaming Network

For change detection in traditional uni-variate time series data, multiple change points were detected from a vector of elements distributed temporally. We have a similar temporal data, which is networked and evolving that makes it complex. In this work we consider not just one vector of uni-variate time series data but we have such $|V|$ number of multiple streams of uni-variate time series considering the centrality measures of each node in the network. This problem is different to multivariate time series as the centrality score stream of each node is treated independent to each other and the change points are detected per node independently. The only dependence is considered while computing centralities. To elucidate the aforementioned process, we delineate the model below.

Definition 8.1.1 *(Evolving Network Stream) We consider an edge stream S which is a continuous and unbounded flow of objects $E_1, E_2, E_3.....$, where each edge E_i is defined by (v, u, t) which represents a connection between vertices/nodes u and v at time t. The vertices $\{u, v, ...\} \in V$ and get added to or deleted from V at anytime t.*

For every incoming edge (v, u, t) from the above defined stream S, the centrality scores $C^m(v)$ and $C^m(u)$ for nodes u and v are updated in the order of t (in case of degree central-

ity, the degree of nodes u and v is updated for every incoming edge $\{u, v\}$ at t where as in the case of betweenness and closeness, the centrality of nodes are updated only after every T). Another variable T is a discrete time-step/time-interval with granularity defined by the user. In our experiments we considered the granularity of T as 1 day. After every T the centrality scores are reset and starts accumulating again. Therefore, we keep track of centrality score of nodes per day. Consequently we store a set of nodes (with changing cardinality) and a streaming vector of its associated centralities per time step T. As a result, we have an independent non stationary stream of centrality scores $\{C^m_{T_1}, C^m_{T_2}, C^m_{T_3}.....\}$ for every node v in S after every time step T. To get a normalized version of scores, after every time-step T the centrality of a node is divided by the number of nodes in graph at T. Therefore we have normalized centrality scores $\{C^m_{T_1}, C^m_{T_2}, C^m_{T_3}.....\}$ in the vector stream. For notational simplicity in the below equations we use C_T for $C^m_T(v)$ as all notations for the techniques below are considered for a stream of centrality scores per node per centrality metric. Further we employed the smoothing mechanisms below (subsections 8.1.3, 8.1.4 and 8.1.5) to the above streams of centrality scores per node.

As the centrality score stream of every node is independent of each other, parallel implementation of the above aggregating mechanisms per node centrality feature stream is practicable. Though here we employed them sequentially for every node in the graph, as computing mean for $|V|$ number of nodes is not expensive.

8.1.2 Computing Centralities

We have considered three centralities for our experiments: degree, betweenness and closeness, which are explained below. The notion of employing three types of measures is to compare their efficiency for predicting preference changes of a node while considering the trade-offs between efficiency, time and space complexity. The process of calculating centralities is explained below

8.1.2.1 Degree Centrality

Degree centrality of a node is the measure of number of edges adjacent to it. Degree centrality can be computed on a fly for streaming data. As explained above the centrality score of a node is updated for every incoming edge. It is then stored in a queue as it should follow first in first out principle for window based approaches, then the edges are discarded. Degree centrality is space efficient with $O(V_T)$ (where V_T is the number of nodes from the time-step T) as we do not need to store the network. For updating centrality at the arrival of edge, the cost is negligible with $O(V_T)$ as it only needs to increment a counter for degree centrality. For window based approaches, the length of queue storing degree centralities is always equal to the window size W_S and the space

used is constant. Whereas for PH test we only need to store the current degree centrality score.

8.1.2.2 Betweenness Centrality

Number of shortest paths passing through a node is the betweenness centrality measure of that node. For computing this measure, we follow the strategy described in (BRANDES, 2001) which is implemented by Gephi API[1]. Different from degree centrality, betweenness is not computed in a stream fashion. This measure is not updated incrementally for every incoming edge, but the edges/network are stored for each T. After every T the betweenness centrality is batch calculated and current centrality score C_T generated. The edges are then discarded and the process restarts. Betweenness centrality requires $O(V_T + E_T)$ space and run in $O(V_T E_T)$ time on unweighted networks. Where V_T and E_T is the number of nodes and edges from time-step T. Note that in this approach, centrality score is not computed incrementally, but after being generated we maintain the streaming strategy by adding in a queue the centrality score (after every T) for window based approaches (constant used space of size W_S) and maintaining only the current betweenness score for PH test.

8.1.2.3 Closeness Centrality

Closeness centrality is the inverse of the average shortest path length between a node and all the other nodes in the graph. The smaller the average shortest path length, the higher the centrality for the node. Computing closeness centrality follows the same strategy above described for betweenness as it is also a shortest-path based centrality. We need to store incoming edges/network at each T, batch process closeness (BRANDES, 2001) and then discard edges and restart the process. The space complexity is $O(V_T + E_T)$ and requires $O(V_T E_T)$ time. In PH test just current closeness score is maintained and for window based approaches the centrality score is stored, consuming W_S space.

8.1.3 Moving Window Average (MWA)

A window of size W_S consists of data points from the latest temporal time steps $\{T, T-1, T-2, ..., T-(W_S-1)\}$. The window keeps on sliding to always maintain the latest W_S time steps and the data points from $T - W_S$ are forgotten. Alongside, the mean of data points within the window is calculated by using simple equation (15) where C_{T-i} is the stream of centrality scores at time-step $T - i$ using measure m per node. In this

[1] github.com/gephi

approach all the data points in the window are assigned equal weights.

$$\mu_T = \frac{1}{W_S} \sum_{i=0}^{W_S-1} C_{T-i} \tag{15}$$

As the window slides the mean of data points in the window is updated, either using the above equation (15) for small window sizes and equation (16) for large window sizes.

$$\mu_T = \mu_{T-1} W_S - C_{T-W_S} + C_T \tag{16}$$

8.1.4 Weighted Moving Window Average (WMWA)

Weighted moving window average follows the same window sliding strategy as in MWA and computes average over the data points in the window. The improvement over MWA is that the accumulated data points per time step T in the window are weighted linearly as given in equation (17). The oldest data points in the window attain a least weight and the latest data point acquires the highest weight linear to the least one. Weights are updated, when the window slides. Assignment of weights per data point depends on the window size.

$$\mu_T = \sum_{i=0}^{W_S-1} \frac{C_{T-i}(W_S - i)}{W_S - i} \tag{17}$$

8.1.5 Page-Hinkley Test (PH)

Page-Hinkley (PAGE, 1954) is one of the memory less sequential analysis techniques typically used for change detection (MOUSS et al., 2004; GAMA et al., 2013; SEBASTIÃO et al., 2013; GAMA et al., 2014). We use it as a non-parametric test, as the distribution is non stationary and not known. This test considers a cumulative variable m_T , defined as the cummulated difference between the latest centrality score at T and the previous mean till the current moment, as given in the equation (18) below:

$$m_T = \sum_{i=1}^{T} |C_T - \mu_{T-1}| - \alpha \tag{18}$$

Where $\mu_T = 1/|T| \sum_{i=1}^{T} C_i$, $\mu_0 = 0$ and α = magnitude of changes that are allowed. For calculating μ_T we also need to store the number of time-steps passed.

Relative α: The equation (18) given above uses fixed α value, which is not pertinent with our multiple vector streams of centralities per node, where the centrality scores of few active nodes are way higher than some least active nodes. Therefore, using same value of α over differing node centralities would not be fair enough. Hence, we use a relative α, which is relative with the differing centrality scores per node. **Relative** α is a point percentage of previous aggregated mean of that node, as given in equation (19).

Example: Consider a node from stream S with a current centrality score $C_T = 2$ and $\mu_{T-1} = 3$ and fixed α =0.1. Using equation (18) we get m_T as $3 - 2 - 0.1 = 0.9$. Using relative alpha $(0.1 * 3 = 0.3)$ in equation (19) we get m_T as $3 - 2 - 0.3 = 0.7$. Consider another node of high activity from the same stream S with $C_T = 60$ and $\mu_{T-1} = 50$, with fixed α $m_T = |50 - 60| - 0.1$ which doesn't make a proper sense, while using relative α i.e (0.1*60) we get $m_T = |50 - 60| - 6$.

$$m_T = \sum_{i=1}^{T} |C_T - \mu_{T-1}| - \alpha\mu_{T-1} \tag{19}$$

Further to calculate change point score we need a variable M_T which is the minimum value of m_T and is always maintained and updated for every new time step T as given in equation (20).

$$M_T = min(m_T; i = 1...T) \tag{20}$$

8.1.6 Change Point Scoring Function

Aiming to detect the change points and their magnitude after every time-step T in MWA and WMWA, we use the change point scoring function given in equation (21).

$$\Gamma_T = \frac{|C_T - \mu_{T-1}|}{max(C_T, \mu_{T-1})} \tag{21}$$

Where C_T is the current centrality score and μ_{T-1} is the mean of previous centrality scores in the window. The change point scoring function gives the percentage point increase or decrease of the current centrality score with the previous mean. It takes values $0 \leq \Gamma_T \leq 1$.

For a PH test, after every time-step T the change points are scored using the equation (22).

$$\Gamma_T = m_T - M_T \tag{22}$$

8.1.7 Change Point Detection

We can decide the magnitude of change allowed by the above change point scoring function. For this we use a threshold θ on Γ, to signal an alarm of change in the preference of the user/node. It takes values either 0 or 1. "1" indicates a preference change and "0" indicates no change.

$$\epsilon_T = \begin{cases} 1, & \text{if } \Gamma_T \geq \theta. \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

Relative θ: As a relative α given in section 8.1.6 (equation (22)), we also apply a relative θ for detecting change points in PH test only, as the change point scores from windowed approaches are already normalized in equation (21). Therefore to normalize threshold over multiple streams of centrality scores in PH test we use a relative threshold θ by multiplying the threshold θ with M_T of that node at time T as in equation 24.

$$\epsilon_T = \begin{cases} 1, & \text{if } \Gamma_T \geq (\theta \times M_T). \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

8.1.8 Assumptions

While carrying out the above-mentioned mechanisms we considered the following assumptions.

- ❏ For window based approaches, change detection starts only after the window of size W_S is filled.
- ❏ If there exists no edges for a node in a time step T, then the mean is calculated assuming a "0" centrality score.
- ❏ If a node is newly introduced (with edges) in the stream in the time interval T then the previous mean at $T-1$ is considered "0" during change point scoring.
- ❏ In window based approaches if a node does not appear in the stream for a W_S time steps, the node is deleted to save space.

8.1.9 Evaluation

The output of our change detection model is a kind of binary classification problem. For every centrality stream, after every T we label C_T as a preference change point or not. We compare this with our ground truth of preference changes in the data set using a strategy described in section 8.3.2. Therefore, we use recall, precision and F-measure. The experimental results are presented in section 8.4. The purpose of our model is not just to evaluate the efficiency of our model but we intend to compare the different centrality measures and the impact of their changes on the dynamics of real world preferences. The motive of using aggregate mechanisms is to investigate the deviations of current centrality values with the past values over preference changes.

8.2 Algorithms

We now present the algorithms for processing evolving network streams to calculate node centric centrality scores and detect change points. The goal of all algorithms is

to process the evolving network handled as edge stream and store in a binary vector ε^v events detected for node v at each time-step T. The centrality metric $m \in$ {degree, closeness, betweenness} slightly impacts on the network processing approach. Algorithms complexity is ruled by computing centralities algorithms, previously described in Section 8.1.2.

Algorithm 4 refers to Moving Window Average (MWA) strategy. The main loop from line 5 indicates the network evolving. In lines 20–23 we distinguish centralities, as degree is incrementally calculated whereas closeness and betweenness need to store edges for each T to be calculated. Line 6 indicates a transition to next time-step. Lines 7–8 compute betweenness or closeness centrality according to input m using algorithms described in Section 8.1.2.2 and 8.1.2.3, respectivelly. While window W is not full, the moving window average μ is accumulated. When W is full (from line 11), $T - W_S$ centrality scores are forgotten and μ updated with current centrality score (lines 12–14). Finally, line 15 implements the change point scoring function (Eq. 21) and line 16 detects a change point based on θ (Eq. 23).

Algorithm 4 MWA

Input: Target node v, an edge stream $E_1...E_r$, window size W_S, threshold θ, centrality metric m
Output: A binary vector ε^v containing v's events for each time-step T

1: $V \leftarrow \emptyset, E \leftarrow \emptyset, N = (V, E)$
2: $\mu \leftarrow 0, C_T \leftarrow 0$
3: $W \leftarrow \emptyset$ //W is a queue structure representing the window
4: $T \leftarrow t$ //T is the initial time-step
5: **for** each incoming edge stream object $E_i = (u, z, t)$ **do**
6: **if** $t \geq T + 1$ **then** //next time-step
7: **if** $m \in$ {betweenness, closeness} **then**
8: $C_T \leftarrow$ computeCentrality(N, v)
9: **if** $W.size \leq W_S$ **then** //window is not full
10: $\mu \leftarrow \mu + C_T/W_S$
11: **else** //slides window
12: $C_{T-W_S} \leftarrow W.head$
13: $\mu \leftarrow \mu - (C_{T-W_S}/W_S) + (C_T/W_S)$
14: $Dequeue(W)$
15: $\Gamma_T \leftarrow |C_T - \mu|/max(C_T, \mu)$ //change point scoring function
16: $\varepsilon^v_T \leftarrow \Gamma_T \geq \theta \ ? \ 1 : 0$ //change point detection
17: $Enqueue(W, C_T)$
18: $T \leftarrow T + 1$
19: $E \leftarrow \emptyset$, $V \leftarrow \emptyset, C_T \leftarrow 0$
20: **if** $m \in$ {betweenness, closeness} **then**
21: $E \leftarrow E \cup \{E_i\}, V \leftarrow V \cup \{u, z\}$
22: **else if** $m \in$ {degree}, $z = v$ or $u = v$ **then** //update degree incrementally
23: $C_T \leftarrow C_T + 1$
24: **return** ε^v

Algorithm 5 describes the Weighted Moving Window Average (WMWA), which follows

the same sliding window strategy as in MWA. The difference is that the accumulated moving window average μ is composed by linearly weighted centrality scores (lines 13–17).

Algorithm 5 WMWA

Input: Target node v, an edge stream $E_1...E_r$, window size W_S, threshold θ, centrality metric m

Output: A binary vector ε^v containing v's events for each time-step T

– copy from line 1 – 8 of MWA algorithm –

9: **if** $W.size > W_S$ **then** //slides window
10: $sum \leftarrow 0$, $denominator \leftarrow 0$
11: **for** $i \leftarrow 0...W_S - 1$ **do**
12: $sum \leftarrow sum + W[i] * (i + 1)$
13: $denominator \leftarrow denominator + (i + 1)$
14: $\mu \leftarrow sum/denominator$
15: $Dequeue(W)$
16: $\Gamma_T \leftarrow |C_T - \mu|/max(C_T, \mu)$ //change point scoring function
17: $\varepsilon^v_T \leftarrow \Gamma_T \geq \theta \ ? \ 1 : 0$ //change point detection

– continue from line 17 of MWA algorithm –

The Page-Hinkley strategy is implemented by Algorithm 6. As in previous algorithms, network stream processing is represented by the main loop in line 4, considering the differences among centralities computation (lines 18–21 and lines 6–7). The cummulative variable m_T represents the cummulated difference between the latest centrality score C_T and the previous mean μ. Remark that in this strategy we do not handle window. Finally, at line 14 change point scoring function is applied and line 15 implements PH test (Eq. 24).

8.3 Methodology

As in Chapter 7, we used Twitter data to run experiments over the Brazilian news domain. Two different evolving networks were built. The first one – homogeneous network, is the same network presented in Chapter 7. Additionally, we built a bipartite network in order to achieve diversity in our experiments, as detailed in the following.

8.3.1 Dataset and Evolving Networks

Through Twitter Streaming APIs[2], during the course of 94 days, we collected tweets related to Brazilian news. All tweets, retweets and quoted-status [3] containing some mention to the Brazilian newspaper, whose Twitter user is *@folha*, were considered. In all, we collected 1,771,435 tweets and 292,310 distinct users in a time span of tweets posting

[2] https://dev.twitter.com/streaming/
[3] Quoted-status are retweets with comments

Algorithm 6 Page-Hinckley Test (PH)

Input: Target node v, an edge stream $E_1...E_r$, threshold θ, centrality metric m
Output: A vector ε^v containing v's events for each time step T

$V \leftarrow \emptyset$, $E \leftarrow \emptyset$, $N = (V, E)$
$T \leftarrow t$ //T is the initial time step
$m_T \leftarrow 0$, $M_T \leftarrow MAX\_VALUE$, $\mu \leftarrow 0$, $C_T \leftarrow 0$, $instancesSeen \leftarrow 0$
for each incoming edge stream object $E_i = (u, z, t)$ **do**
    if $t \geq T + 1$ **then** //next time step
        if $m \in$ {betweenness, closeness} **then**
            $C_T \leftarrow$ computeCentrality(N, v)
        $instancesSeen \leftarrow instancesSeen + 1$
        $percentualValue \leftarrow \mu * \alpha$
        $m_T \leftarrow m_T + |C_T - \mu| - percentualValue$
        $\mu \leftarrow \mu + C_T / instancesSeen$
        if $m_T < M_T$ **then**
            $M_T \leftarrow m_T$
        $\Gamma_T \leftarrow m_T - M_T$ //change point scoring function
        $\varepsilon_T^v \leftarrow \Gamma_T \geq \theta * M_T \ ? \ 1 : 0$ //change point detection
        $T \leftarrow T + 1$
        $E \leftarrow \emptyset$, $V \leftarrow \emptyset$, $C_T \leftarrow 0$
    if $m \in$ {betweenness, closeness} **then**
        $E \leftarrow E \cup \{E_i\}, V \leftarrow V \cup \{u, z\}$
    else if $m \in$ {degree}, $z = v$ or $u = v$ **then** //update degree incrementally
        $C_T \leftarrow C_T + 1$
return ε^v

times from Aug 8, 2016 to Nov 9, 2016. From the collected data, we built two different evolving networks.

8.3.1.1 Homogeneous Network

The first one, homogeneous network H, is based on retweets. Nodes are Twitter users and two nodes u_1 and u_2 have a direct edge (u_2, u_1, t) if u_1 retweeted u_2 at time t (note that edge direction represents the information flow). Figure 31(a) summarizes the network building strategy. This strategy is similar to the one used in (PEREIRA et al., 2016).

An important characteristic of our H network is that it has a low average path length. This is consequence of the fact that in Twitter a retweet always comes from the original post, not mattering from where the user read that post – from the user who originally posted it or from an intermediate user who already retweeted it. As our dataset has a high diversity of Twitter users, such as celebrities, common users and commercial users, in this network we can identify nodes with different centrality roles. There are nodes that maintain high out-degree during the whole evolving period (mostly is retweeted, thus a content producer), nodes with high in-degree (mostly retweets, thus content consumer) and nodes with balanced behavior. Figure 32 describes nodes and edges evolution behavior. On average, H contains 10,189 nodes and 14,662 edges per day.

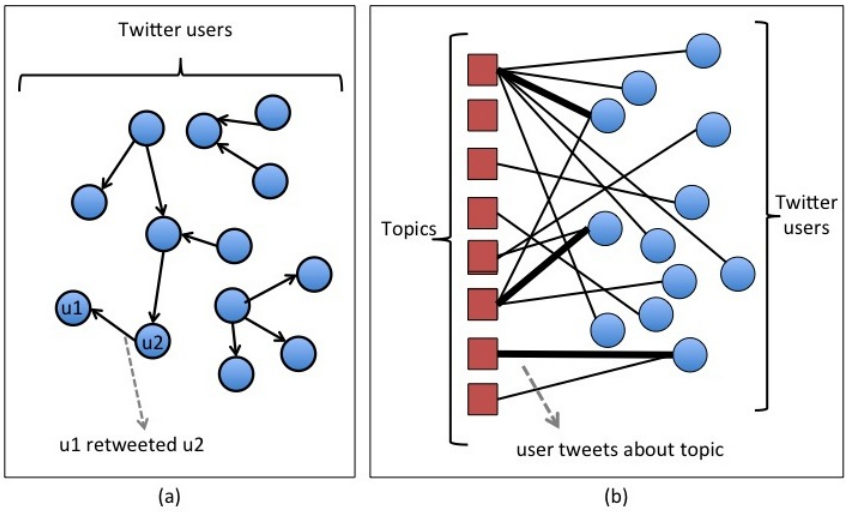

Figure 31 – Strategy to build (a) homogeneous network and (b) bipartite network from Twitter data.

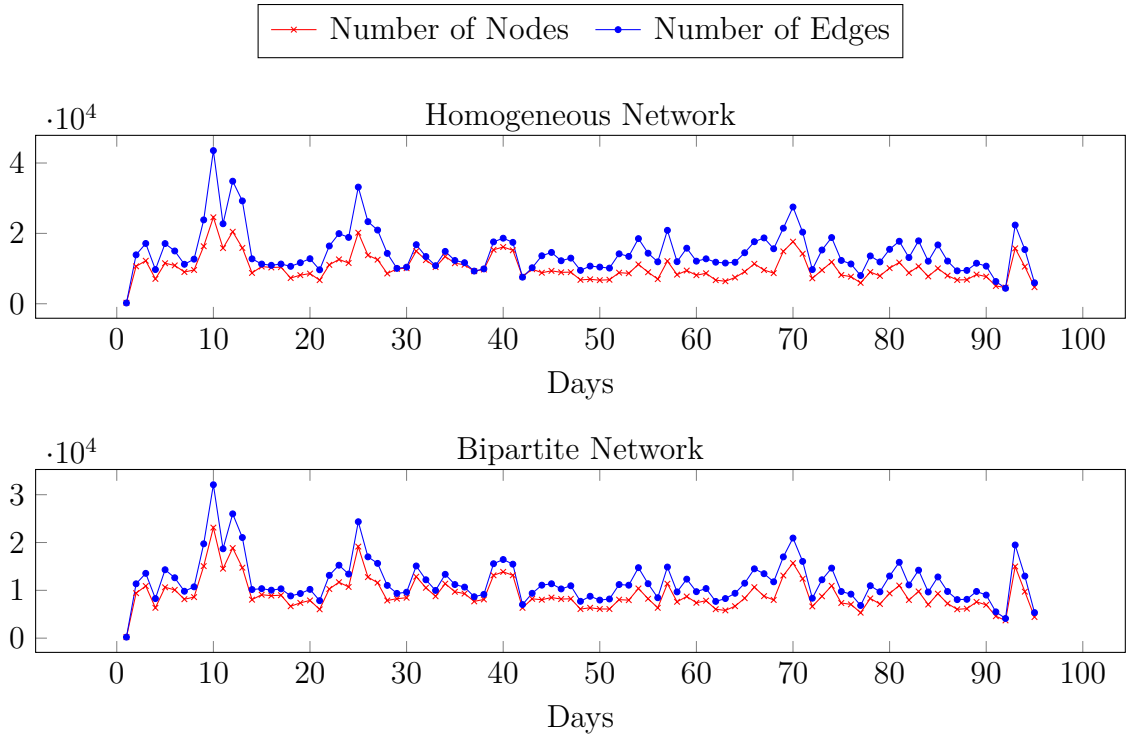

Figure 32 – Evolving behavior description of homogeneous (up) and bipartite (down) networks.

8.3.1.2 Bipartite Network

The second one is a bipartite network B where nodes are Twitter users or topics. Topics represent the main themes that users are *tweeting* about and have been extracted using LDA (Latent Dirichlet Allocation) model. Details of this topic extraction process were presented in Chapter 7. An edge (u, p, t, w) means that user u tweeted/retweeted about topic p at time t, w times. As our time granularity is 1 day, a user can post many times at t and this volume of daily-post is represented through edges weights w. Thus B is a weighted bipartite network. Figure 31(b) summarizes this network building strategy.

The bipartite network represents essentially what is being "talked" in Twitter (topics) and who is talking (users). From a user perspective is possible to track his evolution regarding topics that he mostly interacted with. In all, we defined 10 topics nodes. Thus, these nodes, during the whole network evolution, have high degree value, while user nodes, on average, have low degree values (as users do not talk about several topics in a day). Figure 32 describes nodes and edges evolution behavior. On average, B contains 9,176 nodes, being 10 nodes topics, and 11,892 edges per day.

8.3.2 User Preference Change Events

Considering our dataset, rich of user's interactions over news content, we have extracted users' preferences. The strategy used to extract preferences is the same proposed in Chapter 7. If user u tweets (or retweets) about o at time t, then u has more interest in o over the remaining topics in domain in that moment. We also considered a weight $w_t^u(o)$ based on the number of tweets posted at the same time about some topic o (our time granularity is 1 day; therefore, a user can post many tweets at t). In this case, the top posted topic is preferred over others, the second top posted topic is preferred over remaining ones and so on. $A = \{politics, international, corruption, sports, religion, entertainment, education, economy, security, others\}$ is the set of topics in the preference domain on which we extract user preferences. These are the topics extracted by the LDA model previously described (Chapter 7) and each tweet is labeled with one topic $o \in A$.

In Figure 33 we illustrate the preference change evolution of user u_1 (id=14594813).

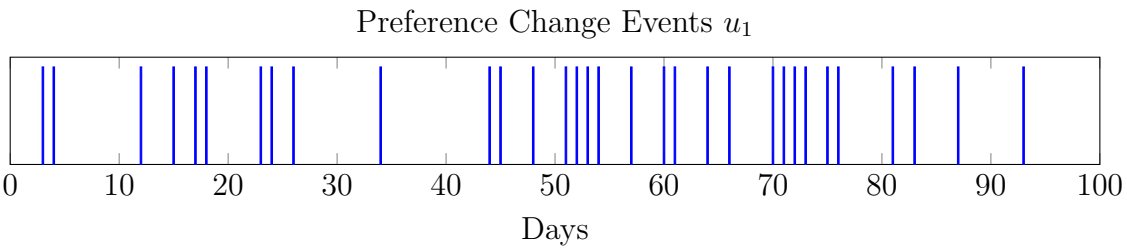

Figure 33 – Ground truth of preference change events for user u_1 in Twitter Brazilian news dataset. A preference change occurs when u_1 posts about different topics in comparison to her usual posting behavior.

Once extracted users' preferences and detected preference change events, we used these change events as ground truth to evaluate our methods for node event detection based on centrality metrics in evolving networks.

8.4 Experiments

The main goal of experiments is to evaluate our proposed preference change detection model. All algorithms were implemented in Java language using Gephi API[4] as foundation[5]. All the experiments run over a server equipped with Intel(R) Xeon(R) CPU @ 2.40GHz on 140GB RAM, twenty cores and Linux Ubuntu operating system.

8.4.1 Experimental Environment

Nodes. We have selected 10 specific nodes to perform our analysis. These nodes were randomly selected among users in the dataset. For the sake of simplicity we refer to them as $u_1, u_2, ..., u_{10}$ and the corresponding dataset ids are: u_1 (14594813), u_2 (334345564), u_3 (3145222787), u_4 (343820098), u_5 (122757872), u_6 (28958495), u_7 (260856271), u_8 (636368737), u_9 (279635698), u_{10} (58488491). All the presented results correspond to the average value among these 10 nodes.

Threshold θ. The threshold corresponds to the magnitude of changes allowed. We varied $\theta = \{0.01, 0.1, 0.2, 0.4\}$ in order to explore how this magnitude impacts on our findings. In PH method we set the factor $\alpha = 0.1$ which is also related to the magnitude of changes we deal with.

Window size W_S. The methods MWA and WMWA are based on sliding window mechanism. We varied $W_S = \{2, 5\}$ and analyzed how this window size can influence in our findings.

Evolving networks. We run experiments considering the homogeneous network H and the bipartite network B.

Centrality metrics. The node centralities we used are degree for B and in-degree for H, betweenness and closeness. During the analysis our target is to compare their efficiency for predicting preference changes of a node.

Methods. Finally, we compare the three proposed methods in this paper MWA, WMWA and PH. Table 13 summarizes the experimental environment.

[4] https://gephi.org/toolkit/
[5] Source codes available at <http://www.lsi.facom.ufu.br/~fabiola/evolving-networks>

Table 13 – Experimental environment

Feature	Variation	Default
Node	$u_1, u_2, ..., u_{10}$	u_1
θ	0.01, 0.1, 0.2, 0.4	0.1
α (PH)	0.1	0.1
W_S (MWA, WMWA)	2, 5	2
Centrality metric	degree/in-degree, betweenness, closeness	in-degree
Evolving Network	bipartite, homogeneous	homogeneous
Methods	MWA, WMWA, PH	MWA

8.4.2 Detecting u_1 change-points

As illustrative example, we describe the change-point detection process for user u_1 considering our default scenario. As time flies, current u_1 in-degree centrality and MWA of past u_1 in-degree centralities are calculated and compared to each other. Then, the change-point scoring function is computed raising alarms when change-points are detected. In the end, the detected change-points and the preference ground-truth are compared to obtain the accuracy of the method.

In Figure 34(a) we show the relation between u_1 in-degree centrality evolution and preference ground truth change-points in homogeneous network (default setup). Intuitively, the expectation is that centrality peaks overlap preference change-points. We can observe that there are some overlaps despite the number of preference change events is higher than peaks.

In Figure 34(b) we depicted the evolution of MWA of u_1 in-degree centrality in the same scenario previously described. The performance of the method is directly related to the balance between the magnitude of changes that we are looking for, i.e. threshold θ value, and the past average values that should be considered, i.e. window size W_S. Considering our default setup (Table 13), we reach precision 0.46, recall 0.9 and F-measure 0.61 for node u_1.

8.4.3 Performance of Proposed Methods

Here we come to the task of evaluating our proposed methods considering different scenarios. Figures 35 and 36 present the results for homogeneous and bipartite networks, respectively. In a general way, bipartite network got lower accuracy than homogeneous network, specially when considering high threshold and window sizes. As in bipartite network topic nodes occupy extremely central positions, the notion of centrality can not be efficient for the task of detecting events.

When observing centrality measures, betweenness clearly got the worst performance. This can be explained by the fact that bridge nodes usually do not vary significantly their positions. Moreover, we can conclude that the notion of bridge itself is not a good

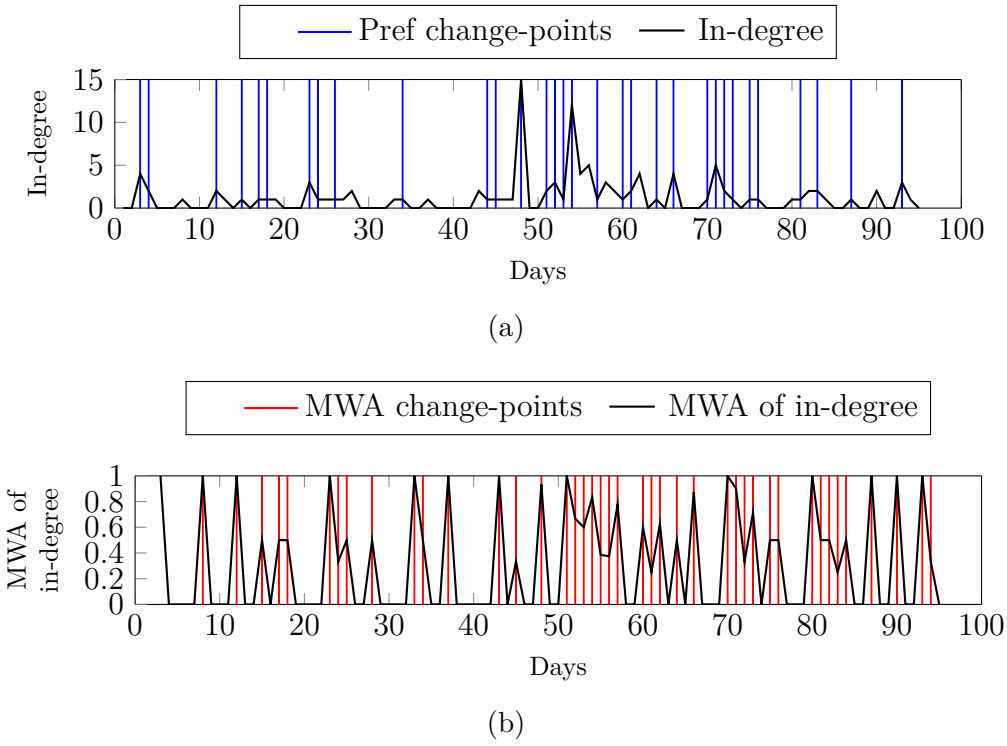

Figure 34 – Comparative analysis between preference change-points and MWA detected change-points considering default setup for node u_1. The accuracy is F-measure = 0.61. (**a**) u_1 in-degree values against u_1 preference ground-truth change-points. (**b**) MWA values for u_1 in-degree against MWA detected change-points.

centrality metric to correlate with preference change. Even so, in most of the scenarios betweenness got performance superior to a naive random baseline (F-measure = 0.5). Comparing degree and closeness, we observe that degree is slightly more accurate than closeness. Considering that the notion of preference change has been defined based on the number of tweets, and consequently number of edges incoming in a node, it was expected that degree centrality fits well in the context.

We also observe that for homogeneous graph, degree centrality performs better and bipartite graphs closeness is superior. This difference in behaviors is based on the graph data. As homogeneous graph is a multi-graph incorporating frequency of edges i.e edge weights are considered while using this graph. In bipartite graph the weights of edges are not considered. Hence we see that when frequency of an edge is considered it is favorable to employ degree centrality, while when we do not know the frequency of edges or for unweighted graphs its beneficial to use closeness. Finally, comparing the performance of our proposed methods, Page-Hinkley got the most significant results, with homogeneous behavior in different scenarios. We can conclude that the idea of accumulating values and base the evolution on the minimum obtained value so far is the most suitable. On average, PH degree performances reach F-measure = 0.75. There is no consensus between MWA and WMWA performances and the choice for one method is not conclusive.

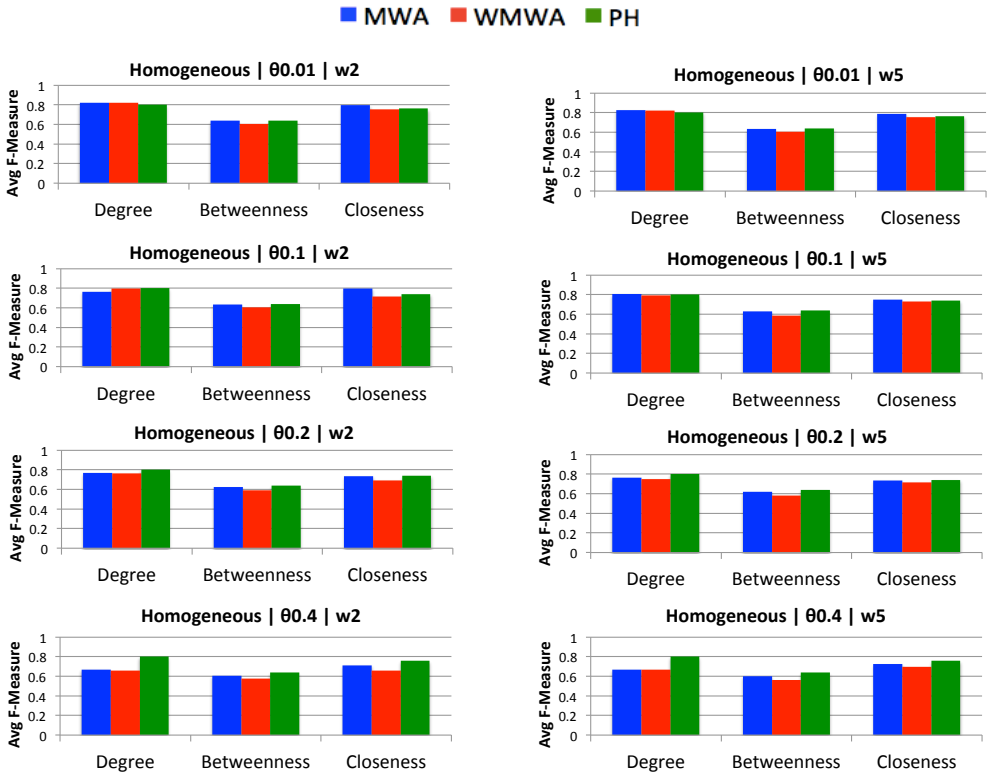

Figure 35 – Performance of our proposed methods in different scenarios for different centrality measures in homogeneous network.

8.4.4 Impact of Parameters

The goal here is to analyze the behavior of parameters trade off. Varying W_S means that we are considering the recent past for low values (short-term events) or a big historic for high values (long-term events). θ adjusts the intensity of the events, varying from smooth to drastic events. From a general viewpoint, we observed that performances keep the proportions according to parameters setup for MWA and WMWA, but not for PH, which got similar results independent of the parameter setup. Another observation is that recall is always higher than precision, except for the highest $\theta = 0.4$. This behavior indicates that performances decrease as the magnitude of allowed changes increase.

8.5 Final Considerations

We have proposed a model for predicting user preference changes. The model processes evolving network streams and detect change points based on centrality measures. We have explored two window-based aggregating mechanisms – moving window average (MWA) and weighted moving window average (WMWA), and a third memory less mechanism, the Page-Hinkley (PH). Moreover, we have implemented algorithms considering degree,

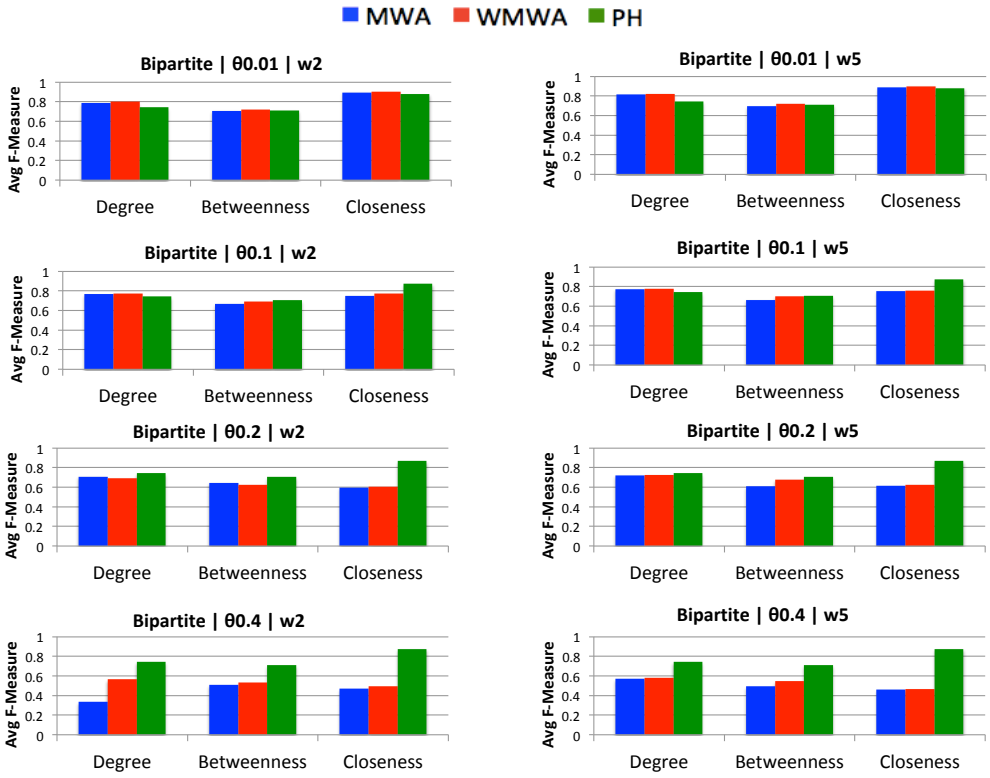

Figure 36 – Performance of our proposed methods in different scenarios for different centrality measures in bipartite network.

betweenness and closeness centrality measures. We have applied our proposed model in the user preference change detection problem and evaluated the performance of our algorithms on homogeneous and bipartite Twitter networks. As result, degree centrality in homogeneous network using PH approach have performed with the highest F-measure values.

It is a working in progress to expand the experiments of our model to the Jam network in music domain, presented in Chapter 7. Moreover, the analysis presented here considered only a sample of 10 users from the whole dataset. We are preparing a set of experiments that explores a generalized behavior considering all the users in the networks.

Chapter 9

Conclusion

Online social networks, such as Facebook,Twitter, and location-based social networks, facilitate the building of social relations among people who share similar interests. Thus, people can stay connected with others and be informed of new trends, consumption preferences and opinions of social friends. Naturally, users tend to change their interest over time, especially in applications where they interact customarily with a wide range of items. At the same time, these social networks grow and change quickly over time with the addition of new nodes and edges, signifying the appearance of new interactions/relations in the underlying social structure. Motivated by this context, the research presented in this monograph aimed to investigate the interplay between User Preferences and Social Networks over time for systems personalization.

The first step was to define what are user preference dynamics (UPD). We have defined a temporal preference model able to describe user preferences over time through user profiles. Then, we have proposed a strategy to detect changes on temporal preferences as time flies. Based on proposed definitions and having in mind our goal, we have stated our main problem: given (i) a user, (ii) her evolving social network and (iii) a set of objects in the preference domain, predict the user preference changes.

As problem solution, we have begun investigating temporal networks. We have proposed modeling a follower/followee Twitter network as a temporal social network. In this analysis the goal was to perceive how nodes evolve in function of centrality metrics.

After temporal networks and evolving centralities in Twitter analysis, we have moved one more step towards problem solution. We have explored the idea of centrality-based node event detection on evolving networks. The goal was to detect at what points in time a node change its behavior significantly. Our proposal was a node event mining model with three different strategies for detecting change points. We have also performed an empirical analysis, now using a Twitter interaction network (retweets), with the intention to observe events detected with closeness centrality and perceive the relation of detected events with network semantics.

We then joined our findings and proposals and performed an experimental evaluation

focused on our main goal: the interplay between user preferences and social networks over time. We have discovered that there is a correlation between preference change events and centrality-based node events, specially when considering temporal networks. Moreover, we have concluded that closeness centrality is more suitable when correlating UPD and node events than betweenness.

Lastly, we have presented a complete solution for the preference change prediction problem, taking into account efficient strategies to detect node events in evolving networks. Remarking on Figure 37 we summarize a generic schema for UPD analysis with social networks focusing on preference change prediction.

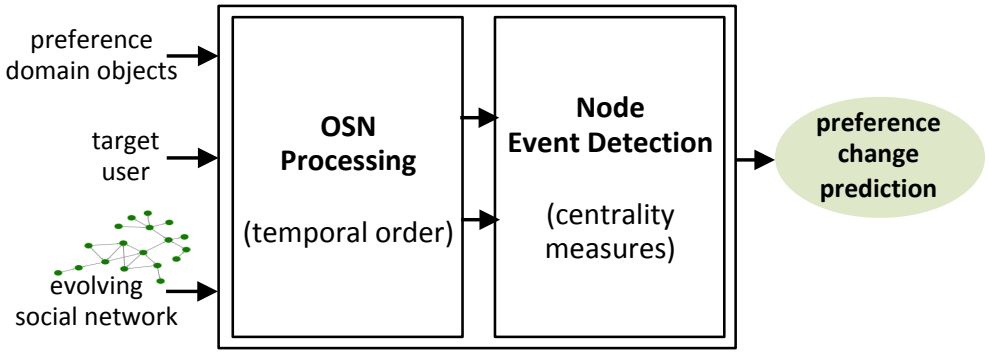

Figure 37 – A generic schema for UPD analysis with social networks focusing on preference change prediction. OSN states for online social network.

9.1 Main Contributions

The empirical research conducted produced a series of original contributions to the fields of social network analysis, dynamic network analysis and preference learning. We highlight:

- ❏ introduction and formalization of the UPD problem focusing on preference changes;
- ❏ an innovative analysis of users' temporal centralities evolution in Twitter network modeled as a temporal social network;
- ❏ a new approach for centrality-based events detection in continuous evolving networks;
- ❏ correlation of user preference changes and centrality-based events in temporal networks;
- ❏ a preference change prediction model.

9.2 Summary of datasets and source codes

We made available all datasets used in this thesis [1]. Those related with Twitter were first collected from crawlers. The TIMJ dataset was released by Jansson et al. (2015) and we made available the temporal network we built. Table 14 summarizes our datasets. Source codes refer to Twitter crawlers, temporal centralities plugin and event detection algorithms, besides user preference extractors.

Table 14 – Summary of network datasets used in this thesis. ITV: interval graph; CT: contact sequence graph.

Dataset	Network Definition	Temporal Network Dimension	Type	Chapter
Twitter celebrities	followers	144,975 nodes 1,222,118 temporal edges 109 days	ITV	5
Twitter Brazilian News	retweets	78,944 nodes 108,133 temporal edges 21 days	CT	6
Twitter Brazilian News	retweets	292,310 nodes 1,392,841 temporal edges 94 days	CT	7, 8
Twitter Brazilian News Bipartite	retweets/ topics	243,284 nodes 1,129,701 temporal edges 94 days	CT	8
This Is My Jam	likes	54,393 nodes 1,667,335 temporal edges 1493 days	CT	7

9.3 Bibliographical Contributions

The main chapters that make up the body of this thesis correspond to improved and extended versions of the following published articles:

Journal articles

1. Pereira, F. S. F.; Gama, J.; Amo, S.; Oliveira, G. M. B.. *On analyzing user preference dynamics with temporal social networks.* Machine Learning Journal. 2018. 107:1745. https://doi.org/10.1007/s10994-018-5740-2

2. Tabassum, S.; Pereira, F. S. F.; Fernandes, S.; Gama, J. *Social Network Analysis: An Overview.* WIREs Data Mining and Knowledge Discovery. 2018. e1256. https://doi.org/10.1002/widm.1256.

[1] Datasets and source codes available at: <http://lsi.facom.ufu.br/~fabiola/evolving-networks/>

Under submission:

3. Pereira, F. S. F.; Tabassum, S.; Cordeiro, M.; Gama, J. *Sociometrics on Streaming Graphs.* ACM Computing Surveys.

4. Pereira, F. S. F.; Gama, J.; Amo, S.; Oliveira, G. M. B.. *A Gephi Plugin for Temporal Centrality Metrics.* Social Network Analysis and Mining.

5. Pereira, F. S. F.; Gama, J.; Amo, S.; Oliveira, G. M. B.. *Am I really eclectic? Temporal Dynamics of Users' Musical Preferences.* iSys Brazilian Journal of Information Systems.

Conference/workshop articles

1. Pereira, F. S. F. *Mining Comparative Sentences from Social Media Text.* In: 2nd Workshop on Data Mining and Natural Language Processing (DMNLP) co-located with ECML PKDD 2015.

2. Pereira, F. S. F.; de Amo, S. *Mineração de Preferências do Usuário em Textos de Redes Sociais usando Sentenças Comparativas.* In Symposium on Knowledge Discovery, Mining and Learning (KDMiLe) 2015.

3. Pereira, F. S. F.; de Amo, S.; Gama, J. *Evolving centralities in temporal graphs: a twitter network analysis.* In: Mobile Data Management (MDM), 2016 17th IEEE International Conference on., 2016.

4. Pereira, F. S. F.; de Amo, S.; Gama, J. *On using temporal networks to analyze user preferences dynamics.* In 19th International Conference on Discovery Science (DS) 2016.

5. Pereira, F. S. F.; de Amo, S.; Gama, J. *Detecting Events in Evolving Social Networks through Node Centrality Analysis.* Workshop on Large-scale Learning from Data Streams in Evolving Environments (StreamEvolv) co-located with ECML PKDD 2016.

6. Pereira, F. S. F. *Student Research Abstract: User Preferences Dynamics on Evolving Social Networks - Learning, Modeling and Prediction.* 32nd Annual ACM Symposium on Applied Computing 2017 (SAC '17). [**among top-5 posters**]

7. Pereira, F. S. F.; Linhares, C.; Ponciano, J.; Gama, J.; Amo, S.; Oliveira, G. M. B.. *Sou mesmo eclético? Uma Análise Temporal sobre a Evolução das Preferências dos Usuários em uma Rede Social de Músicas.* BraSNAM 2018. [**best paper award - 2nd place**]

Book chapters

1. Pereira, F. S. F.; Gama, J.; Oliveira, G. M. B. *Extraindo Conhecimento de Redes Sociais Temporais.* Tópicos em Gerenciamento de Dados e Informações. 2017.

2. Pereira, F. S. F.; Tabassum, S.; Gama, J.; Amo, S.; Oliveira, G. M. B. *Processing Evolving Social Networks for Change Detection based on Centrality Measures.* Moamar Sayed-Mouchaweh (eds.), Learning from Data Streams in Evolving Environments, Studies in Big Data 41, Springer, 2019.

Tutorials/short-courses

1. Pereira, F. S. F.; Gama, J.; Oliveira, G. M. B. *É uma questão de tempo! Extraindo Conhecimento de Redes Sociais Temporais.* Short-course in Brazilian Symposium on Databases SBBD'17.

2. Pereira, F. S. F.; Tabassum, S.; Gama, J. *Knowledge Discovery from Temporal Social Networks.* SIAM SDM 2018.

During the course of my Ph.D. studies, I published additional articles in related topics and embraced opportunities to collaborate with other researchers. These originated the following (selected) manuscript:

1. Claudio Linhares, Jean Ponciano, Fabiola S. F. Pereira, Luis Rocha, José Gustavo Paiva, Bruno Travençolo. *A Scalable Node Ordering Strategy Based on Community Structure for Enhanced Temporal Network Visualization.* Information Visualization. (under submission)

9.4 Directions for Future Research

The new methodologies described in this monograph can be improved and extended in a number of ways, opening several opportunities for future research. Some ideas are outlined below.

1. **Social Network Analysis**

   - ❑ **Evaluation metrics.** Evaluation without a ground truth in social media research is a pressing need (ZAFARANI; LIU, 2015). In our context, for instance how could be possible to determine if detected node events indeed are events for a given user? As we have proposed several node-centrality events detection strategies, could be interesting to explore these different ways and measure the accuracy based on a metric founded on the intersection between the events detected by all. A survey about evaluation metrics in social networks research would be valuable.

- ❑ **UPD Analytics on social networks.** The idea is to explore many other research questions when analyzing UPD in social networks. For instance, is there a causality relation? Who are the most influential users and how they are positioned in the network when considering a preference change? Are there exogenous factors influencing user preferences? Are there bots influencing real users in social media? To what extent bots can impact the preference change phenomenon?
- ❑ **Applications of UPD analysis on different domains.** Though our analysis is limited to the Twitter news and music domains due to availability of public datasets, we expect our results to generalize to other items like movies, videos, books, vacation packages, shopping etc. which are fairly susceptible to social influence effects. UPD analysis over different social networks is valuable and can result in rich contributions.
- ❑ **Applications of UPD analysis on different social network measures.** A natural extension of our study is to explore other social network measures. For instance, is there a relation among network communities and the users' preferences? Is PageRank an important metric for UPD? Furthermore, our approach was developed based on bottom-up decisions in which we first raised centrality measures and then investigated correlations. A future research can explore top-down decisions based first on meaningful correlation questions.

2. **Evolving Networks**

- ❑ **Stream processing centrality metrics.** Design algorithms that avoid a full re-computation of centrality metrics is an open challenge. For instance, evaluating the betweenness centrality for a graph $G = (V, E)$ is computationally demanding and the best known algorithm for unweighted graphs has an upper bound time complexity of $O(V^2 + VE)$. Consequently, it is desirable to find a way to avoid a full re-computation of betweenness centrality when a new edge is inserted into the graph. The works (KAS et al., 2013b), (KAS et al., 2013a) and (BAHMANI et al., 2010) propose fast algorithms for betweenness, closeness and PageRank computation, respectively.
- ❑ **Algorithms for temporal centrality metrics.** There is a lack of algorithms designed for temporal centralities. We have proposed a baseline algorithm for temporal closeness and betweenness (Chapter 5). Specially, as in the previous item, algorithms that avoid full re-computation as the network evolves.

3. **Preference Learning, Modeling and Prediction**

- ❑ **Preference mining in social networks.** We have noticed a lack of concise methods for preference mining in social networks. Enumerating, we have

explored text mining comparative sentences (PEREIRA; AMO, 2015), mining from network structure (PEREIRA et al., 2016b) and the topic modeling strategy which is a most mature method that resulted in better findings (Chapter 7). There are many open avenues in this line, specially in evaluation methods.

- ❑ **Comparison among methods for preference change prediction.** In this thesis we do not expand our analysis for comparison with related work that adopt different preference models than the order over pairwise objects that we define. A future research line is to deep into methods for correspondence among user preferences formalisms. The results will enable further comparisons between our preference change prediction approach and those proposed in (KAPOOR, 2014; MOORE et al., 2013; ZHANG et al., 2014b).
- ❑ **Descriptive preference change prediction.** In our work we are able to predict that a preference change will occur for a given user, but we are not able to describe *which* preference the user will change. There are several open avenues in this line of research. Predictive methods with high descriptive quality are valuable for business insights.

Bibliography

ABBASI, M. A.; TANG, J.; LIU, H. Scalable learning of users' preferences using networked data. In: **Proceedings of the 25th ACM Conference on Hypertext and Social Media**. New York, NY, USA: ACM, 2014. (HT '14), p. 4–12. ISBN 978-1-4503-2954-5. Disponível em: <http://doi.acm.org/10.1145/2631775.2631796>.

AGARWAL, D.; CHEN, B.-C. flda: Matrix factorization through latent dirichlet allocation. In: **Proceedings of the Third ACM International Conference on Web Search and Data Mining**. New York, NY, USA: ACM, 2010. (WSDM '10), p. 91–100. ISBN 978-1-60558-889-6. Disponível em: <http://doi.acm.org/10.1145/1718487.1718499>.

AGGARWAL, C.; SUBBIAN, K. Evolutionary network analysis: a survey. **ACM Computing Surveys**, v. 47, n. 1, p. 10–36, 2014.

AGGARWAL, C.; XIE, Y.; YU, P. S. On dynamic link inference in heterogeneous networks. p. 415–426, 2012. Disponível em: <http://epubs.siam.org/doi/abs/10.1137/1.9781611972825.36>.

AGGARWAL, C. C. Recommender systems: The textbook. Springer Publishing Company, Incorporated, 2016.

AGGARWAL, C. C.; LI, N. On node classification in dynamic content-based networks. p. 355–366, 2011. Disponível em: <http://epubs.siam.org/doi/abs/10.1137/1.9781611972818.31>.

AGGARWAL, C. C.; SUBBIAN, K. Event detection in social streams. In: **12th SIAM International Conference on Data Mining, USA.** [S.l.: s.n.], 2012. p. 624–635.

AHMED, N. K.; NEVILLE, J.; KOMPELLA, R. Network sampling: From static to streaming graphs. **ACM Trans. Knowl. Discov. Data**, ACM, New York, NY, USA, v. 8, n. 2, p. 7:1–7:56, jun. 2013. ISSN 1556-4681. Disponível em: <http://doi.acm.org/10.1145/2601438>.

AKOGLU, L.; FALOUTSOS, C. Event detection in time series of mobile communication graphs. In: **Proceedings of 27th army science conference**. [S.l.: s.n.], 2010. (18, 3), p. 1–8.

AKOGLU, L.; TONG, H.; KOUTRA, D. Graph based anomaly detection and description: a survey. **Data Mining and Knowledge Discovery**, v. 29, n. 3, p. 626–688, 2015.

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. **Rev. Mod. Phys.**, American Physical Society, v. 74, p. 47–97, Jan 2002. Disponível em: <http://link.aps.org/doi/10.1103/RevModPhys.74.47>.

ALTHOFF, T.; JINDAL, P.; LESKOVEC, J. Online actions with offline impact: How online social networks influence online and offline user behavior. In: **Proceedings of the Tenth ACM International Conference on Web Search and Data Mining**. New York, NY, USA: ACM, 2017. (WSDM '17), p. 537–546. ISBN 978-1-4503-4675-7. Disponível em: <http://doi.acm.org/10.1145/3018661.3018672>.

AMO, S. de et al. Contextual preference mining for user profile construction. **Information Systems**, v. 49, p. 182 – 199, 2015.

ARIAS, M.; ARRATIA, A.; XURIGUERA, R. Forecasting with twitter data. **ACM Trans. Intell. Syst. Technol.**, v. 5, n. 1, p. 8:1–8:24, 2014.

BAHMANI, B.; CHOWDHURY, A.; GOEL, A. Fast incremental and personalized pagerank. **Proc. VLDB Endow.**, VLDB Endowment, v. 4, n. 3, p. 173–184, dez. 2010. ISSN 2150–8097. Disponível em: <http://dx.doi.org/10.14778/1929861.1929864>.

BASTIAN, M.; HEYMANN, S.; JACOMY, M. Gephi: An open source software for exploring and manipulating networks. **International AAAI Conference on Weblogs and Social Media**, 2009. Disponível em: <http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154>.

BERTIN-MAHIEUX, T. et al. The million song dataset. **Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)**, 2011.

BIANCONI, G.; BARABÁSI, A.-L. Competition and multiscaling in evolving networks. **EPL (Europhysics Letters)**, v. 54, n. 4, p. 436, 2001. Disponível em: <http://stacks.iop.org/0295-5075/54/i=4/a=436>.

BIFET, A. et al. Mining frequent closed graphs on evolving data streams. In: **17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2011. (KDD '11), p. 591–599.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. **J. Mach. Learn. Res.**, v. 3, p. 993–1022, mar. 2003. ISSN 1532-4435.

BOCCALETTI, S. et al. Complex networks: Structure and dynamics. **Physics Reports**, v. 424, n. 4–5, p. 175–308, 2006.

BöRZSöNYI, S.; KOSSMANN, D.; STOCKER, K. The skyline operator. In: **Proceedings of the 17th International Conference on Data Engineering**. [S.l.: s.n.], 2001. p. 421–430.

BRANDES, U. A faster algorithm for betweenness centrality. **Journal of Mathematical Sociology**, v. 25, p. 163–177, 2001.

BUNTAIN, C.; LIN, J. Burst detection in social media streams for tracking interest profiles in real time. In: **39th International ACM SIGIR conference**. [S.l.: s.n.], 2016. p. 777–780.

CADILHAC, A. et al. Preference change. **Journal of Logic, Language and Information**, v. 24, n. 3, p. 267–288, 2015. ISSN 1572-9583.

CARTWRIGHT, D.; HARRARY, F. A generalization of heider's theory. **Psychological Review**, v. 63, p. 277–292, 1956.

CATTUTO, C. et al. Time-varying social networks in a graph database: a neo4j use case. In: ACM. **First International Workshop on Graph Data Management Experiences and Systems**. [S.l.], 2013. p. 1–6.

CHAKRABARTI, D.; FALOUTSOS, C.; MCGLOHON, M. Graph mining: Laws and generators. Springer US, Boston, MA, p. 69–123, 2010. Disponível em: <http://dx.doi.org/10.1007/978-1-4419-6045-0_3>.

CHANDRAMOULI, B. et al. Streamrec: A real-time recommender system. In: **Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data**. [S.l.: s.n.], 2011. (SIGMOD '11), p. 1243–1246. ISBN 978-1-4503-0661-4.

CHRISTIDIS, K.; APOSTOLOU, D.; MENTZAS, G. Exploring customer preferences with probabilistic topics models. In: **Preference Learning workshop, ECML/PKKD**. [S.l.: s.n.], 2010. p. 1–13.

CORDEIRO, M.; GAMA, J. Online social networks event detection: A survey. Springer International Publishing, Cham, p. 1–41, 2016. Disponível em: <http://dx.doi.org/10.1007/978-3-319-41706-6_1>.

COSTA, L. F. et al. Characterization of complex networks: a survey of measurements. **Advances in Physics**, v. 56, n. 1, p. 167–242, 2007.

DEBNATH, S.; GANGULY, N.; MITRA, P. Feature weighting in content based recommendation system using social network analysis. In: **Proceedings of the 17th International Conference on World Wide Web**. [S.l.: s.n.], 2008. (WWW '08), p. 1041–1042. ISBN 978-1-60558-085-2.

EBERLE, W.; HOLDER, L. Identifying anomalies in graph streams using change detection. In: **KDD Workshop on Mining and Learning in Graphs (MLG)**. [S.l.: s.n.], 2016.

FAIRBANKS, J. et al. A statistical framework for streaming graph analysis. In: **IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining**. [S.l.: s.n.], 2013. (ASONAM '13), p. 341–347.

FELÍCIO, C. Z. et al. Exploiting social information in pairwise preference recommender system. **Journal of Information and Data Management (JIDM)**, v. 7, n. 2, p. 99–115, 2016.

FURNKRANZ, J.; HULLERMEIER, E. **Preference Learning**. [S.l.]: Springer, New York, 2010.

GAMA, J. **Knowledge Discovery from Data Streams**. [S.l.]: Chapman & Hall/CRC, 2010. ISBN 1439826110, 9781439826119.

GAMA, J.; SEBASTIÃO, R.; RODRIGUES, P. P. On evaluating stream learning algorithms. **Machine learning**, Springer, v. 90, n. 3, p. 317–346, 2013.

GAMA, J. et al. A survey on concept drift adaptation. **ACM Computing Surveys (CSUR)**, ACM, v. 46, n. 4, p. 44, 2014.

GLANZER, M. Stimulus satiation: An explanation of spontaneous alternation and related phenomena. **Psychological Review**, v. 60, n. 4, p. 257–268, August 1953.

GUHA, S.; MCGREGOR, A. Graph synopses, sketches, and streams: A survey. **Proc. VLDB Endow.**, VLDB Endowment, v. 5, n. 12, p. 2030–2031, ago. 2012. ISSN 2150-8097. Disponível em: <http://dx.doi.org/10.14778/2367502.2367570>.

GUILLE, A.; HACID, H. A predictive model for the temporal dynamics of information diffusion in online social networks. In: **Proceedings of the 21st International Conference on World Wide Web**. New York, NY, USA: ACM, 2012. (WWW '12 Companion), p. 1145–1152. ISBN 978-1-4503-1230-1. Disponível em: <http://doi.acm.org/10.1145/2187980.2188254>.

GUILLE, A. et al. Information diffusion in online social networks: A survey. **SIGMOD Rec.**, ACM, New York, NY, USA, v. 42, n. 2, p. 17–28, jul. 2013. ISSN 0163-5808. Disponível em: <http://doi.acm.org/10.1145/2503792.2503797>.

GUPTA, M. et al. Evolutionary clustering and analysis of bibliographic networks. In: **2011 International Conference on Advances in Social Networks Analysis and Mining**. [S.l.: s.n.], 2011. p. 63–70.

GUY, I. Social recommender systems. Springer US, Boston, MA, p. 511–543, 2015. Disponível em: <http://dx.doi.org/10.1007/978-1-4899-7637-6_15>.

HANSSON, S. O. Changes in preference. **Theory and Decision**, v. 38, n. 1, p. 1–28, 1995. ISSN 1573-7187.

HOLME, P. Analyzing temporal networks in social media. **Proceedings of the IEEE**, v. 102, n. 12, p. 1922–1933, 2014.

HOLME, P.; SARAMAKI, J. Temporal networks. **Physics Reports**, v. 519, n. 3, p. 97–125, 2012.

HUANG, Y. et al. Tencentrec: Real-time stream recommendation in practice. In: **Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data**. [S.l.: s.n.], 2015. (SIGMOD '15), p. 227–238. ISBN 978-1-4503-2758-9.

IDE, T.; KASHIMA, H. Eigenspace-based anomaly detection in computer systems. In: **Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2004. (KDD '04), p. 440–449.

IMRAN, M.; CHAWLA, S.; CASTILLO, C. A robust framework for classifying evolving document streams in an expert-machine-crowd setting. In: **Proceedings of the 18th International Conference on Data Mining (ICDM)**. [S.l.: s.n.], 2016. p. 961–966.

IMRAN, M.; LYKOURENTZOU, I.; CASTILLO, C. Engineering crowdsourced stream processing systems. **CoRR**, 2013. Disponível em: <http://arxiv.org/abs/1310.5463>.

JANSSON, A.; RAFFEL, C.; WEYDE, T. This is my jam – data dump. **16th International Society for Music Information Retrieval Conference Late Breaking and Demo Papers**, p. 1–2, 2015.

KAPOOR, K. **Models of dynamic user preferences and their applications to recommendation and retention**. Tese (Doutorado) — University of Minnesota, 2014.

KAPOOR, K. et al. Measuring spontaneous devaluations in user preferences. In: **Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2013. (KDD '13), p. 1061–1069. ISBN 978-1-4503-2174-7.

KAS, M.; CARLEY, K. M.; CARLEY, L. R. Incremental closeness centrality for dynamically changing social networks. In: **Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining**. New York, NY, USA: ACM, 2013. (ASONAM '13), p. 1250–1258. ISBN 978-1-4503-2240-9. Disponível em: <http://doi.acm.org/10.1145/2492517.2500270>.

KAS, M. et al. Incremental algorithm for updating betweenness centrality in dynamically growing networks. In: **Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining**. New York, NY, USA: ACM, 2013. (ASONAM '13), p. 33–40. ISBN 978-1-4503-2240-9. Disponível em: <http://doi.acm.org/10.1145/2492517.2492533>.

KEMPE, D.; KLEINBERG, J.; KUMAR, A. Connectivity and inference problems for temporal networks. In: **Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing**. [S.l.]: ACM, 2000. p. 504–513.

KIM, H.; ANDERSON, R. Temporal node centrality in complex networks. **Phys. Rev. E**, American Physical Society, v. 85, p. 026107, Feb 2012.

KIM, M.-S.; HAN, J. A particle-and-density based evolutionary clustering method for dynamic networks. **Proc. VLDB Endow.**, VLDB Endowment, v. 2, n. 1, p. 622–633, ago. 2009. ISSN 2150–8097. Disponível em: <http://dx.doi.org/10.14778/1687627.1687698>.

KOREN, Y. Collaborative filtering with temporal dynamics. In: **Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2009. (KDD '09), p. 447–456. ISBN 978-1-60558-495-9.

KWAK, H.; CHUN, H.; MOON, S. Fragile online relationship: A first look at unfollow dynamics in twitter. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2011. (CHI '11), p. 1091–1100. ISBN 978-1-4503-0228-9. Disponível em: <http://doi.acm.org/10.1145/1978942.1979104>.

LEE, M.-J.; CHOI, S.; CHUNG, C.-W. Efficient algorithms for updating betweenness centrality in fully dynamic graphs. **Information Sciences**, Elsevier Science Inc., New York, NY, USA, v. 326, n. C, p. 278–296, jan. 2016. ISSN 0020-0255. Disponível em: <http://dx.doi.org/10.1016/j.ins.2015.07.053>.

LESKOVEC, J.; KLEINBERG, J.; FALOUTSOS, C. Graphs over time: Densification laws, shrinking diameters and possible explanations. In: **Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining**. New York, NY, USA: ACM, 2005. (KDD '05), p. 177–187. ISBN 1-59593-135-X. Disponível em: <http://doi.acm.org/10.1145/1081870.1081893>.

LESKOVEC, J.; RAJARAMAN, A.; ULLMAN, J. D. **Mining of Massive Datasets**. 2nd. ed. [S.l.]: Cambridge University Press, 2014. ISBN 9781107077232.

LI, J.; RITTER, A.; JURAFSKY, D. Inferring user preferences by probabilistic logical reasoning over social networks. **CoRR**, 2014. Disponível em: <http://arxiv.org/abs/1411.2679>.

LIU, F. **Reasoning about Preference Dynamics**. 1. ed. [S.l.]: Springer Netherlands, 2011. (354). ISBN 9789400713444.

LIU, X. Modeling users' dynamic preference for personalized recommendation. In: **Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)**. [S.l.: s.n.], 2015. p. 1785–1791.

LOU, J.-K. et al. Modeling the diffusion of preferences on social networks. In: **Proceedings of the 2013 SIAM International Conference on Data Mining**. [S.l.: s.n.], 2013. p. 605–613.

MACROPOL, K. et al. I act, therefore i judge: Network sentiment dynamics based on user activity change. In: **Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining**. [S.l.: s.n.], 2013. (ASONAM '13), p. 396–402.

METAXAS, P. et al. What do retweets indicate? results from user survey and meta-review of research. **Ninth International AAAI Conference on Web and Social Media (ICWSM)**, p. 658–661, 2015.

MOORE, J. et al. Taste over time: the temporal dynamics of user preferences. In: **Proceedings of the 14th International Society for Music Information Retrieval Conference**. [S.l.: s.n.], 2013. p. 1–6.

MOUSS, H. et al. Test of page-hinckley, an approach for fault detection in an agro-alimentary production system. In: IEEE. **Control Conference, 2004. 5th Asian**. [S.l.], 2004. v. 2, p. 815–818.

NICOSIA, V. et al. **Temporal Networks**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. 15–40 p. ISBN 978-3-642-36461-7.

OLIVEIRA, M.; GUERREIRO, A.; GAMA, J. Dynamic communities in evolving customer networks: an analysis using landmark and sliding windows. **Social Network Analysis and Mining**, v. 4, n. 1, p. 208, 2014. ISSN 1869-5469. Disponível em: <http://dx.doi.org/10.1007/s13278-014-0208-2>.

PAGE, E. S. Continuous inspection schemes. **Biometrika**, JSTOR, v. 41, n. 1/2, p. 100–115, 1954.

PAN, R. K.; SARAMÄKI, J. Path lengths, correlations, and centrality in temporal networks. **Phys. Rev. E**, v. 84, 2011.

PAPINI, J. A. J.; AMO, S. d.; SOARES, A. K. S. Strategies for mining user preferences in a data stream setting. **Journal of Information and Data Management**, v. 5, n. 1, p. 64–73, February 2014.

PEREIRA, F. S. F. . User preferences dynamics on evolving social networks - learning, modeling and prediction: Student research abstract. In: **Proceedings of the Symposium on Applied Computing**. [S.l.: s.n.], 2017. (SAC '17), p. 1090–1091. ISBN 978-1-4503-4486-9.

PEREIRA, F. S. F. Mining comparative sentences from social media text. In: **Workshop on Interactions between Data Mining and Natural Language Processing (DMNLP) co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)**. [S.l.: s.n.], 2015. p. 41–48.

PEREIRA, F. S. F.; AMO, S.; GAMA, J. Evolving centralities in temporal graphs: a twitter network analysis. In: **Mobile Data Management (MDM), 2016 17th IEEE International Conference on**. [S.l.: s.n.], 2016. p. 43–48.

PEREIRA, F. S. F.; AMO, S. d.; GAMA, J. On using temporal networks to analyze user preferences dynamics. In: **Discovery Science: 19th International Conference, DS 2016, Bari, Italy, 2016.** [S.l.: s.n.], 2016.

PEREIRA, F. S. F.; AMO, S. de. Mineracao de preferencias do usuario em textos de redes sociais usando sentencas comparativas. In: **Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)**. [S.l.: s.n.], 2015. p. 94–97.

PEREIRA, F. S. F.; AMO, S. de; GAMA, J. Detecting events in evolving social networks through node centrality analysis. **Workshop on Large-scale Learning from Data Streams in Evolving Environments co-located with ECML/PKDD**, 2016.

PEREIRA, F. S. F. et al. On analyzing user preference dynamics with temporal social networks. **Machine Learning**, p. 408–423, 2018.

PEREIRA, F. S. F.; TABASSUM, S.; GAMA, J. **Processing Evolving Social Networks for Change Detection Based on Centrality Measures**. Cham: Springer International Publishing, 2019. 155–176 p. ISBN 978-3-319-89803-2. Disponível em: <https://doi.org/10.1007/978-3-319-89803-2_7>.

RAFAILIDIS, D.; NANOPOULOS, A. Modeling the dynamics of user preferences in coupled tensor factorization. In: ACM. **Proceedings of the 8th ACM Conference on Recommender systems**. [S.l.], 2014. p. 321–324.

RANSHOUS, S. et al. Anomaly detection in dynamic networks: a survey. **Wiley Interdisciplinary Reviews: Computational Statistics**, v. 7, n. 3, p. 223–247, 2015. ISSN 1939-0068.

ROSSI, R. et al. Role-dynamics: Fast mining of large dynamic networks. In: **Proceedings of the 21st International Conference on World Wide Web**. New York, NY, USA: ACM, 2012. (WWW'12 Companion), p. 997–1006. ISBN 978-1-4503-1230-1. Disponível em: <http://doi.acm.org/10.1145/2187980.2188234>.

ROZENSHTEIN, P. et al. Event detection in activity networks. In: **Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2014. (KDD '14), p. 1176–1185.

SANTORO, N. et al. Time-varying graphs and social network analysis: Temporal indicators and metrics. **CoRR**, 2011. Disponível em: <http://arxiv.org/abs/1102.0629>.

SCHLITTER, N.; FALKOWSKI, T. Mining the dynamics of music preferences from a social networking site. In: **Social Network Analysis and Mining, 2009. ASONAM '09. International Conference on Advances in**. [S.l.: s.n.], 2009. p. 243–248.

SCHREIBER, H. Improving genre annotations for the million song dataset. In: **Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR**. [S.l.: s.n.], 2015. p. 241–247.

SEBASTIÃO, R. et al. Real-time algorithm for changes detection in depth of anesthesia signals. **Evolving Systems**, Springer, v. 4, n. 1, p. 3–12, 2013.

SIDDIQUI, Z. F. et al. xstreams: Recommending items to users with time-evolving preferences. In: **Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)**. [s.n.], 2014. (WIMS '14), p. 22:1–22:12. ISBN 978-1-4503-2538-7. Disponível em: <http://doi.acm.org/10.1145/2611040.2611051>.

SUN, J.; TANG, J. A survey of models and algorithms for social influence analysis. In: AGGARWAL, C. C. (Ed.). **Social Network Data Analytics**. [S.l.]: Springer US, 2011. p. 177–214. ISBN 978-1-4419-8461-6.

SUN, Y. et al. Measuring user preference changes in digital libraries. In: **Proceedings of the 17th ACM Conference on Information and Knowledge Management**. [s.n.], 2008. (CIKM '08), p. 1497–1498. ISBN 978-1-59593-991-3. Disponível em: <http://doi.acm.org/10.1145/1458082.1458353>.

TABASSUM, S.; GAMA, J. Sampling massive streaming call graphs. In: **Proceedings of the 2016 ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2016. (SAC '16), p. 923–928.

TAN, C. et al. Social action tracking via noise tolerant time-varying factor graphs. In: **Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2010. (KDD '10), p. 1049–1058. ISBN 978-1-4503-0055-1.

TANG, J. et al. Temporal distance metrics for social network analysis. In: **Proceedings of the 2nd ACM Workshop on Online Social Networks**. [S.l.: s.n.], 2009. (WOSN '09), p. 31–36. ISBN 978-1-60558-445-4.

TANG., J. et al. Analysing information flows and key mediators through temporal centrality metrics. In: **Proceedings of the 3rd Workshop on Social Network Systems**. New York, NY, USA: ACM, 2010. (SNS '10), p. 3:1–3:6. ISBN 978-1-4503-0080-3. Disponível em: <http://doi.acm.org/10.1145/1852658.1852661>.

THIMM, M. Dynamic preference aggregation under preference changes. In: **Proceedings of the Fourth Workshop on Dynamics of Knowledge and Belief (DKB'13)**. [S.l.: s.n.], 2013.

VINAGRE, J.; JORGE, A. M.; GAMA, J. Fast incremental matrix factorization for recommendation with positive-only feedback. Springer International Publishing, Cham, p. 459–470, 2014.

VISWANATH, B. et al. On the evolution of user interaction in facebook. In: **Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)**. [S.l.: s.n.], 2009.

WALLACH, H. M.; MIMNO, D.; MCCALLUM, A. Rethinking lda: Why priors matter. In: **Proceedings of the 22Nd International Conference on Neural Information Processing Systems**. [S.l.: s.n.], 2009. (NIPS'09), p. 1973–1981.

WEI, W.; CARLEY, K. M. Measuring temporal patterns in dynamic social networks. **ACM Trans. Knowl. Discov. Data**, ACM, New York, NY, USA, v. 10, n. 1, p. 9:1–9:27, jul. 2015. ISSN 1556-4681. Disponível em: <http://doi.acm.org/10.1145/2749465>.

WENZEL, F.; KIESSLING, W. A preference-driven database approach to reciprocal user recommendations in online social networks. In: HARTMANN, S.; MA, H. (Ed.). **Database and Expert Systems Applications: 27th International Conference, DEXA 2016, Porto, Portugal, September 5-8, 2016, Proceedings, Part II**. Cham: Springer International Publishing, 2016. p. 3–10. ISBN 978-3-319-44406-2. Disponível em: <http://dx.doi.org/10.1007/978-3-319-44406-2_1>.

WILSON, N. Extending cp-nets with stronger conditional preference statements. In: **Proceedings of the 19th National Conference on Artifical Intelligence**. [S.l.: s.n.], 2004. (AAAI'04), p. 735–741.

WU, H. et al. Path problems in temporal graphs. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 7, n. 9, p. 721–732, 2014.

WU, L. et al. Modeling users' preferences and social links in social networking services: A joint-evolving perspective. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2016. p. 279–286.

XIANG, L. et al. Temporal recommendation on graphs via long- and short-term preference fusion. In: **Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2010. (KDD '10), p. 723–732. ISBN 978-1-4503-0055-1. Disponível em: <http://doi.acm.org/10.1145/1835804.1835896>.

YANG, J.; LESKOVEC, J. Modeling information diffusion in implicit networks. In: **Proceedings of the 2010 IEEE International Conference on Data Mining**. Washington, DC, USA: IEEE Computer Society, 2010. (ICDM '10), p. 599–608.

YANOFF, T. G.; HANSSON, S. O. **Preference Change - Approaches from Philosophy, Economics and Psychology**. [S.l.]: Springer Netherlands, 2009. v. 42.

ZAFARANI, R.; ABBASI, M. A.; LIU, H. **Social Media Mining: An Introduction**. New York, NY, USA: Cambridge University Press, 2014. ISBN 1107018854, 9781107018853.

ZAFARANI, R.; LIU, H. Evaluation without ground truth in social media research. **Commun. ACM**, ACM, New York, NY, USA, v. 58, n. 6, p. 54–60, maio 2015. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2666680>.

ZHANG, J. A survey on streaming algorithms for massive graphs. Springer US, Boston, MA, p. 393–420, 2010. Disponível em: <http://dx.doi.org/10.1007/978-1-4419-6045-0_13>.

ZHANG, J.; WANG, C.; WANG, J. Learning temporal dynamics of behavior propagation in social networks. In: **Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2014. (AAAI'14), p. 229–236.

ZHANG, J. et al. Inferring continuous dynamic social influence and personal preference for temporal behavior prediction. **Proc. VLDB Endow.**, VLDB Endowment, v. 8, n. 3, p. 269–280, nov. 2014. ISSN 2150-8097.

ZHANG, Y.; ZHOU, J.; CHENG, J. Preference-based top-k influential nodes mining in social networks. In: **2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications**. [S.l.: s.n.], 2011. p. 1512–1518. ISSN 2324-898X.

ZIGNANI, M.; GAITO, S.; ROSSI, G. P. Follow the mastodon: Structure and evolution of a decentralized online social network. In: **International AAAI Conference on Web and Social Media**. [S.l.: s.n.], 2018. p. 541–550.