

Tilburg University

Tempora Tempore Tempora: Time molds the past

Arena, Giuseppe

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

Arena, G. (2023). *Tempora Tempore Tempora: Time molds the past: Time-sensitive methods for modeling the influence of past social interactions in relational event networks*. Gildeprint.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

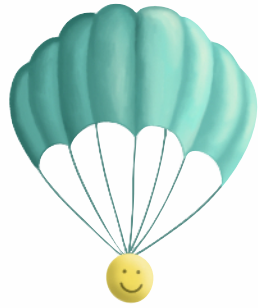
Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



TEMPORA TEMPORE TEMPERA: TIME MOLDS THE PAST

Time-sensitive methods for modeling the
influence of past social interactions in relational
event networks



Giuseppe Arena

TEMPORA TEMPORE TEMPERA: TIME MOLDS THE PAST

Time-sensitive methods for modeling the influence
of past social interactions in relational event networks

Giuseppe Arena

Copyright © 2023 Giuseppe Arena, The Netherlands, CC-BY 4.0.

ISBN: 978-94-6419-883-6

Design by Giuseppe Arena

Printed by Gildeprint – www.gildeprint.nl

The research presented in this dissertation is funded by the European Research Council (ERC) [Starting Grant project number 758791].



European Research Council
Established by the European Commission

TEMPORA TEMPORE TEMPORA: TIME MOLDS THE PAST

Time-sensitive methods for modeling the influence
of past social interactions in relational event networks

Proefschrift ter verkrijging van de graad van doctor aan Tilburg University op
gezag van de rector magnificus, prof. dr. W.B.H.J. van de Donk, in het
openbaar te verdedigen ten overstaan van een door het college voor
promoties aangewezen commissie in de Aula van de Universiteit op vrijdag 20
oktober 2023 om 13.30 uur

door

Giuseppe Arena

geboren te Catania, Italië

Promotors:

dr. ir. J. Mulder (Tilburg University)

prof. dr. R.T.A.J. Leenders (Tilburg University)

Leden promotiecommissie:

prof. dr. T.A.B. Snijders (University of Groningen)

prof. dr. J.K. Vermunt (Tilburg University)

prof. dr. E.M. Wagenmakers (University of Amsterdam)

dr. B.M. Brandenberger (ETH, Zürich)

Ai miei nipoti,

Michele, Salvo e Alessandra

CONTENTS

1 Introduction to relational event networks, endogenous network dynamics and the concept of memory decay	1
1.1 Outline of the dissertation	8
2 Understanding employee communication with longitudinal social network analysis of email flows	11
2.1 Introduction	12
2.2 Digital innovation communication networks	13
2.3 The relational event modeling framework	14
2.3.1 Description of the data	14
2.3.2 The model	15
2.3.3 Model comparison	17
2.4 Discussion and Conclusion	20
3 A Bayesian semi-parametric approach for modeling memory decay in dynamic social networks	21
3.1 Introduction	22
3.2 Relational event models that capture memory decay	26
3.3 A step-wise memory decay model	28
3.3.1 Step-wise decay for first-order endogenous effects	28
3.3.2 Step-wise decay for higher order endogenous effects	31
3.3.3 Estimation of a relational event model with a step-wise memory decay	36
3.4 The gradual nature of memory decay	37
3.5 A semi-parametric approach to estimate a smooth memory decay	40
3.5.1 Generating a bag of step-wise relational event models	41
3.5.2 Evaluating the fit of the step-wise relational event models	43
3.5.3 Bayesian model averaging for approximating smooth decay functions	45
3.5.4 Computational details of the BMA	47
3.6 Case study: investigating the presence of memory decay in the sequence of demands sent among Indian socio-political actors	50

3.6.1	Relational events between socio-political actors	51
3.6.2	Predefined step-wise decay models	53
3.6.3	Approximately smooth memory decay models	53
3.6.4	Assessing the predictive performance: a comparison with parametric memory decays	56
3.7	Discussion	60
4	How fast do we forget our past social interactions? Understanding memory retention with parametric decays in relational event models	65
4.1	Introduction	66
4.2	Parametric functions for modeling memory decay	70
4.3	The Profile log-likelihood in REM	75
4.4	Simulations: synthetic relational event histories with memory decay	78
4.4.1	Exploring bias based on a misspecified memory decay	86
4.4.2	Testing different decay functions via the Bayes Factor	87
4.4.3	Exploring estimation errors due to a misspecified hazard function	89
4.5	Investigate memory decay in empirical relational event networks	90
4.5.1	Demands among Indian socio-political actors	90
4.5.2	Text messages among students	95
4.6	Discussion	101
5	Weighting the past: An extended relational event model for negative and positive events	103
5.1	Introduction	104
5.2	SentiREM: A model for relational events with sentiments	107
5.2.1	The estimation of the model parameters	112
5.2.2	Testing of model parameters	115
5.3	Numerical performance of the SentiREM	117
5.4	Case study: modeling memory decay of trades and attacks between players in an online strategy game	120
5.4.1	Data and research questions	121
5.4.2	Model specification and results	123
5.4.3	A comparison of SentiREMs with fixed memory decays or without memory decay	134
5.5	Discussion	135

6 Discussion	137
A	143
A.1 Endogenous statistics	144
A.2 From step-wise to continuous effects	145
A.3 Interval generator (the algorithm)	147
A.4 Maximum likelihood estimates for the models specified in the model comparison	148
B	151
B.1 Weights following a step-wise function	152
B.2 Sms data (sub-networks with 1 cluster and 2 clusters): trend of MLEs when the weight decay is exponential	154
C Software packages for the analysis of relational event networks	157
C.1 <code>remify</code> : pre-processing raw relational event sequences	158
C.1.1 A function for processing raw data	158
C.1.2 A function for transforming processed event histories into different formats	170
C.2 <code>remestimate</code> : optimization tools for relational event history data	172
C.3 <code>bremory</code> : modeling the influence of past social interactions in relational event networks	176
Bibliography	179
Summary	186
Acknowledgements	189

INTRODUCTION TO RELATIONAL EVENT
NETWORKS, ENDOGENOUS NETWORK
DYNAMICS AND THE CONCEPT OF
MEMORY DECAY

1

Chapter 1

Statistical models for longitudinal social network data aim to understand which factors and network dynamics trigger actors' interactions and to investigate the evolution of global level social structures starting from the analysis and quantification of behavioral patterns observed at an actor level. Drivers of social interaction can depend on the past event history or on actors' attributes.

Until the first decades of the 2000s, network data were available in an aggregated form at different time points and only in some case studies researchers analyzed networks in which the timestamp of each interaction was inferred by some means of communication such as e-mails (Kossinets & Watts, 2006) or online messages (Panzarasa et al., 2009). However, the increasing availability of such longitudinal network data stimulated researchers to develop modeling frameworks that account for the specific network structure. Models like SAOM (Snijders, 2017b), TERGM (Hanneke et al., 2010) and software packages like `RSiena` (Snijders, 2017a), `tergm` (Krivitsky & Handcock, 2022), `btergm` (Leifeld et al., 2018), are known to have contributed to the analysis of such type of aggregated data structures.

With the constant advancements of new technologies, such as internet of things (IoT), radio-frequency identification (RFID), virtual reality (VR), the collection of longitudinal network data has become considerably easier on different fields and with higher precision (such as the exact timing of relational observations) resulting in new challenges to properly deal with the analysis of such continuous streams of relational data with the goal to better understand (i.e., with higher resolution) temporal social interaction dynamics. Such improvements have led to the formulation of statistical models that were more appropriate to capture the intrinsic dynamic nature of the data and to handle the presence of a fine-grained time dimension.

A time-to-event sequence of social interactions among actors can be referred to as a "relational event network" where a relational event was defined by Butts (2008) as a directed social interaction that is initiated by a sender and is targeted to one or more receivers at a specific time point.

An example of a sequence of relational events is shown in Table 1.1, where students of a classroom interact with one another. In the example, we do not only know the exact time of an event, in addition to the sender and receiver, but we also know the sentiment of each event which describes a characteristic of the action that might be either the verb describing the action itself or some

time	sender	receiver	sentiment
t_1	Alice	Peter	positive
t_2	Peter	Alice	positive
t_3	Grace	Alice	negative
...
...
t_M	David	Peter	negative

Table 1.1: Example of hypothetical events in a network of students in a classroom.

adjective that qualifies the interaction, for instance, if a student is offending another student we might classify that event as a “negative” interaction, or if a student is praising another student that event can be labeled as a “positive” interaction. We describe any event in the sequence via the 4-tuple (time, sender, receiver, sentiment), for instance, the first event can be written as $(t_1, \text{Alice}, \text{Peter}, \text{positive})$, the second event as $(t_2, \text{Peter}, \text{Alice}, \text{positive})$. In the third event, Grace joins the conversation with a negative remark to Alice $(t_3, \text{Grace}, \text{Alice}, \text{negative})$, which could trigger either a negative remark of Alice to Grace (as a response) or a negative remark of Peter to Grace (because Peter had positive communication with Alice who was the receiver of a negative remark by Grace).

Relational event networks can be observed in many other contexts in which different behavioral patterns might be investigated as well as factors that are related to the actors and to the social background in which they are embedded. For instance, we might study a sequence of interactions occurring among gangs that belong to different neighborhoods in the same metropolitan area, a sequence of e-mails and in-person communications among employees in a company, a network of digital and in-person interactions among freshmen in a university and so forth.

In literature, there exist two statistical models for modeling relational event data: a tie-oriented modeling framework that is the relational event model (REM) introduced by Butts (2008), and an actor-oriented modeling framework that was formulated by Stadtfeld and Block (2017). The tie-oriented framework models the event rate of any possible relational event, that is the speed at which relational events occur over a period of time. Whereas, the actor-oriented approach models the dynamic process of a relational event by means of two separate models: one model for the sender’s interaction rate and one choice model for the selection of the receiver operated by the observed sender. By the choice

Chapter 1

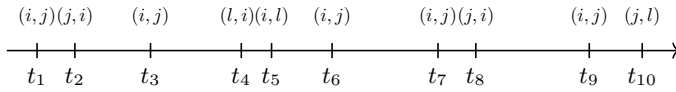


Figure 1.1: Example of relational event sequence of ten events where 3 actors (i , j and l) interact one another.



Figure 1.2: Graph of the pattern characterizing the statistic Inertia for the dyad (i , j).

of any of the two modeling approaches, social scientists aim to inquire which actor or dyad level statistics and which measures of network dynamics play a crucial role in the establishment and development or in the dissolution of network structures.

In this dissertation we will distinguish statistics into two categories: exogenous statistics and endogenous statistics, which serve as time-varying predictor variables in a REM. The exogenous statistics are actor-level or dyad-level statistics that are constant or time-varying and are not computed based on the network of past relational events. The endogenous statistics, instead, summarize network dynamics of different complexities and, their value is updated at each observed time point (time-varying) with respect to the network of past occurred events. We further classify endogenous statistics based on how many actors describe the behavioral pattern that characterizes them: first-order endogenous statistics quantify network dynamics in which only two actors are involved (one directed dyadic event), second-order endogenous statistics in which three actors describe a triangular structure (two directed dyadic events) and, higher-order statistics in which four or more actors describe more complex patterns (three or more directed dyadic events). Furthermore, it would also be possible to include interactions between endogenous and exogenous statistics.

To simplify the introduction and explanation of several concepts that establish the fundamentals of this dissertation, we consider a short relational event sequence of ten events in Figure 1.1 where 3 actors i , j and l interact one another and we focus on the computation of inertia (first-order endogenous statistic) for the dyad (i , j). In relational event networks, inertia quantifies the number of times in which a dyadic event is observed prior to a specific time point. Therefore, the value of inertia is time-varying and different for every dyad in

Inertia						
t_m	dyad					
	(i, j)	(j, i)	(i, l)	(l, i)	(j, l)	(l, j)
t_1	0	0	0	0	0	0
t_2	1	0	0	0	0	0
t_3	1	1	0	0	0	0
t_4	2	1	0	0	0	0
t_5	2	1	0	1	0	0
t_6	2	1	1	1	0	0
t_7	3	1	1	1	0	0
t_8	4	1	1	1	0	0
t_9	4	2	1	1	0	0
t_{10}	5	2	1	1	0	0

Table 1.2: Value of inertia over time and across the six possible dyads from the example in Figure 1.1. At the first time point, t_1 , the value of inertia is set to 0 for all the dyads.

the network. In a relational event model, the estimated effect for the inertia measures the tendency of actors to target their future interactions towards recipients that have been their usual targets in the past. In other terms, the estimated effect for the statistic inertia helps to understand the presence of forms of routinizations between individuals in the network. An example of the behavioral pattern of inertia is described by the graph in Figure 1.2 where it is referred to the dyad (i, j) . Therefore, at each time point, the computation of inertia consists of the volume of past events in which we observe the dyad of interest. For the dyad (i, j) the value of inertia at any observed time point t_m with $m = 1, \dots, 10$ is calculated as

$$\text{inertia}(i, j, t_m) = \sum_{t < t_m} \mathbb{I}(s_{e_t} = i, r_{e_t} = j) \quad (1.1)$$

where the sum is extended to those events that occurred prior the time point t_m , s_{e_t} and r_{e_t} are the sender and the receiver of the past event e observed at time t , $\mathbb{I}(\cdot)$ is an indicator function that assumes value 1 if the condition inside the function is true, 0 otherwise. In this case, the indicator variable assumes value one if both the sender and the receiver of the past event were respectively i and j . The formula in (1.1) can be easily extended to the other dyads in the network. Table 1.2 reports the value of inertia over time and for the six possible dyads (i, j) , (j, i) , (i, l) , (l, i) , (j, l) and (l, j) . The considerations that are made here and for the rest of the introduction apply also to other network dynamics

short-run Inertia								
t_m	timestamp		dyad					
	(dd-mm hh:mm)		(i, j)	(j, i)	(i, l)	(l, i)	(j, l)	(l, j)
t_1	02-10	08:53	0	0	0	0	0	0
t_2	02-10	11:01	1	0	0	0	0	0
t_3	02-10	21:31	1	1	0	0	0	0
t_4	03-10	09:47	2	1	0	0	0	0
t_5	03-10	11:35	1	1	0	1	0	0
t_6	03-10	19:18	1	0	1	1	0	0
t_7	04-10	11:07	2	0	1	1	0	0
t_8	04-10	13:06	2	0	1	0	0	0
t_9	05-10	01:23	2	1	0	0	0	0
t_{10}	05-10	05:14	2	1	0	0	0	0

Table 1.3: Value of short-run inertia defined by events occurred in the last 24 hours. The time of occurrence of each event is provided as a timestamp (“dd-mm hh-mm”, that is day-month hours-minutes). By column, the statistic is calculated for the six possible dyads from the example in Figure 1.1. Two rectangles on the table highlight: (i) the value of the short-run inertia at t_{10} for all the dyads (rectangle on the right) and, (ii) the set of events that occurred in the last 24 hours before t_9 and that are considered in the computation of the short-run statistic (rectangle on the left).

like reciprocity, in-/out-degree of the sender (receiver), transitivity closure, cyclic closure and so forth.

The computation of the network dynamics as a count inherently assumes that long-passed events have the same influence as more recent events, and thereby contribute equally to the interaction rate between actors. In other terms, past events would never change their influence as the time transpired since their occurrence increases, even for events that have passed a very long time ago. Therefore, this means that actors’ future interactions are influenced by long-passed events as much as by events that occurred very recently (days or even hours). This definition is not realistic, rather we expect that the influence of events that happened in the remote past is lower than more recent events.

In the literature, several studies propose novel definitions of endogenous statistics based on time intervals of the past history of events where, for instance, they use a short-run and long-run definition of the statistics (Quintane et al., 2013) exploring the differences between the effects of short-run and long-run dynamics. In Table 1.3, we calculate the value of short-run inertia as the count of the past events that occurred in the last 24 hours. We can rewrite the

formula in (1.1), as

$$\text{inertia}_{\text{short-run}}(i, j, t_m) = \sum_{t: (t_{m-1}-t) \leq 24\text{h}} \mathbb{I}(s_{e_t} = i, r_{e_t} = j) \quad (1.2)$$

In order to provide an example of the short-run formula applied to the sequence of events in Figure 1.1, consider the last time point t_{10} and the two rectangles on the Table 1.3 which highlight: (i) the value of the short-run inertia at t_{10} for all the dyads (rectangle on the right) and, (ii) the set of events that occurred in the last 24hours before t_9 and that are considered in the computation of the short-run statistic (rectangle on the left).

A similar approach consists in dividing the past history of events in multiple intervals and compute the network dynamics of interest in each interval (Perry & Wolfe, 2013), in this way obtaining a step-wise-like function of effects for each endogenous statistics specified in the model. Other studies assume an exponential decay of the event weight that is embedded in the computation of the endogenous statistics (Brandenberger, 2018b; Brandes et al., 2009). Therefore, the weight of every past event is updated at each time point according to the assumed parametric decay and their contribute to any endogenous statistic decreases over time towards zero. Such methods explored to different extents the concept that past behavioral patterns have a time-varying influence both on the computation of endogenous statistics and on the resulting effect on the rate of future social interactions.

Even though these contributions have been useful, and a fertile ground to further develop and extend REMs to better understand temporal social interaction dynamics, these methods are limited as the bounds for short-run and long-run effects and the steepness of memory decay are prespecified, and therefore not estimated from the data. This is limiting from a statistical point of view as specifying badly fitting values for these elements can result in bias and incorrect inferences, but also substantively, as it would not allow us to better understand how and how fast the importance of past events changes as a function of the time that has transpired since observing the event. The main goal of this dissertation is to address these shortcomings by developing time-sensitive model extensions around the relational event network modeling framework. Given these extensions, another aim is to contribute to our understanding to whether memory retention is present in relational event networks.

1.1 Outline of the dissertation

The methods presented in this dissertation build upon the tie-oriented approach (REM) but they can also easily be adapted and applied to an actor-oriented modeling framework.

In Chapter 2, we present a real case study in which we analyze the sequence of emails about innovation that were sent among employees in a multinational service company. In this study we want to understand whether and how the location of employees, the difference in hierarchy, the norms of reciprocity and inertia defined in both short-run and long-run form explain the email rate of employees and the level of information sharing about innovation.

In Chapter 3, we introduce a semi-parametric method for estimating the shape of memory decay in relational event models. In the method, first we consider a "bag" of many step-wise relational event models where endogenous statistics are calculated based on different interval configurations (one interval configuration per each step-wise model), we fit each step-wise model and, finally, we model the shape of the decay with an approach based on the Bayesian Model Averaging theory. The semi-parametric approach is then applied to an empirical event sequence of demands sent among socio-political actors in India.

In Chapter 4, we propose a method that assumes a parametrized function for the memory decay (exponential, linear or step-wise) which depends on one memory parameter that is optimized using the observed event sequence. We also propose a Bayesian test to establish which decay function fits the data best. Finally, we apply the methodology to two case studies: the first case study is about the same network of demands analyzed in Chapter 3, the second case study is about the time-ordered sequence of text messages sent among a group of freshmen university students.

In Chapter 5, we present the SentiREM for modeling the event rate of the next dyadic event separately from the probability of the next event type. We focus on the case of two discrete event types and we model the event sentiment via Probit regression. We define sentiment-based endogenous statistics which assume the same parametric decay function but the memory parameters differ based on the sentiment. We optimize the sentiment-based memory parameters along with the effects of the statistics using the observed event sequence and we introduce some Bayesian test for the memory parameters as well as for

the sentiment-based endogenous effects. Finally, we apply the method to a sequence of attacks and trades occurring among a group of players of an online strategy game.

The methods developed in Chapters 3, 4 and 5, are available in the R package `bremory`. In Appendix C, we describe the aim and the usage of the R software packages `remify`, `remstimate` and `bremory`.

UNDERSTANDING EMPLOYEE COMMUNICATION WITH LONGITUDINAL SOCIAL NETWORK ANALYSIS OF EMAIL FLOWS

2

Abstract

In modern society, innovation in organizations consists of team-based activities where groups of individuals work together on projects and produce novel ideas from their interaction. In this chapter, we analyze a real case study in which a multinational service company stimulated employees' mind towards the generation of innovative ideas by different means, such as: providing new stimulating resources to employees, sending newsletters, organizing gatherings and events, throwing challenges with prizes and other activities. The study focuses on the analysis of the stream of e-mails about innovation ideas that were sent among the employees and aims to understand the influence of specific factors on the e-mail rate. We estimate and compare relational event models where network dynamics were included with a short-run and a long-run definition. We also discuss how the estimated short and long term effects highlight the difference in importance on future communication between interactions that happened recently and interactions that occurred in a remote past.

2.1 Introduction

Innovation is the spice of life for organizations and is generally seen as a requirement for long-term survival and attaining and sustaining above-average performance. Yet, innovation can be hard to accomplish.

In this case study, we consider the innovation struggle of a European branch of a multinational service company (referred to in the case study as STRATSERV). Innovation typically requires a company's employees to change the way they do their work, either by doing different things (such as providing a new service or engaging in new procedures) or by doing things differently (such as using new technology to do the work more efficiently). This means that, especially in service organizations, innovation can hardly be successful without the willingness of employees to change (the way they do) their work. This realization stimulated STRATSERV's management to attempt to open the minds of their employees to innovation. Hence, they organized various events where employees could suggest innovative ways of working, offered prizes for the best ideas, and provided resources to employees to explore their ideas further. In sum, the approach was to first open the minds of employees to the idea of innovation, stimulate the employees to come up with innovative suggestions, and then build on that joint openness to the innovation in order to implement new services and new procedures. Of course, this assumes that the minds of the STRATSERV employees would respond favorably and long-lasting to the company's innovative wishes.

Although the STRATSERV management believed in this approach, they also realized that they needed a way to test whether their approach was working. Did their efforts indeed create an innovation mindset in the heads of their employees and did that mindset last? Moreover, they wondered if all employees responded alike or whether the competitions, gatherings, newsletters, challenges, and other activities organized by the company's taskforce only affected certain employees but not others.

In this situation, it makes little sense to send out a survey to the employees, asking them whether they were thinking about innovation regularly. This would likely trigger socially acceptable answers and could not provide the detailed insight into the effect of the activities that the company was looking for. In addition, surveys are poorly suited to monitor how employees respond over time, including repeated surveys. The company reached out for help to an external

team of researchers. Below, we will show part of the analysis that was performed.

2.2 Digital innovation communication networks

When employees discuss innovation, an innovation communication network emerges within the company. The structure and pervasiveness of this network are key indicators whether STRATSERV's approach is working. In addition, innovative activity is essentially a network activity (Aalbers et al., 2016; Kratzer et al., 2006; R. T. Leenders et al., 2003). Innovation is, by necessity, a collaborative effort. Existing knowledge and ideas merge into new combinations, and as formerly separated knowledge comes together, new knowledge emerges. Although the imagery of the lone inventor profoundly developing is appealing, it is an image rarely found in modern times. Innovation is a "team sport", where individuals work together in teams, teams work together in projects, organizations work together in alliances, and countries work together in international technology agendas. In fact, even the mythical lone inventor probably rarely operated in splendid isolation anyway, since it is likely that much of the inventor's inspiration came from interaction with other people or organizations, the financial resources may have been granted by banks or friends, the actual development of the product often involved the help of factories, and customers had to become involved in order to test the product for feasibility. No matter which (great) innovation one would look at, it is bound to be couched in network interaction of some sort (R. T. A. J. Leenders & Dolfsma, 2016). In sum, an ideal approach to see if innovation was catching on as a core topic and activity inside STRATSERV was to measure how the innovation communication network developed.

Networks can be measured in a number of ways. The most common approach is to administer surveys to ask who communicates with whom. Alternatively, one could observe the interactions of employees throughout their working activities. These methods don't work in our case, since we wanted to follow the interactions of employees in real time for a full year. Alternative tools such as using video to see who interacts with whom or collecting data from proximity badges would not provide information on whether the conversation included innovation as a topic. Hence, the choice was made to analyze the email interaction between the employees over the course of a year.

Digital communication, in particular email, has become one of the most important means of communication in organizations. As email leaves digital traces about senders, receivers, and timing, these rich network data contain high-resolution information to understand how communication structures change when working teams reach deadlines, to understand new employee integration processes (and how these are affected by cultural differences and team compositions), or to understand how ideas spread through a network of employees (and how this is affected by the actors' hierarchical positions, for example). Besides the academic/theoretical interest these insights are also useful from a practical point of view as they can be used to optimize communication structures in deadline situations, they can be used to optimize the integration processes of new employees, and they can be used to reach all employees regarding certain working topics as fast as possible.

In this case study, we show one approach that can be used to study and understand how networks evolve over time, in real time, and how this knowledge can be leveraged in practice.

2.3 The relational event modeling framework

2.3.1 Description of the data

Our analysis focuses on the innovation communication networks in a European branch of STRATSERV. After developing and implementing procedures to ensure employee privacy and informed consent was received from the parties involved, we used text mining techniques to score the email messages on whether the exchanged text addressed innovation-related topics. The empirical data in this case study consist of a time-ordered sequence of $M = 1340$ email messages that were exchanged between 153 employees over the course of a year. An example of the data is given in Table 2.1 where each row represents the 3-tuple (t_m, s_m, r_m) , with respectively the time, the sender, and the receiver of the m -th email in the sequence of emails $E = \{(t_1, s_1, r_1), \dots, (t_M, s_M, r_M)\}$.

We assume that email interaction is regulated and driven by factors that can depend either on workers' characteristics (e.g., one's status or outgoingness), on the dyadic characteristics of sender and receiver (e.g., hierarchy differences, co-location), on the history of workers' past interactions (e.g., the exchange of email that occurred in the past), or on the workers' location in the social structure (e.g.,

time	sender	receiver
03 Jan 2010 08:21:33	Marco	Jane
03 Jan 2010 08:43:09	Jane	Marco
⋮	⋮	⋮
31 Dec 2010 18:39:22	Paul	Jane

Table 2.1: Example of longitudinal network of emails

interaction with joint colleagues, norms of reciprocity). In particular, we will focus on modeling whether and how this email stream depends on working in the same building, the difference in hierarchy level between sender and receiver of the email, the tenure of the sender, the tendency of sender and receiver to continue to exchange email messages among each other (i.e. persistence or inertia) and the norms of reciprocity between employees. Moreover, we allow a possible memory effect where recent email activity may have a relatively large effect on the future activity between actors.

2.3.2 The model

The novel modeling framework that is well suited to analyze time-to-event sequence data in networks is the so-called *Relational Event Model* (REM) (Butts, 2008; R. T. A. J. Leenders et al., 2016; Mulder & Leenders, 2019). This framework aims to model the rate at which specific directed interaction (i.e. a given email being sent) between two actors (here: employees) occurs; in other words, we model the *emailing rate* among any pair of employees. In social network terms, such a pair is called a *dyad*. Within this framework, each email message constitutes a *relational event* characterized by the *sender* (s), who initiates the action (i.e., who sends the email); the *receiver* (r), to whom the action is targeted (i.e., who receives the email); and *time* (t), the exact time point at which the relational event occurs. At each time point in the sequence, 153 potential senders can send an email to 152 potential receivers (excluding email messages people send to themselves), which means that at any point in time $153 \times 152 = 23256$ email dyads that can potentially occur. The aim of the analysis is to model who sends an email message to whom at what point in time over the course of 1 year. Mathematically, the joint probability to model the whole sequence of emails is similar to the well-known event history model or survival model (Cox, 1972; Lawless, 2002).

Predictor variable	Description
ShortInertia	The number of messages a potential sender sent to a potential receiver in the last 30 days
LongInertia	The number of messages a potential sender sent to a potential receiver more than 30 days ago
ShortReciprocity	The number of messages a potential sender received from a potential receiver in the last 30 days
LongReciprocity	The number of messages a potential sender received from a potential receiver more than 30 days ago
SameBuilding	A binary variable which indicates whether potential sender and potential receiver work in the same building (1) or not (0)
DiffHierarchy	The hierarchical difference between the sender and receiver on a scale from 1 to 9
LogSenderTenure	The number of years a potential sender works in the organization on a log scale

Table 2.2: Variables: predictor variables and their interpretations.

In the REM, we model the rate at which an email is sent from a given sender to a given receiver at a given point in time as a log-linear model that (apart from the exponent that occurs in the equation) resembles the well-known linear regression structure. The model then takes into account every possible sender, every possible receiver, and every possible point in time, for the entire observation period. One of our substantive interests in this study is whether the emailing rates of employees depend only (or mainly) on the recent email interactions of the employees or whether they also take into account email exchanges that happened longer ago. This is important for STRATSERV, as it shows how long the effects of interventions last. If it turns out that employees mainly respond to innovation-related messages they received recently, and much less to messages received or exchanged longer ago, this is a sign that employees apparently need to be “reminded” of innovation constantly and that it has not become a routine part of their conversations.

In particular, we will investigate this for inertia and reciprocity (see Table 2.2).

Variable	Model 1				Model 2			
	$\hat{\beta}$	se($\hat{\beta}$)	z-value	p-value	$\hat{\beta}$	se($\hat{\beta}$)	z-value	p-value
Intercept	-11.6322	0.0862	-134.914	0.000	-9.2559	0.0249	-371.323	0.000
ShortInertia	0.0831	0.0005	151.582	0.000	0.0869	0.0005	181.294	0.000
LongInertia	0.0058	0.0005	10.871	0.000	0.0065	0.0005	14.025	0.000
ShortReciprocity	0.0484	0.0104	4.628	0.000	0.0345	0.0101	3.406	0.0006
LongReciprocity	-0.0070	0.0170	-0.409	0.682	-0.0094	0.0162	-0.579	0.563
SameBuilding	0.9854	0.0401	24.591	0.000				
DiffHierarchy	-0.3003	0.0096	-31.307	0.000				
LogSenderTenure	0.9234	0.0378	24.413	0.000				
AIC		16004.33				16981.15		
BIC		16045.93				17007.15		

Table 2.3: Model 1 and Model 2: maximum likelihood estimates, standard errors, z-values, p-values, AIC and BIC.

In order to accomplish this, both the inertia and reciprocity statistics are calculated according to two different event history lengths. For both statistics we include into the model a short-run version where only past events that occurred *until 30 days* before the time of the email are included (*recent past*) and a long-run version that includes the past events that occurred *more than 30 days* before the email was sent (*less recent past*) (cf. Quintane et al. (2013)). A complete description of the variables used in our analysis can be found in Table (2.2).

2.3.3 Model comparison

We estimate two models: in Model 1, all the variables in Table (2.2) are embedded in the log-linear predictor; in Model 2, only the short-run and long-run versions of inertia and reciprocity are included. Via this model comparison, we can learn whether a simpler model without exogenous effects may be enough for a good fit for the data. Considering the specification of Model 1, the email

Chapter 2

rate (λ) at time t_m for the dyad (sender, receiver) = (Marco, Jane) is,

$$\begin{aligned}
 \lambda(t_m, \text{Marco, Jane}) = \exp \{ & \beta_{\text{Intercept}} + \\
 & + \beta_{\text{ShortInertia}} \text{ShortInertia}(t_m, \text{Marco, Jane}) + \\
 & + \beta_{\text{LongInertia}} \text{LongInertia}(t_m, \text{Marco, Jane}) + \\
 & + \beta_{\text{ShortReciprocity}} \text{ShortReciprocity}(t_m, \text{Marco, Jane}) + \\
 & + \beta_{\text{LongReciprocity}} \text{LongReciprocity}(t_m, \text{Marco, Jane}) + \\
 & + \beta_{\text{SameBuilding}} \text{SameBuilding}(\text{Marco, Jane}) + \\
 & + \beta_{\text{DiffHierarchy}} \text{DiffHierarchy}(\text{Marco, Jane}) + \\
 & + \beta_{\text{LogSenderTenure}} \text{LogSenderTenure}(\text{Marco}) \}
 \end{aligned} \tag{2.1}$$

where

$$\boldsymbol{\beta} = \{ \beta_{\text{Intercept}}, \beta_{\text{ShortInertia}}, \beta_{\text{LongInertia}}, \beta_{\text{ShortReciprocity}}, \\
 \beta_{\text{LongReciprocity}}, \beta_{\text{SameBuilding}}, \beta_{\text{DiffHierarchy}}, \beta_{\text{LogSenderTenure}} \}$$

is the vector of effects describing the impact of statistics on the rate of occurrence of an email being sent from a sender to a receiver. Positive effects (negative effects) imply that as the statistic increases in value, it increases (decreases) the email rate. As regards Model 2, the rate of an email sent from Marco to Jane at time t_m becomes,

$$\begin{aligned}
 \lambda(t_m, \text{Marco, Jane}) = \exp \{ & \beta_{\text{Intercept}} + \\
 & + \beta_{\text{ShortInertia}} \text{ShortInertia}(t_m, \text{Marco, Jane}) + \\
 & + \beta_{\text{LongInertia}} \text{LongInertia}(t_m, \text{Marco, Jane}) + \\
 & + \beta_{\text{ShortReciprocity}} \text{ShortReciprocity}(t_m, \text{Marco, Jane}) + \\
 & + \beta_{\text{LongReciprocity}} \text{LongReciprocity}(t_m, \text{Marco, Jane}) \}
 \end{aligned} \tag{2.2}$$

The results of both models can be found in Table [2.3](#). Model 1 seems to be better supported by data since the BIC and AIC for Model 1 are lower than for Model 2. In addition to this, email rate is mainly affected by recent email history, i.e., by the short-run effects of inertia and reciprocity. Although the effect of long-run inertia (LongInertia) is statistically significant, the effects of long-run inertia and long-run reciprocity (LongReciprocity) are negligibly small and hence barely affect the email rate. The results of Model 2 (which only includes inertia and reciprocity) show that these effects are stable and unaffected by the other

statistics. In other words, the employees tend to repeat their recent behavior and mainly respond to innovation-related messages received in the recent past, while innovation messages that were sent or received more than 30 days ago seem to no longer affect emailing behavior today. In other words, employees appear to discuss innovation because it is what they recently discussed, not because it something that is on their minds in the long run. This is a sign that STRATSERV has not been able to make innovation an integral part of their employees' mindset.

From Model, 1 we see that employees send emails at lower rates to other employees who are lower in the organizational hierarchy than they are themselves and send their email messages at higher rates to those who have higher hierarchy levels than they have themselves ($\hat{\beta}_{\text{DiffHierarchy}} = -0.3003$). In other words, email messages about innovation are more readily sent up the organizational hierarchy than down. This is consistent with the idea that the STRATSERV employees are willing to inform their superior about potential innovation but are less likely to put their ideas into action themselves by discussing it with those lower in the chain of command. Conversely, employees who enjoy higher hierarchical positions are more popular targets for such email messages than are those who occupy low status positions in the organization. Again, innovation discussion is directed up the chain, but much less to the lower levels.

Except for DiffHierarchy, all other statistics in Model 1 have positive effects on the emailing rates. For instance, the email rate of a sender to a receiver who works in the same building (SameBuilding = 1) is around two and a half times ($\exp\{\hat{\beta}_{\text{SameBuilding}}\} = 2.679$) higher than the email rate from that same sender to a colleague working in a different building, holding constant all the other variables. This is an important finding, as it suggests that physical boundaries (i.e. working in a different building) also appear to function as communication boundaries: STRATSERV employees more intensely discuss innovation-related topic with those whom they routinely meet at the coffee machine, and much less with those they don't run into that often.

We also observe that the rate at which employees send innovation-related email increases with the time they have been at the organization. Conversely, newcomers and juniors turn out less active in communicating about innovation than are the seniors of the firm, which makes sense.

2.4 Discussion and Conclusion

The relative importance of the different effects can be used to improve and optimize information sharing. For example, as there is a large positive (negative) effect of interaction when employees work in the same (in different) buildings, interaction may be greatly improved by setting up interventions in the organizations that stimulate discussions across employees in different buildings. In addition, it is important to know for managers that STRATSERV's employees are less likely to share innovation-related communication with colleagues they are no co-located with. Although this can partly be addressed by strategically placing employees in their various locations, it is also important for managers to realize where communication may flow more easily and where it is likely to be hampered.

Furthermore, STRATSERV learns from this analysts that a temporary silence in innovation-related activity tends to remove the topic from the active attention of its employees. This could potentially be addressed by organizing activities around innovation, but it also signals that the current activities haven't been successful in making innovation part of the normal conversation of STRATSERV's employees. This may be a reason to reevaluate the effectiveness of the current strategy while, at the same time, taking into account that it may take a long time to establish an innovation mindset.

Thanks to the relational event model we are able to understand which factors play regarding employee interaction. Specifically, the observed differences in intensities and signs of the relative effects showed that certain characteristics can impact the email rate to different degrees and in different directions. Using targeted interventions, these insights can be used to reach more employees in a shorter amount of time. For further reading on relational event models, we refer interested readers to R. T. A. J. Leenders et al. (2016), Pilny et al. (2016), and Schecter et al. (2018a).

A BAYESIAN SEMI-PARAMETRIC APPROACH FOR MODELING MEMORY DECAY IN DYNAMIC SOCIAL NETWORKS

3

Abstract

In relational event networks, the tendency for actors to interact with each other depends greatly on the past interactions between the actors in a social network. Both the volume of past interactions and the time that has elapsed since the past interactions affect the actors' decision-making to interact with other actors in the network. Recently occurred events may have a stronger influence on current interaction behavior than past events that occurred a long time ago—a phenomenon known as “memory decay”. Previous studies either predefined a short-run and long-run memory or fixed a parametric exponential memory decay using a predefined half-life period. In real-life relational event networks, however, it is generally unknown how the influence of past events fades as time goes by. For this reason, it is not recommendable to fix memory decay in an ad-hoc manner, but instead we should learn the shape of memory decay from the observed data. In this chapter, a novel semi-parametric approach based on Bayesian Model Averaging is proposed for learning the shape of the memory decay without requiring any parametric assumptions. The method is applied to relational event history data among socio-political actors in India and a comparison with other relational event models based on predefined memory decays is provided.

3.1 Introduction

As a result of the growing automated collection of information, fine-grained longitudinal network data are increasingly available in many disciplines, such as sociology, psychology, and biology. These data have the potential to revolutionize our understanding about complex social network dynamics as we can learn how the past affects the future, how interaction behavior changes in continuous time, and how past social interactions lose their influence on the future away as time progresses. This has inspired social network scientists to develop network models that suit the inherent dynamic nature of these so-called *relational event* data. A relational event is defined as an action initiated by a sender and targeted to one or more receivers at a specific point in time. The relational event modeling framework aims to model the *event rate*: the speed at which relational events occur over a period of time between the actors in the model. The event rate can be expressed as a function of characteristics that quantify endogenous network patterns or exogenous characteristics that (jointly) determine how the network unfolds at some point in time (Butts, 2008). In sociological and psychological research, the application of these relational event models aims to find behavioral patterns and to shed light on the emergence of a global structure from network dynamics occurring at a local (typically, dyadic) level (R. T. A. J. Leenders et al., 2016; Pilny et al., 2016; Schechter et al., 2018b).

Of particular interest is to understand what triggers actors to interact with each other. Actors might decide which mutual recipient to target their actions to depending on various aspects such as homophily, norms of reciprocity, the volume of past social interactions, triadic closure mechanisms, et cetera (Rivera et al., 2010). Past relational events influence future events in different ways. First, qualitative aspects of the past events play a role, such as whether the interaction was positive or a negative or who was the sender of the past event. For example, receiving a message from the company's president might have a greater effect than getting a message from a regular colleague. Similarly, the valence of events may play a role: events with a negative connotation have been argued to have a greater effect than events with a positive connotation (Brass & Labianca, 1999; Labianca & Brass, 2006; Moerbeek & Need, 2003; Offer, 2021). Second, recent past events are generally expected to have a greater influence on the present than events that occurred a long time ago (Brandes et al., 2009; Butts, 2008; Mulder & Leenders, 2019; Quintane et al., 2013). Having recently

received praise from a colleague is likely to affect current interaction more than if that praise dates back a year ago.

While studies using relational event data tend to focus on the effects of endogenous statistics (e.g., to what extent actors repeat their past interactions, do they reciprocate interactions aimed at them, or do they prefer to interact with others with whom they share many other interaction partners with?) or exogenous statistics (e.g., does information sharing tend to go from lower-status actors to higher status actors, do friends share information at higher rates than non-friends, how much does co-location matter for communication in IT-enabled teams?), much less attention has been paid to exactly how long past events retain their influence on the present and future. This is the very subject of this chapter. In particular, our aim is to derive a method that allows a researcher to empirically derive the shape of the function by which past events lose their influence on the future. This shape can be linear, exponentially decaying, or have any other shape. To unify our terminology, we will use the term “memory decay” for this phenomenon, even though we do not aim to model cognitive functions of the actors in the network. This terminology is not new. For example, Brandes et al. (2009) specify a half-life function that governs the decaying influence of events “motivated by the assumption that actors forget (or forgive)”. Similarly, Mulder and Leenders (2019) and R. T. A. J. Leenders et al. (2016) explicitly refer to this phenomenon as “memory decay.” Within the context of Temporal ERGM’s, Leifeld et al. (2018) and Leifeld and Cranmer (2019) include so-called “memory terms” and allow the researcher to specify time-based functions (“time trends”) of how the time since a past tie affects the occurrence of later ties.

Our focus is on the way the influence of past events on the future changes, that is akin to how long people “remember” (or care about) the past actively enough to still make it count towards the present and future. Because the effect of the past will almost always decrease as time passes, we will use the term “memory decay” throughout this chapter to refer to the shape of the function that captures how the influence of a past event on future events changes as the time since the event increases.

Already in Butts (2008) seminal paper and the accompanying software (Butts, 2021), the importance of memory retention of past relational events is highlighted. So-called “participation shifts” were introduced that capture how the interaction dynamics shifts between dyads depending on the very last event that happened. These statistics assume that actors respond to the immediate

past, regardless of what happened before that. In addition, a “recency” statistic is considered where the potential receivers for each potential sender are ordered based on their recent activity and a power-law is used to create a predictor variable (i.e., the reciprocal of the rank). This mechanism captures the extent to which actors take into account the last events they had with every other actor, discounting events from farther into the past. Finally, other endogenous statistics (such as inertia and reciprocity) are computed as the total volume of past interactions between actors and, hence, count all past events as equally important to the future and assume that no past event, however distant in the past, is ever forgotten. In sum, these statistics already capture three distinct ways in which the past is (dis)counted towards the present and the future and each reflect a different shape of memory decay.

More recently, other approaches have also been considered to better understand how (long) past activity affects future events. One approach has been to quantify a specific pattern of interactions according to specific predefined time intervals, such as a *short-run* expression (calculated by considering recently passed events) and a *long-run* expression (considering long-passed events in the computation) (Kitts et al., 2017; Patison et al., 2015; Perry & Wolfe, 2013; Quintane & Carnabuci, 2016; Quintane et al., 2013). The estimated effects for these intervals describe how different the impact of the specific pattern is on the event rate according to different recency of events constituting the pattern itself. Another approach consists of estimating the model while using a moving time window with a predefined fixed memory length with the result of a trend of the effects over the windows (Mulder & Leenders, 2019). An alternative to time-intervals-based methods weighs the influence of past events by an exponentially decreasing function with a given *half-life* parameter that describes the elapsed time beyond which the influence of an event in the calculation of the statistic is halved (Brandes et al., 2009; R. T. A. J. Leenders et al., 2016; Lerner et al., 2013).

In all of these approaches, a researcher needs to predefine the *memory lengths* for the discretized model or predefine the *steepness* of the decay in the case of the continuous half-life model. Typically, heuristic considerations are used to specify this function. Notable exceptions include Brandenberger (2018a) and Brandes et al. (2009) who explored the fit and robustness of the results by considering different choices for the half-life parameter. The question is, however, whether a prespecified memory decay appropriately captures the dependence

between the time that has passed since the event and the current event. Depending on the context, certain decay shapes may be more suitable in terms of fit than other shapes. Model misfit may result in poor predictions and unreliable inferences.

Considering the dearth of time-sensitive theory to draw from (cf. Ancona et al. (2001), Cronin et al. (2011), and R. T. A. J. Leenders et al. (2016)), there is little theory (if any) to truly guide a researcher in the choice of an appropriate memory decay function for a research project at hand. Researchers have dealt with this by specifying choices for the decay function based on their experience with the empirical context or based on their own assumptions regarding the influence of time. Alternatively, an approach that we propose in this chapter is to present a semi-parametric method for learning the actual shape of memory decay in relational event models. The method is semi-parametric in the sense that it does not make assumptions about a specific functional form for memory decay. Indeed, parameters that potentially govern the memory process and, in turn, determine its shape over time are often unknown and our intent is to minimize the challenge that is involved in prespecifying a memory function by a researcher. Our method can be used for finding any functional form of memory decay which could be an exponentially decreasing trend, a smoothed step-wise function, or other, possibly more (or less) complex, functional trends. Our semi-parametric method combines the relational event modeling framework (as in Butts (2008)) with Bayesian inference in the context of a model selection problem (Bayesian Model Averaging) (Volinsky et al., 1999). The idea is to consider a large “bag” of step-wise models with different interval configurations. Next, the fit is computed for all step-wise models, and subsequently, we model the shape as an average of these models weighted according to their respective fit to the observed data.

The chapter is structured as follows. In Section 2, we introduce the relational modeling framework along with the concept of memory decay. In Section 3, we formulate a step-wise memory decay model. In Section 4, we present a continuous memory decay model and highlight the potential use of step-wise models in approximating the continuous shape of the decay. In Section 5, we present a semi-parametric method based on a Bayesian Model Averaging along with two weighting systems for generating random draws from the posterior memory decay. In Section 6 we apply the method to empirical data and we compare it to other models that predefine parametric memory decays. Concluding the chap-

ter, in Section 7 we discuss some considerations regarding the methodology and potential further development.

3.2 Relational event models that capture memory decay

In the relational event framework (Butts, 2008), a relational event e_m is characterized by the 3-tuple (s_{e_m}, r_{e_m}, t_m) , respectively sender, receiver, and time of occurrence of the event. The joint probability of the realized ordered sequence of M relational events, $E_{t_M} = (e_1, \dots, e_M)$, can be modeled as

$$p(E_{t_M}; \beta) = \prod_{m=1}^M \left[\lambda(s_{e_m}, r_{e_m}, X_{e_m}, E_{t_{m-1}}, \beta) \prod_{e' \in \mathcal{R}} \exp \left\{ -\lambda(s_{e'}, r_{e'}, X_{e'}, E_{t_{m-1}}, \beta) (t_m - t_{m-1}) \right\} \right] \quad (3.1)$$

where t_0 (at $m = 1$) is assumed to be equal to zero or to the starting time point of the case study. Further, $\lambda(s_{e_m}, r_{e_m}, X_{e_m}, E_{t_{m-1}}, \beta)$ is the rate of the event e_m occurred at time t_m and $\lambda(s_{e'}, r_{e'}, X_{e'}, E_{t_{m-1}}, \beta)$ represents the event rate of any event e' that could have happened at time t_m (including e_m). Indeed, e' belongs to the risk set \mathcal{R} consisting of all sender/receiver combinations $S \times R$: where S and R are, respectively, sets of all possible senders and receivers for the entire event sequence. If all actors can be senders as well as receivers in an interaction, then $S \equiv R$ and the set of actors is simply referred to as S . Equation (3.1) can be viewed as the well-known survival model with time-varying covariates, where hazard and survival components form the likelihood in the same way (Lawless, 2002).

The *rate* of the specific dyadic event $e' \in \mathcal{R}$ at a generic time t_m is modeled as a log-linear function of statistics as follows

$$\lambda(s_{e'}, r_{e'}, X_{e'}, E_{t_{m-1}}, \beta) = \exp \left\{ \sum_{p=1}^P \beta_p u_p(s_{e'}, r_{e'}, X_{e'}, E_{t_{m-1}}) \right\} \quad (3.2)$$

where:

- β_p with $p = 1, \dots, P$, are parameters describing the effects of statistics on the logarithm of the event rate;

- $X_{e'}$ is the set of covariates (exogenous attributes, possibly time-varying) associated with event e' ;
- $E_{t_{m-1}}$ refers to the collection of all of those events that occurred before t_m ;
- $u_p(s_{e'}, r_{e'}, X_{e'}, E_{t_{m-1}})$ with $p = 1, \dots, P$, are the statistics of interest and each one can depend either on transpired events (endogenous statistics calculated for all the dyads at each time point and given $E_{t_{m-1}}$) or on exogenous attributes ($X_{e'}$).

In the standard specification of the model, endogenous statistics describe patterns of interactions occurring in the network that are quantified at each time point by considering the whole history of events that happened from the initial state of the network (i.e., the first observed relational event) until the time point before the current one (i.e., t_{m-1} in (3.2)). For instance, consider the standard formulation of the inertia statistic, which is a dyadic endogenous statistic that quantifies the volume of interactions of a specific dyad that occurred until the current time point. Inertia quantifies the extent to which specific relational events keep repeating over time. The corresponding formula at a generic time point t_m with history $E_{t_{m-1}}$ is

$$\text{inertia}(i, j, t_m) = \sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(i, j) \quad (3.3)$$

where $\mathbb{I}_e(i, j)$ is the indicator variable that assumes value 1 if the event $e \in E_{t_{m-1}}$ has $s_e = i$ and $r_e = j$, 0 otherwise. The event rate for any possible event $e' \in \mathcal{R}$ at time t_m with only the inertia in the linear predictor can be written as

$$\lambda(s_{e'}, r_{e'}, E_{t_{m-1}}, \beta) = \exp \{ \beta_{\text{inertia}} \text{inertia}(s_{e'}, r_{e'}, E_{t_{m-1}}) \} \quad (3.4)$$

A positive estimate for β_{inertia} reflects that actors interact at higher rates with those actors who were often receivers of their past interactions. This is a sign of social routinization: what happened in the past is bound to be repeated over and over into the future. For instance, consider Figure 3.1 where a sequence of events from t_1 to t_{14} is represented on a time line. In order to calculate the inertia at time t_{15} for the specific dyad (i, j) we need to count the number of past events in the history $E_{t_{14}}$ where i targeted an action to j , which is six in the

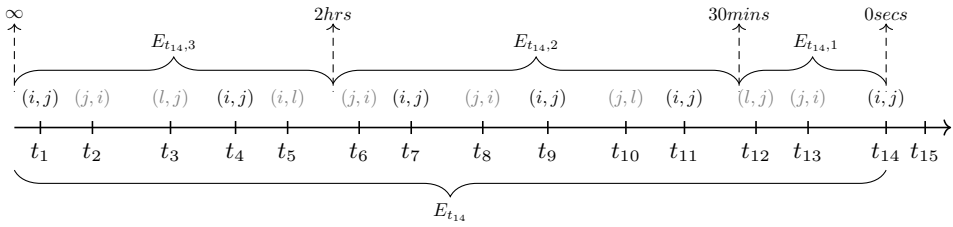


Figure 3.1: Example of the calculation of *Inertia* for the dyadic event (i, j) , given the history of events $E_{t_{14}} = \{e_{t_1}, \dots, e_{t_{14}}\}$. The event of interest in the calculation of the statistic is written in black, others are gray. Without considering intervals, the value of inertia at time t_{15} is 6: the total count of (i, j) events already occurred. When considering intervals (time bounds of each interval of the event history are highlighted by upwards arrows and labeled as $E_{t_{14},1}$, $E_{t_{14},2}$ and $E_{t_{14},3}$), the value of inertia across the three intervals becomes $\text{inertia}_1(i, j, t_{15}) = 1$, $\text{inertia}_2(i, j, t_{15}) = 3$, and $\text{inertia}_3(i, j, t_{15}) = 2$, where each one corresponds to the number of times that the event (i, j) is observed within each interval.

example. Although this approach would give insights into how previous interactions between actors have influence on the event rate, we would be assuming long-passed events (such as those that happened 14 and 11 events ago, over two hours ago) to be equally influential as recent ones (such as the events that are only 1 or 4 events–or 45 minutes or so–old) in the computation of the statistics as well as on the event rate itself. This assumption may not be realistic for relational event data in practice as indicated earlier. Hence, our objective is to specify a model that is capable of accounting for this mutable effect of past events on the dyadic event rate.

3.3 A step-wise memory decay model

3.3.1 Step-wise decay for first-order endogenous effects

As a first step, we model the relative importance of past events as a function of the transpired time since the event was observed using a discretized, step-wise memory decay model (Perry & Wolfe, 2013). After the transpired time is divided into fixed intervals, endogenous statistics are computed for each interval and the corresponding endogenous effects are estimated. These effects quantify the relative importance of past events in predicting future events. For instance, considering the event sequence in Figure 3.1, we observe that at t_{15} more than two hours have transpired since the starting time point and we divide the history

of events $E_{t_{14}}$ into three sub-histories according to a set γ of increasing time lengths, for example, $\gamma = (0secs, 30mins, 2hrs, \infty)$

$$\begin{aligned} E_{t_{14},1} &= \{e \in E_{t_{14}} : (t_{15} - t_e) \in (0secs, 30mins]\} \\ E_{t_{14},2} &= \{e \in E_{t_{14}} : (t_{15} - t_e) \in (30mins, 2hrs]\} \\ E_{t_{14},3} &= \{e \in E_{t_{14}} : (t_{15} - t_e) \in (2hrs, \infty)\} \end{aligned} \quad (3.5)$$

where the first sub-history $E_{t_{14},1}$ contains all events transpired until 30 minutes before t_{15} ; the second, $E_{t_{14},2}$, includes those events happened between 30 minutes and 2 hours before t_{15} ; lastly, the third sub-history, $E_{t_{14},3}$, includes all events happened more than 2 hours before t_{15} (the right bound is left undefined here). In Figure 3.1, the partition into sub-histories is shown by the upwards arrows corresponding to the time lengths γ .

Therefore, three values of inertia can be calculated at any time point t_m in the observed sequence by considering the three different partitions of the event history according to the increasing time lengths (γ).

$$\text{inertia}_k(i, j, t_m) = \sum_{e \in E_{t_{m-1},k}} \mathbb{I}_e(i, j) \quad \text{with } k = 1, 2, 3 \quad (3.6)$$

Following the example in Figure 3.1, corresponding values of inertia according to intervals at time point t_{15} are: $\text{inertia}_1(i, j, t_{15}) = 1$, $\text{inertia}_2(i, j, t_{15}) = 3$ and $\text{inertia}_3(i, j, t_{15}) = 2$. We may expect that events that occurred in $E_{t_{14},1}$ have a larger impact on the event rate than those occurring in $E_{t_{14},2}$ and $E_{t_{14},3}$. Although we do not make this assumption (as the goal is to learn from the data), the estimated effects relative to the three statistics will generally decrease in actual data, making the regression coefficient for inertia based on the most recent sub-history higher than that of inertia based on the most distant events, that is $\beta_{\text{inertia}_1} > \beta_{\text{inertia}_2} > \beta_{\text{inertia}_3}$.

In a more general case where K partitions of the current event history are defined according to increasing time lengths, such as

$$\gamma = (\gamma_0, \gamma_1, \dots, \gamma_K) \quad \text{with } 0 = \gamma_0 < \gamma_1 < \dots < \gamma_K = \infty \quad (3.7)$$

we can partition the event history $E_{t_{m-1}}$ at time t_m into subsets as

$$\begin{aligned}
 E_{t_{m-1},1} &= \{e \in E_{t_{m-1}} : \gamma_e(t_m) \in (0, \gamma_1]\} \\
 E_{t_{m-1},2} &= \{e \in E_{t_{m-1}} : \gamma_e(t_m) \in (\gamma_1, \gamma_2]\} \\
 &\vdots \\
 E_{t_{m-1},K} &= \{e \in E_{t_{m-1}} : \gamma_e(t_m) \in (\gamma_{K-1}, \infty)\}
 \end{aligned} \tag{3.8}$$

where $\gamma_e(t_m) = t_m - t_e$ represents the elapsed time at t_m since the past event $e \in E_{t_{m-1}}$. The general formula for inertia relative to the dyadic event e with $(s_e = i, r_e = j)$ in the k -th partition of the $E_{t_{m-1}}$ at time t_m is

$$\text{inertia}_k(i, j, t_m) = \sum_{e \in E_{t_{m-1},k}} \mathbb{I}_e(i, j) \quad \text{with } k = 1, \dots, K \tag{3.9}$$

The event rate for any possible event $e' \in \mathcal{R}$ at time t_m where inertia is defined across K partitions is

$$\lambda(s_{e'}, r_{e'}, E_{t_{m-1}}, \beta) = \exp \left\{ \sum_{k=1}^K \beta_{\text{inertia}_k} \text{inertia}_k(s_{e'}, r_{e'}, t_m) \right\} \tag{3.10}$$

Once statistics are calculated across the K partitions, their corresponding parameters β_{inertia_k} , with $k = 1, \dots, K$, can be estimated using the likelihood function in (B.1). In the interval case for the inertia, parameters express how the propensity of actors to target their actions to the same past receivers changes as a function of the recency of past events.

The use of interval statistics according to K partitions of the event history directly relates to the dynamic of the estimated effects and their evolution will follow a step function as in Figure B.2 with a mathematical function as in (B.11), that is based on the time lengths γ used to create the partitions:

$$\beta_{\text{inertia}}(\gamma) = \begin{cases} \beta_{\text{inertia}_1} & \text{if } \gamma \in (\gamma_0, \gamma_1] \\ \vdots & \\ \beta_{\text{inertia}_K} & \text{if } \gamma \in (\gamma_{K-1}, \gamma_K] \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

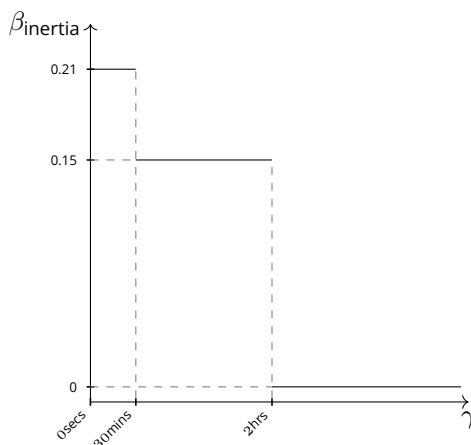


Figure 3.2: step-wise function for the effect of *Inertia* on the event rate. The function defines three intervals of the elapsed time γ (on the x-axis): the first interval $\gamma \in (0\text{secs}, 30\text{mins}]$, the second interval $\gamma \in (30\text{mins}, 2\text{hrs}]$ and the third interval $\gamma \in (2\text{hrs}, +\infty)$. The y-axis shows the value of the effect $\beta_{inertia}$ for each interval.

Step-wise memory effects can also be modeled for other first-order endogenous statistics such as reciprocity, sender/receiver-in/out-degree whose formulas can be found in Appendix [A.1](#).

3.3.2 Step-wise decay for higher order endogenous effects

Besides statistics that are based only on past interactions within a given dyad, the effects of higher order statistics involving more than two actors, can be used as well within this approach. Higher order endogenous statistics are characterized by more than one dyadic relational event in their formula. As such, the behavioral pattern of interest is more complex substantively as well as its computation. Indeed, in the case of triadic statistics, as with transitivity, the computation consists in the quantification of the number of times a dyad could potentially close a particular triangular structure if it occurred as next interaction after a specific sequence of past events.

Figure [B.3](#) describes the pattern of the transitivity closure (Schechter et al., [2018b](#)) in the context of relational event data where interactions are time-ordered. The search for specific behavioral patterns can be improved by introducing such time-ordering in the calculation of the statistics. Specifically for

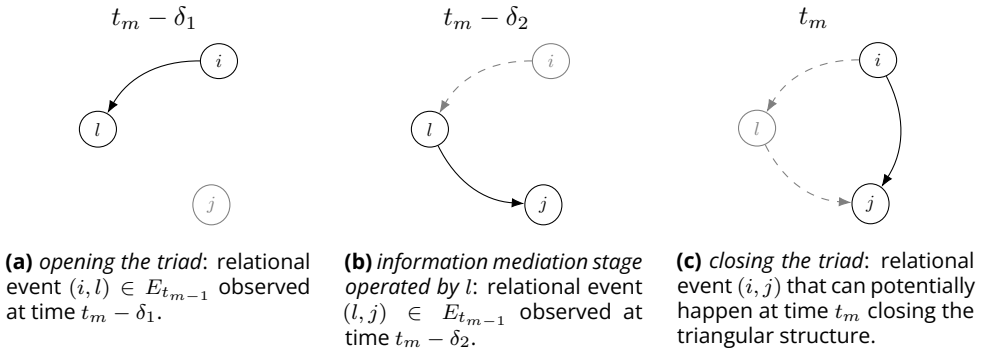


Figure 3.3: Figures from left to right describe the pattern of the transitivity closure in three time-framed steps. The time order of the three steps is described on the top of each graph, and it goes from the left, where the event (i, l) opens the potential triad at $t_m - \delta_1$, to the right, where the last event (i, j) closes the triad at t_m . Therefore, given the event history $E_{t_m - 1}$, the possible event (i, j) occurring at t_m (3.3c) can close a triad already opened with a third actor (l in the example) who acts as a broker in the process of information sharing/mediation. Events (i, l) (3.3a) and (l, j) (3.3b) occur by following the time order in the example, with δ_1 and δ_2 at time t_m being the transpired times since the two events (i, l) and (l, j) , such that $t_m - \delta_1 < t_m - \delta_2$ and $0 \leq \delta_2 < \delta_1 < t_m$. Therefore, in this formulation the time order of the occurrence of events characterizing the triangular structure is taken into account. Gray nodes and dashed gray arrows indicate, respectively, inactive actors and events already occurred, whereas active actors and the occurring dyadic event are in black.

transitivity closure, the following formula computes the statistic for the dyad (i, j) at time t_m :

$$\text{transitivity closure}(i, j, t_m) = \sum_{l \in \mathcal{S} \setminus \{i, j\}} \sum_{e \in E_{t_m - 1}} \sum_{\substack{e^* \in E_{t_m - 1}: \\ t_{e^*} \in [t_e - \gamma_e(t_m), t_e]}} \mathbb{I}_e(l, j) \mathbb{I}_{e^*}(i, l) \quad (3.12)$$

where:

- $\mathbb{I}_e(l, j)$ is the indicator variable that assumes value 1 if the event $e \in E_{t_m - 1}$ has $s_e = l$ and $r_e = j$, and value 0 otherwise (the same reasoning applies to the other indicator variables in (3.12));
- e and e^* are any pair of events belonging to the event history $E_{t_m - 1}$ such that $t_{e^*} < t_e$;
- $\gamma_e(t_m) = t_m - t_e$ is the time transpired at t_m since the event $e \in E_{t_m - 1}$.

Figure 3.4 shows an example of the formula in (3.12) for just one $l \in \mathcal{S} \setminus \{i, j\}$

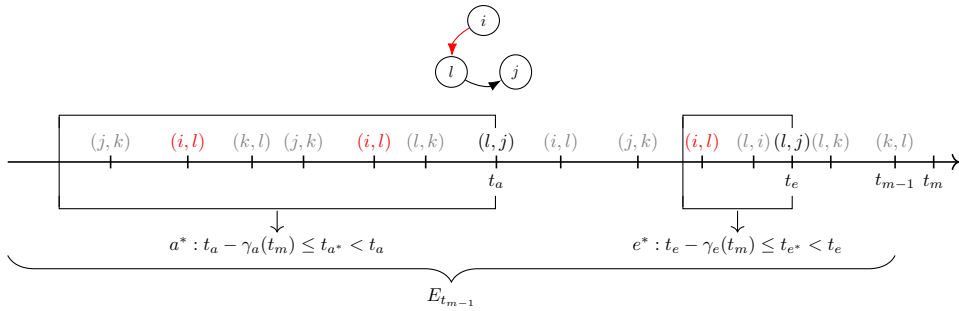


Figure 3.4: Example of the calculation of transitivity at t_m for the dyad (i, j) and information mediator l : the event history $E_{t_{m-1}}$ counts only two events (l, j) , at time t_e and t_a . In order to quantify the contribute of l to the transitivity (i, j, t_m) we : (i) find the second-last event in the pattern, that is (l, j) , two in the example at t_e and t_a (in black); (ii) consider the backward intervals $[t_a - \gamma_a(t_m), t_a)$ and $[t_e - \gamma_e(t_m), t_e)$ (black squares in the figure); (iii) for each interval quantify the number of (i, l) observed (in red). In the example, the contribution of events a and e to the statistic is, respectively, 2 (because two events (i, l) are observed in the backward interval of t_a) and 1 (because one event (i, l) is observed in the backward interval of t_e). Thus, the value of transitivity for (i, j) at t_m with mediator l is given by their sum, that is 3: if (i, j) is the next event to occur it is going to close three potential triads where the information mediator was l .

at time t_m , with a history of events $E_{t_{m-1}}$. In the example, two dyadic events (l, j) , noted as e and a , occurred at t_e and t_a before t_m . For each of them we seek backward for those events e^* and a^* that occurred within intervals based on the transpired time of e ($\gamma_e(t_m) = t_m - t_e$) and a ($\gamma_a(t_m) = t_m - t_a$) that are respectively $[t_e - \gamma_e(t_m), t_e)$ and $[t_a - \gamma_a(t_m), t_a)$. Hence, if any event e^* or a^* in these intervals has sender i and receiver l then the product of the two indicator variables in (3.12) will be one and so will be contribute to the sum, and is zero otherwise. In the specific example, as to event a we observe two dyadic events (i, l) that happened in $[t_a - \gamma_a(t_m), t_a)$, whereas for e we find just one event (i, l) that occurred in $[t_e - \gamma_e(t_m), t_e)$. Therefore, if the dyad (i, j) is going to occur at t_m it would close at least three potential triangular structures (of the type described in Figure 3.3) where the actor l is the information mediator. The quantification in Figure 3.4 is just a simple example where the calculation of the transitivity is performed only in the case where the specific actor l is the mediator (with l being a different actor from i and j). To quantify the transitivity closure for the dyad (i, j) , which describes the total number of triangular structures closed by the occurrence of (i, j) at t_m , we have to sum all the potential triads that could be closed considering all the possible $N - 2$ information mediators. This is described in formula (3.12) by the outer sum across all the actors in the network

excluding i and j ($S \setminus \{i, j\}$) and indexed by l . The new formula for transitivity closure accounts for the time order of events in the triadic behavioral pattern and assumes that those events (i, l) happened earlier than an event (l, j) and will count in the formula if and only if they transpired within the same time span of the specific (l, j) .

The event rate for any possible event $e' \in \mathcal{R}$ at time t_m with only the transitivity in the linear predictor is written as

$$\lambda(s_{e'}, r_{e'}, E_{t_{m-1}}, \beta) = \exp \{ \beta_{\text{transitivity}} \text{transitivity}(s_{e'}, r_{e'}, E_{t_{m-1}}) \} \quad (3.13)$$

A positive $\beta_{\text{transitivity}}$ means that the more partners $s_{e'}$ and $r_{e'}$ had in common in the past the more likely $s_{e'}$ will choose $r_{e'}$ as receiver of its next interaction. Vice versa, when $\beta_{\text{transitivity}} < 0$, the rate of the event e' lowers, meaning that there is a tendency by actors to discourage closure and thus to engage in fewer interactions with those actors they had shared a partner with. The statistic in (3.12) refers to the event history $E_{t_{m-1}}$, that is the entire sequence of events since the onset until t_{m-1} (including $e_{t_{m-1}}$). The $\beta_{\text{transitivity}}$ may depend on how recently the event (l, j) occurred. Thus, transitivity can be redefined across intervals in the same way as inertia in Section 3.3.1.

Consider the more general case of K partitions of the current event history (as in (3.8)) according to $K + 1$ increasing time lengths γ (as in (3.7)). The transitivity as regards the k -th interval, for the dyad (i, j) at time t_m will be,

$$\text{transitivity}_k(i, j, t_m) = \sum_{l \in S \setminus \{i, j\}} \sum_{e \in E_{t_{m-1}, k}} \sum_{\substack{e^* \in E_{t_{m-1}}: \\ t_{e^*} \in [t_e - \gamma_e(t_m), t_e]}} \mathbb{I}_e(l, j) \mathbb{I}_{e^*}(i, l) \quad (3.14)$$

where the quantification of potential triads is divided through the K intervals of the history $E_{t_{m-1}} = \{E_{t_{m-1}, 1}, \dots, E_{t_{m-1}, K}\}$ according to the time transpired at t_m since the event e , that is $\gamma_e(t_m)$. However, the seeking of the event e^* still considers the time interval as in (3.12). By using the interval formulation we are interested in understanding whether there exists an evolution of the transitivity effect on the event rate that depends on the recency of events constituting the triadic pattern. According to the step-wise formulation of transitivity, we can

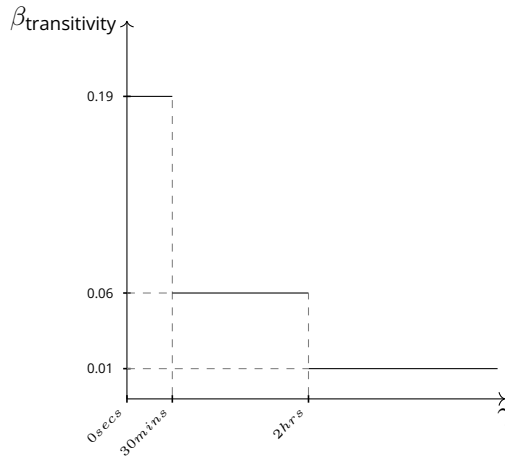


Figure 3.5: step-wise function for the effect of *Transitivity* on the event rate. The function defines three intervals of elapsed time γ (on the x-axis): the first interval $\gamma \in (0secs, 30mins]$, the second interval $\gamma \in (30mins, 2hrs]$ and the third interval $\gamma \in (2hrs, +\infty)$. The y-axis shows the value of the effect $\beta_{transitivity}$ for each interval.

rewrite the rate in (3.13) as follows:

$$\lambda(s_{e'}, r_{e'}, E_{t_{m-1}}, \beta) = \exp \left\{ \sum_{k=1}^K \beta_{transitivity_k} transitivity_k(s_{e'}, r_{e'}, t_m) \right\} \quad (3.15)$$

The effect of transitivity across intervals conveys more information than in the case without intervals. Although, the interpretation of positive and negative effects remains the same (i.e. positive effects still promote the closure of triads as well as negative effects keep discouraging it), the intensity of such behaviors that promote/discourage triadic closure can change over time and this is the additional information we are after. For instance, if the effects from the first to the last interval are positive and decreasing, that is $\beta_{transitivity_1} > \dots > \beta_{transitivity_K}$, this means that the closer in time the events in the triad are to each other the faster the third event in the pattern is likely to happen.

The function in (3.11) can be written also in the case of triadic statistics:

$$\beta_{\text{transitivity}}(\gamma) = \begin{cases} \beta_{\text{transitivity}_1} & \text{if } \gamma \in (\gamma_0, \gamma_1] \\ \vdots & \\ \beta_{\text{transitivity}_K} & \text{if } \gamma \in (\gamma_{K-1}, \gamma_K] \\ 0 & \text{otherwise} \end{cases} \quad (3.16)$$

A simple example of step-wise effects for transitivity closure is shown in Figure 3.5: if we only consider transitivity closure in the model we can conclude that the more triadic events occurred recently, the sooner the third event in the triadic pattern is likely to happen. Formulas of further second-order statistics can be found in Appendix A.1.

3.3.3 Estimation of a relational event model with a step-wise memory decay

The relational event model with step-wise memory decay of endogenous effects has the advantage that it can be easily estimated using existing software as `relevant` (Butts, 2008), `goldfish` (Stadtfeld & Hollway, 2020), `rem` (Brandenberger, 2018a), or `remverse` (Mulder et al., 2020). This can be done as follows. First, the transpired time needs to be divided into disjoint intervals with bounds $\gamma_0, \dots, \gamma_K$. The bounds should be determined such that the step-wise function will be able to capture the expected memory; for periods where a fast (slow) decay is expected narrow (wide) intervals should be chosen. Next, each endogenous statistic (e.g., inertia, transitivity) is split in K separate statistics that capture the volume of past interactions in the K intervals of transpired time. The resulting set of relational event statistics can then be plugged into existing functions for fitting relational event models.

Despite the computational advantage, the step-wise memory decay in (3.11) and in (3.16) has two potential challenges: a substantive challenge is that it may not always be realistic that memory decay occurs in a step-wise fashion in real life; a methodological challenge is that it may be unclear how many intervals (K) should be chosen and where the boundaries $\gamma = (\gamma_0, \dots, \gamma_K)$ should be placed. When a researcher aims to learn a more fine-grained, potentially smoother continuous decay, it is of course possible to increase the number of intervals. However, we would still be constraining results to prespecified bound-

aries (the choice for which may not be obvious) and estimates could lose accuracy as this would greatly increase the number of free parameters in the model to be estimated and reduce the number of events per interval. Therefore, we now take the following two steps. First, we develop a continuous memory decay approach that solves these issues. Next, we show how the step-wise model can be used as a building block for an approximation of this continuous decay model.

3.4 The gradual nature of memory decay

Since past events often lose their effect gradually over time (rather than step-wise), we propose an often more realistic form of the memory decay in (3.11) and (3.16) where, instead of constraining effects to be constant within intervals of γ , their change can be continuous over it and depends on a vector of parameters θ that define the resulting shape of the decay. The continuous effect for statistic u can be written as

$$\beta_u(\gamma, \theta) \tag{3.17}$$

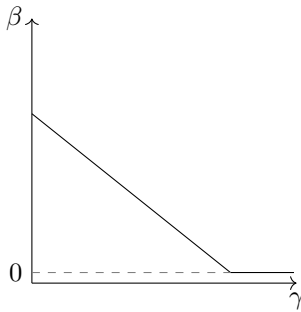
where β_u is a continuous function on γ , describing the trend of the effect of u such that $\beta_u : \mathcal{D} \rightarrow \mathbb{R}$ and $\mathcal{D} = \mathbb{R}^+ \setminus \{\gamma > \gamma_K\}$, with γ_K being a time length limit either due to the empirical data or simply justified by the researcher. The set of parameters $\theta \in S(\theta)$ defines the shape of the decay, where $S(\theta)$ is their support.

We propose several monotonously decreasing functions $\beta_u(\gamma, \theta)$ that might reflect the actual underlying memory decay. The continuous trends in Figure 3.6 assume effects to be positive and decreasing towards zero as the time transpired since the event increases.

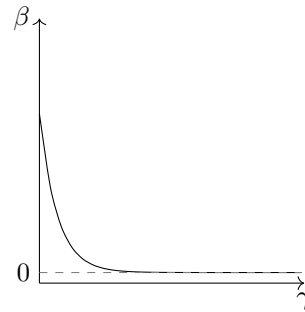
- linear decrease (Figure 3.6a):

$$\beta_u(\gamma, \theta_1, \theta_2) = \begin{cases} \theta_2 - \frac{\theta_2}{\theta_1}\gamma & \text{for } \gamma < \theta_1 \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

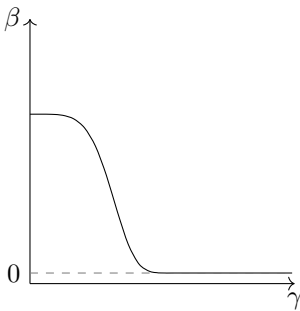
where $\theta = \{\theta_1, \theta_2\}$, $\theta_2 > 0$ is the maximum value assumed by the function and $-\frac{\theta_2}{\theta_1}$ (with $\theta_1 > 0$) is the slope of the line that describes the steepness of the decrease;



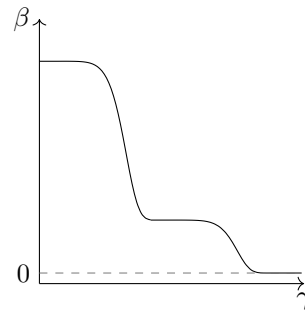
(a) linear decay.



(b) exponential decay.



(c) one-smooth-step decay.



(d) two-smooth-steps decay.

Figure 3.6: Possible trends of the effect β for any endogenous statistics. All four trends develop over γ (x-axis), which is the elapsed time of the event characterizing the statistic. In these specific examples, trends decrease towards zero with different shapes depending on a set of memory parameters θ : (a) linear, (b) exponential, (c) an initial step with a smoothed decrease, (d) two smoothed and decreasing steps.

- exponential and one-smooth-step decrease (Figure 3.6b and Figure 3.6c):

$$\beta_u(\gamma, \theta_1, \theta_2, \theta_3) = \theta_3 \exp \left\{ - \left(\frac{\gamma}{\theta_1} \right)^{\theta_2} \right\} \quad (3.19)$$

where the set of parameters $\theta = \{\theta_1, \theta_2, \theta_3\}$ consists of: $\theta_1 > 0$ and $\theta_3 > 0$ that are scale parameters (θ_3 corresponds to the maximum value assumed by the function), $\theta_2 > 0$ is a shape parameter. The survival function of a Weibull distribution is a specific case of the function (3.19) where the maximum value is $\theta_3 = 1$. Moreover, where $\theta_2 = 1$, $\theta_3 = \frac{1}{\theta_1}$, the (3.19) reduces to

the exponential decreasing weight in Brandes et al. (2009) and the half-life parameter is then calculated as $T_{1/2} = \theta_1 \log 2$. In most cases (except for the exponential one) the trend starts evolving at an initial constant value (one-smooth-step trend) that is the maximum value θ_3 and then decreases to zero as γ increases;

- smoothed multiple steps (Figure 3.6d): this is a combination of two or more smoothed one-step trends.

The relative influence of past events on the dyadic event rate can follow other more complex shapes than those presented in Figure 3.6. As a result of this continuous definition of effects, inertia as well as other endogenous statistics are no longer computed as the accumulated number of past events but now consist of a sum of weights, where each weight changes according to the transpired time γ of each event; this reflects the relative importance of past events updated at t_m . Therefore, the event rate in (3.10) where only inertia effect is considered and inertia is divided in K intervals becomes:

$$\lambda(s_{e'}, r_{e'}, E_{t_{m-1}}, \theta) = \exp \left\{ \sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(i, j) \beta_{\text{inertia}}(\gamma_e(t_m), \theta) \right\} \quad (3.20)$$

where $\beta(\gamma_e(t_m), \theta)$ is a continuous function that returns the relative effect as to the event e contributing to the inertia statistics, $\gamma_e(t_m) = t_m - t_e$ is the time transpired at t_m since t_e (and increases over time), and θ is the set of parameters that describe the shape of the decay. A formal mathematical procedure about moving from a step-wise effect function to a continuous effect function can be found in the Appendix A.2.

However, the process of estimation of the set of parameters θ governing the memory evolution results in a computationally complex maximization of the likelihood in (3.1). The more realistic scenario that the influence of past events changes as a continuous function of their elapsed time since the current time comes at the expense of constantly changing values of the network statistics; this increases the complexity of their estimation. Hence, in the next subsection we revalue the step-wise approach and present a Bayesian approach to approximate continuous memory decay with it.

3.5 A semi-parametric approach to estimate a smooth memory decay

In this section we propose a methodology that (i) builds on the computational advantage of the step-wise model introduced in Section (3.3.1), (ii) avoids the issue of arbitrarily choosing intervals, and (iii) results in an approximate continuous estimate for memory decay. This is achieved by applying *Bayesian Model Averaging* (BMA) (Volinsky et al., 1999) to model memory decay in endogenous REM statistics. The idea is to randomly generate a bag of many step-wise models with different interval configurations for the transpired time. Next, the fit of all these models is evaluated and a weighted average of all step-wise models (weighted according to their relative fit) is achieved. This results in that approximate smooth trend for the memory decay that best fits the data.

We start with a simple example where we look at inertia. If we consider Q step-wise models and denote a single step-wise model by \mathcal{M}_q , then the Bayesian model average of the posterior distribution of the decay of the inertia effect β_{inertia} as a function of the transpired time γ is defined by

$$p(\beta_{\text{inertia}}(\gamma)|E_{t_M}) = \sum_{q=1}^Q p(\beta_{\text{inertia}}(\gamma)|E_{t_M}, \mathcal{M}_q)p(\mathcal{M}_q|E_{t_M}). \quad (3.21)$$

Bayesian model averaging is, in fact, a direct application of the law of total probability where we marginalize over the discrete model space $\{\mathcal{M}_1, \dots, \mathcal{M}_Q\}$. Note that the law of total probability can be applied because a Bayesian framework allows us to quantify the uncertainty about a statistical model using probabilities. For other endogenous effects or other quantities of interest, Bayesian model averaging can be used in a similar manner. The posterior probabilities, $p(\mathcal{M}_q|E_{t_M})$, serve as relative weights in the Bayesian model average. Below, we consider two approaches to quantify these probabilities: BIC and WAIC. Before discussing these we explain how we can generate a bag of step-wise models to approximate different memory decay functions.

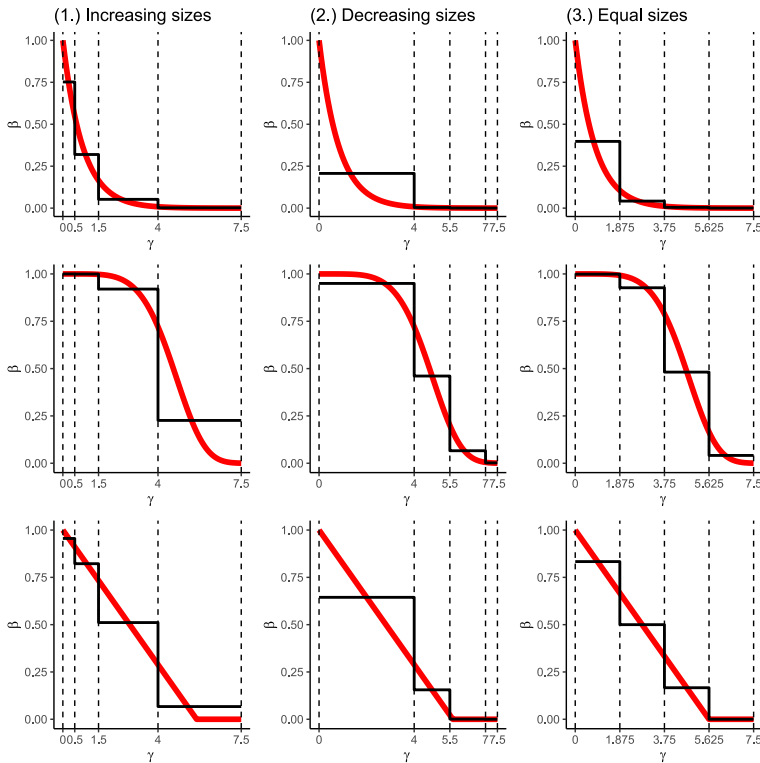


Figure 3.7: Examples of approximation of three different decays (red lines) by means of three types of step-wise functions (black lines) defined according to three different types of interval widths. The type of decay differs row-wise, from the top to the bottom: exponential decay, one-smooth-step decay and linear decay. The type of interval widths differs column-wise, from left to the right: increasing size, decreasing size and equal size intervals. The maximum time width is fixed to $\gamma_K = 7.5$.

3.5.1 Generating a bag of step-wise relational event models

First, we define a bag of Q step-wise relational event models where the transpired time is divided into interval configurations:

$$\mathcal{M}_q : \gamma_q, \text{ with } \gamma_q = (\gamma_{q0}, \dots, \gamma_{qK_q}),$$

where K_q denotes the number of intervals in model \mathcal{M}_q . In order for the bag of models to approximate a variety of possible shapes, we vary both the number of intervals (K) and the widths of the intervals. The sequences of time widths may be generated according to three features reflecting three possible changes of the decay over time:

- (i) when memory change is likely to be stronger for the more recent events and to change less for events that already are in the farther past (where it is approximately constant) (e.g., an exponential decay), then intervals with increasing size will better catch this behavior and their widths will follow the inequality: $\gamma_k - \gamma_{k-1} < \gamma_{k+1} - \gamma_k$ for $k = 1, \dots, K - 1$. In other words, memory is short such that events are "forgotten" fairly fast and the most recent events carry a much higher weight than less recent events, and fairly distant events have as little effect on the future as events from the far past. The increasing size intervals (i) are generated by means of an algorithm based on the Dirichlet distribution and its pseudocode can be found in Appendix [A.3](#).
- (ii) if the decay is expected to occur in the long term (close to γ_K) whereas it is steady during the more recent past (e.g., a one-smoothed step decay), then intervals with decreasing size will be best capable of catching this behavior and their widths will satisfy the inequality: $\gamma_k - \gamma_{k-1} > \gamma_{k+1} - \gamma_k$ for $k = 1, \dots, K - 1$. These widths can be generated by simply inverting the increasing widths in (i). This represents the situation where the effect of events decays only slowly for a while until they are far enough back in time, which is when they lose their effect fast (e.g., where events from the past week matter, but anything beyond that is quickly forgotten). The decreasing size intervals (ii) are generated by first drawing random intervals using increasing intervals according to (i), and subsequently, the order of the widths are inverted.
- (iii) if the decay is likely to decrease at a constant pace (e.g., a linear decreasing function), intervals of the same size will most easily emulate this behavior.

Figure [3.7](#) illustrates how different interval configurations can approximate different possible shapes. The figure also shows that a single step-wise model cannot approximate these smooth shapes accurately. Rather, an appropriate approximation can be achieved by taking a weighted average of many step-wise models. We discuss the computation of these weights next.

3.5.2 Evaluating the fit of the step-wise relational event models

In this section, we describe two weighting systems for the Q step-wise models that were generated in the previous section. The first weighting system is based on the BIC (capturing the probability of the observed data under each step-wise model (Raftery, 1995; Schwarz, 1978)). The second weighting system is based on the WAIC (which quantifies the predictive performance of each step-wise model (Vehtari et al., 2017; Watanabe, 2013)).

BIC weights

In a Bayesian analysis, the posterior probability of a model is obtained using Bayes' theorem:

$$p(\mathcal{M}_q | E_{t_M}) = \frac{p(E_{t_M} | \mathcal{M}_q) p(\mathcal{M}_q)}{p(E_{t_M})},$$

where $p(E_{t_M} | \mathcal{M}_q)$ denotes the probability of the observed data under a given model (also referred to as the marginal likelihood), $p(\mathcal{M}_q)$ is the prior probability of the model, and $p(E_{t_M})$ is the marginal probability of the data. We assume that all step-wise models are equally likely a priori, i.e., $p(\mathcal{M}_q) = \frac{1}{Q}$. The computation of the marginal likelihood can be expensive (Kass & Raftery, 1995). For this reason the Bayesian information criterion is used as an approximation (Raftery, 1995; Schwarz, 1978):

$$p(E_{t_M} | \mathcal{M}_q) \approx \exp\{-BIC_q/2\},$$

where the BIC of model M_q is computed as

$$BIC_q = d_q \log(n) - 2p(E_{t_M} | \hat{\beta}_q),$$

where d_q is the number of parameters under model M_q and $p(E_{t_M} | \hat{\beta}_q)$ is the maximized log likelihood under M_q .

Thus, the normalized BIC weight for the q -th model is

$$w_q^{\text{BIC}} = \frac{\exp\{-BIC_q/2\}}{\sum_{r=1}^Q \exp\{-BIC_r/2\}} \quad (3.22)$$

Despite its theoretical and computational appeal, it has been shown that the

marginal likelihood, and its approximation via the BIC, may not perform well in Bayesian model averaging problems when the “true model” is not part of the bag of models that is considered. This is also called a \mathcal{M} -open model selection problem (Yao et al., 2018). In the current setting this would be the case when the true decay function is smooth, that is not part of the bag of models but it could very well be the true shape of the decay in real-life networks. In this case, the relative weight in (3.22) converges to 1 for the step-wise model that is closest to the truth as the sample size grows. However, a smooth function can better be approximated by averaging over multiple step-wise models than by placing all its weight on one step-wise model. In such \mathcal{M} -open problems it is preferable to use weights that are based on the WAIC.

WAIC weights

WAIC weights build upon the Expected Log-pointwise Predictive Density (ELPD) (Vehtari et al., 2017; Watanabe, 2013; Yao et al., 2018). In each step-wise model, the ELPD quantifies the quality of the posterior predictions given the estimated posterior distribution of the model parameters. Therefore, if the model performs well in predicting new observations, then the predictive power quantified by the ELPD will assume a high value on a log-density scale as well as on a density scale. The calculation of the *Watanabe-Akaike Information Criterion* (WAIC) is based on an approximation of the ELPD as follows:

$$\widehat{elpd}_q^{\text{waic}} = \widehat{lpd}_q - \hat{p}_q^{\text{waic}} \quad \text{for } q = 1, \dots, Q \quad (3.23)$$

where the Log-pointwise Predictive Density (\widehat{lpd}_q) represents the predictive log-density calculated on in-sample observations and typically overestimates the actual ELPD. This can be corrected by subtracting \hat{p}_q^{waic} , which quantifies the uncertainty introduced by the posterior distribution of the model parameters (β_q) in predicting the in-sample observations and can be seen as a form of penalization.

Hence, WAIC weights are computed as

$$w_q^{\text{WAIC}} = \frac{\exp \left\{ \widehat{elpd}_q^{\text{waic}} \right\}}{\sum_{q=1}^Q \exp \left\{ \widehat{elpd}_q^{\text{waic}} \right\}}, \quad q = 1, \dots, Q \quad (3.24)$$

Thus, the higher the estimated predictive power of a model ($\widehat{elpd}_q^{\text{waic}}$), the higher its WAIC-based weight (w_q^{WAIC}).

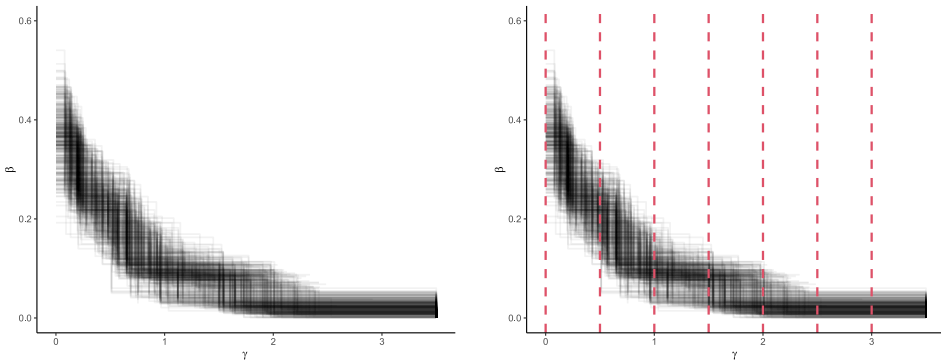
3.5.3 Bayesian model averaging for approximating smooth decay functions

By means of BMA one can elicit a posterior estimate of a quantity of interest as well as its average posterior predictive distribution by finding the optimal linear combination of a set of models, and accounting, in turn, for their uncertainty. A crucial aspect of BMA is the use of model weights that quantify the relative importance of the models according to their posterior probability. In Section 3.5.2, we considered two weighting systems that can be employed in the estimation of the memory decay trend. Here we explain how to get posterior draws of the decay function of an endogenous effect from the Bayesian model averaged posterior.

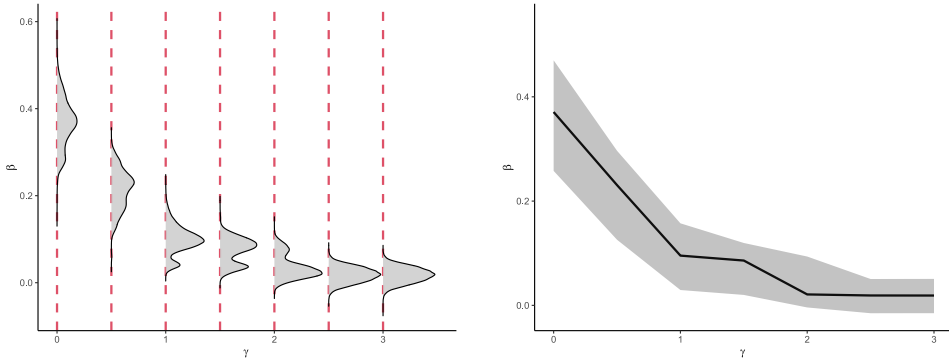
In BMA, the posterior estimate of any parameter of interest can be calculated as the weighted mean of the posterior estimates provided by each model in the averaging. Considering (3.21), we can generate a posterior draw by first randomly selecting a model from the bag of models according to their relative weights, and then generate a trend from the posterior distribution of the selected model. We achieve this last step by approximating the posterior of β using a multivariate normal distribution where the mean is equal to the maximum likelihood estimates and the posterior covariance matrix is set equal to the error covariance matrix. This is an application of large sample theory in a Bayesian framework (Gelman et al., 2013). We consider the following steps to get posterior draws:

1. Draw a model from $\mathcal{M}_q | E_{t_M} \sim \text{Multinomial}(\mathbf{w})$, where the vector of normalized weights $\mathbf{w} = (w_1, \dots, w_Q)$ quantify the relative fit of the respective step-wise models;
2. Generate a vector of posterior effects from $\beta | \mathcal{M}_q, E_{t_M} \sim \text{MVN}(\hat{\beta}_q, \hat{\Sigma}_q)$. The posterior distribution for the step-wise model M_q (the model drawn at the first step) is approximated by a multivariate normal distribution with parameters given by maximum likelihood estimates under model M_q and corresponding error covariance matrix;
3. Repeat steps 1 and 2 a sufficient number of times.

Chapter 3



(a) Result of the Bayesian Model Averaging: posterior draws of (step-wise) β generated by repeating step 1. and 2. (b) Estimating the posterior trend (first step): defining a dense grid of evenly spaced γ 's (vertical dashed red lines).



(c) Estimating the posterior trend (second step): for each γ the corresponding interval effect in each posterior step-wise draw is selected. The resulting density characterizes the posterior density at the specific γ . (d) Posterior trend of β : for each γ and given the corresponding estimated posterior density (estimated in (c)), the posterior mode as well as the highest posterior density interval are estimated.

Figure 3.8: The estimate of the posterior decay is explained here in four plots: (a) BMA: result of repeating step 1. and 2. in the approximation of the posterior distribution of β over γ by means of step-wise models; (b) selecting a grid of γ 's (vertical dashed red lines); (c) posterior conditional densities at given γ 's; (d) resulting posterior trend of β (the gray region represents the highest posterior density interval at 95%).

After these three steps, the resulting posterior distribution of each endogenous effect β over γ resembles Figure 3.8a. Then, we estimate the posterior decay of the effect over γ as follows: (i) define a (dense) grid with evenly spaced $\gamma \in [0, \gamma_K]$, where γ_K is usually based on the data (Figure 3.8b, first step); (ii)

for each γ select the corresponding interval effect in each posterior draw (as shown by the step-wise functions in (3.11) and (3.16)), this selection results in a posterior density at a given γ (Figure 3.8c, *second step*); (iii) calculate the posterior mode of these densities as well as their highest posterior density intervals at each γ , resulting in a semi-continuous effect decay (Figure 3.8d). As a consequence of this, the posterior estimate of those statistics that are not defined in intervals (e.g. a baseline effect) is simply obtained with the draws generated after the three initial steps.

3.5.4 Computational details of the BMA

The most expensive step before estimating the posterior decay with the BMA is the estimation of all the Q step-wise models in the bag. This subsection focuses on computational complexity of step-wise models compared to parametric decay models (e.g., exponential decay). We describe two stages where such models show differences in terms of their computational complexity: (1) the computation of endogenous statistics and (2) the estimation stage.

Calculation of endogenous statistics: a comparison on the number of operations performed in a single model

The computation of endogenous statistics is a time-consuming stage as it must be carried out across all the observed time points (M) and for all the dyads that can occur over time. Without loss of generality we assume that at each time point all dyads are at risk of occurring, thus we consider the complete risk set as it is assumed in (3.1) where $D = |\mathcal{R}| = N \times (N - 1)$, with N being the number of actors and D the number of dyads in the risk set \mathcal{R} . When a parametric weight decay is used (e.g., exponential decay), the computation of the endogenous statistics requires more operations than what is required in a step-wise model: the weight of past events has to be updated at each time point where an event is observed and according to the weight decay function. Such update requires the numerical evaluation of the decay function and this eventually increases the needed computational time.

The continuous update of the event weights is not required for the step-wise decay model where past events are assumed to have a unitary weight in each interval. Therefore, for a step-wise model the main steps for computing each endogenous statistic consist of: (i) at each time point defining the partitions of

the event history according to the K intervals describing the step-wise model, (ii) computing each endogenous statistic within such intervals. We optimize these two steps by minimizing the number of times that the algorithm has to compute the endogenous statistics according to each specific interval. Some time intervals might appear more than once along the event sequence. Therefore, we first find the time boundaries of the K intervals across all the time points, then we consider the reduced set of intervals and calculate the endogenous statistics according to this reduced set. Finally, for each interval of the reduced set, the value of the endogenous statistics for all dyads is assigned to the correspondent interval in the original data-structure for the statistics, which is used in the estimation stage. This improvement makes the computation of the statistics faster since we avoid to compute the same statistic more than once. This optimization only works for endogenous statistics such as inertia, reciprocity, in-/out-degree, and other first-order endogenous statistics as well as for second- or higher-order endogenous statistics where the time order of the events doesn't affect the value of the statistic.

The number of operations required in the computation of a single endogenous statistic can be quantified as follows:

- In a step-wise decay model without our optimization, the number of operations is $(M - 1) \times D \times K + (M - 1) \times (K + 1)$, where $(M - 1) \times D \times K$ consists of the number of times the statistic is computed, which is at each time point for each interval and for all dyads. We consider $M - 1$ because at time t_1 all endogenous statistics assume value zero for all dyads. Furthermore, $(M - 1) \times (K + 1)$ is the total number of updates of the time boundaries characterizing the K intervals throughout the event sequence. This step runs fast because it only requires simple subtractions between numbers;
- In a step-wise decay model where our optimization is performed the number of operations is $\Psi(\mathbf{t}, \boldsymbol{\gamma}) \times D + (M - 1) \times (K + 1)$, where $\Psi(\mathbf{t}, \boldsymbol{\gamma})$ is the size of the reduced set of intervals; this is $\Psi < (M - 1) \times K$ and it depends both on the vector of observed time points $\mathbf{t} = (t_1, \dots, t_M)$ and on the vector of $K + 1$ increasing widths $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \dots, \gamma_K)$ that define the K intervals over time. Furthermore, $(M - 1) \times (K + 1)$ again is the number of times we have to update the time boundaries before finding the reduced risk set;
- In a parametric decay model (e.g., exponential, linear, or other decays) the

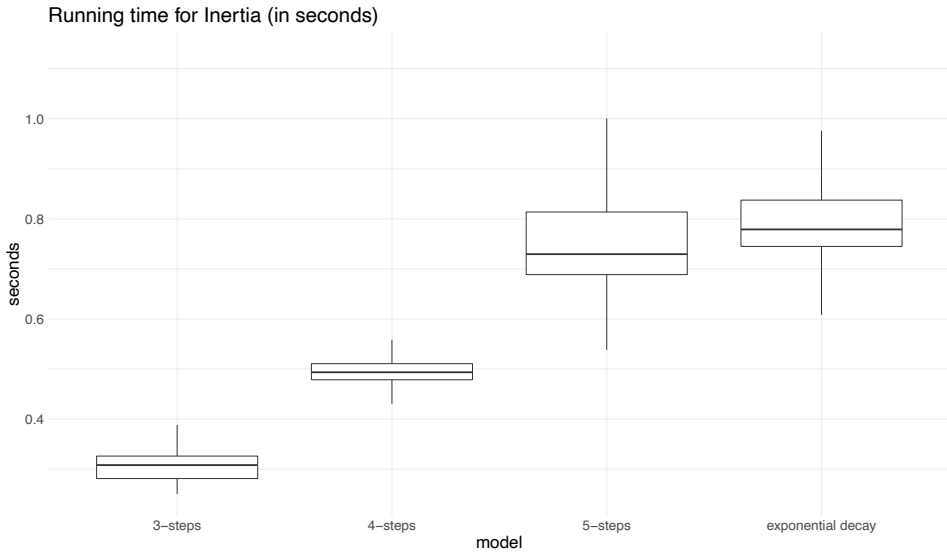


Figure 3.9: Distributions of running times for the endogenous statistic Inertia (in seconds). 3-steps, 4-steps and 5-steps models are compared to the parametric model with exponential decay. For each type of model, the running time was measured 1000 times.

number of operations is $(M - 1) \times D + \frac{M \times (M - 1)}{2}$, where $(M - 1) \times D$ is the number of times the statistic is computed and $\frac{M \times (M - 1)}{2}$ is the number of total updates for the weights of the already-occurred dyads.

Let us compare the optimized step-wise decay with the parametric decay model. For this effort, we assume that: (i) K is set to a low number around 3, 4 or 5 intervals; and (ii) the update of one single event weight requires as much computational time as the update of one time bound. Then, the number of updates in a parametric decay increases faster than in a step-wise decay. Indeed, the $(M - 1) \times (K + 1)$ operations for the computation of the time boundaries in the optimized step-wise model follow a linear function of the number of events (M), whereas the $\frac{M \times (M - 1)}{2}$ operations for the update of the weights in the parametric model follow a quadratic function of the number of events. Unfortunately, the optimized approach for the step-wise model cannot be performed on the transitivity closure introduced in Section 3.3.2, because the order of events in the triadic pattern matters. However, the optimization for the first-order statistics already saves much computational time, because in the estimation of more endogenous statistics the reduced set of intervals will be shared and calculated only once. In Figure 3.9, we compare the running times for estimating inertia

using four models: the optimized step-wise model with $K = \{3, 4, 5\}$ and the parametric decay model with exponential decay. Per each model, a set of intervals or half-life values were chosen, and their running times for computing inertia were repeatedly measured (each run of the algorithm was parallelized on 8 threads). Finally, every model has a total number of 1000 samples of running times. We performed such analysis on the empirical data used in Section [3.6](#) where the number of actors is $N = 10$, the number of dyads is $D = 90$ and the number of events is $M = 7567$.

Estimation stage: comparison on the number of parameters to be estimated

Considering U_{exo} exogenous statistics (including the intercept) and U_{endo} endogenous statistics, in the estimation stage the total number of parameters to be estimated is

- $U_{\text{exo}} + U_{\text{endo}}$ in REMs where endogenous statistics follow any parametric weight decay (e.g., exponential, linear, one-step decay);
- $U_{\text{exo}} + (U_{\text{endo}} \times K)$ in step-wise REMs where K is the number of intervals (steps) and all the endogenous statistics follow the same step-wise model.

Therefore, a step-wise model has always more parameters than a model with any parametric decay. However, this disadvantage at the estimation stage is not really an issue because it is not recommended to consider many intervals as the uncertainty around estimates increases when intervals become narrower and only a few events fall inside them.

3.6 Case study: investigating the presence of memory decay in the sequence of demands sent among Indian socio-political actors

We have now introduced our modeling approach, starting from a purely step-wise decay model to a continuous decay model based on model averaging of a set of step-wise models. In this section, we illustrate the method by applying it to empirical data. First, we describe the empirical application and dataset. Next, we present analyses using different prespecified step-wise decay functions, fol-

lowed by an application of the Bayesian model averaging estimated to obtain approximate smooth decay functions. Finally, we compare the semi-parametric model (that results from the Bayesian Model Averaging) with other relational event models where the memory decay is fixed either to a step-wise or exponential decay. In this comparison we focus on the predictive performance of the models as well as their resulting fit.

3.6.1 Relational events between socio-political actors

We retrieved data from the ICEWS (Integrated Crisis Early Warning System) (Boschee et al., 2015) repository, which is hosted in the Harvard Dataverse repository. ICEWS consists of relational events interactions between socio-political actors that were extracted from news articles. Information about the source actor, the target actor, and the event type is recorded along with geographical and temporal data that are available within the same news article. Event types are coded according to the CAMEO (Conflict and Mediation Event Observations) ontology. In this example analysis, we focus on the sequence of relational events within the country of India. Each event represents a request from an actor targeted to another actor. These requests range from humanitarian to military or economic in nature and in this analysis this distinction is not made.

The event sequence includes $M = 7567$ dyadic events between June 2012 and April 2020 among the ten most active actor types: citizens, government, police, member of the Judiciary, India, Indian National Congress Party, Bharatiya Janata Party, ministry, education sector, and "other authorities." Since the time variable is recorded at a daily level, we consider events that occurred on the same day as evenly spaced throughout that day.

The network dynamics of interest are inertia, reciprocity, and transitivity closure. Given a generic step-wise model with K steps, the log-rate at any time $t \in [t_1, t_M]$ and for any request e' is:

$$\begin{aligned} \log \lambda(s_{e'}, r_{e'}, E_t, \beta) = & \beta_0 + \sum_{k=1}^K \beta_{\text{inertia}_k} \text{inertia}_k(s_{e'}, r_{e'}, t) + \\ & + \sum_{k=1}^K \beta_{\text{reciprocity}_k} \text{reciprocity}_k(s_{e'}, r_{e'}, t) + \\ & + \sum_{k=1}^K \beta_{\text{transitivity closure}_k} \text{transitivity closure}_k(s_{e'}, r_{e'}, t) \end{aligned} \quad (3.25)$$

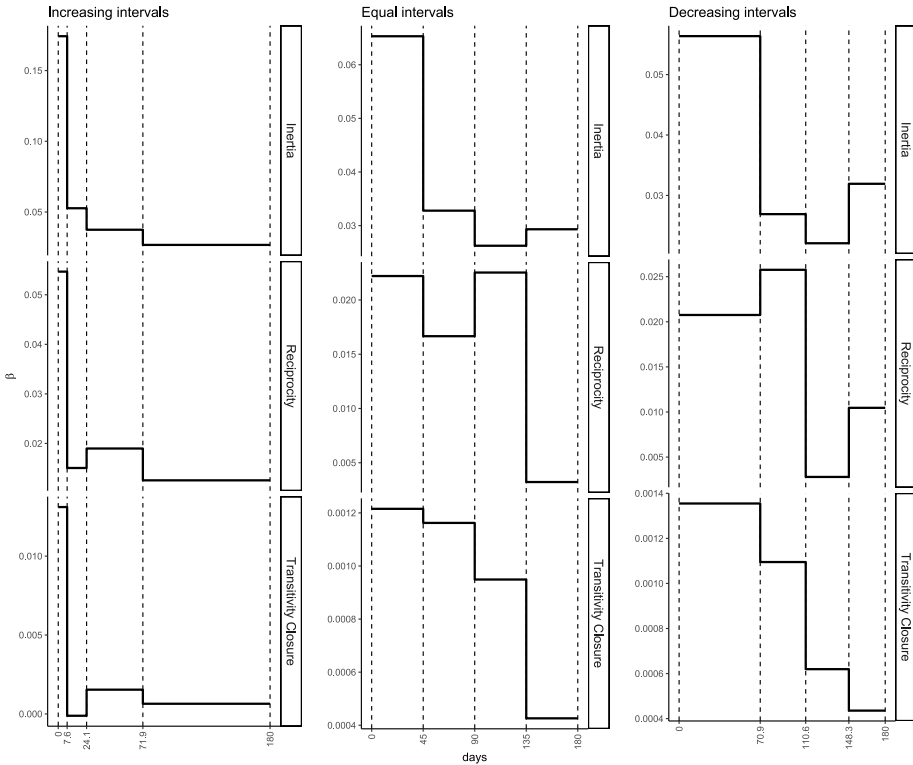


Figure 3.10: MLE estimates of $\hat{\beta} = \{\hat{\beta}_{inertia_1}, \dots, \hat{\beta}_{inertia_4}, \hat{\beta}_{reciprocity_1}, \dots, \hat{\beta}_{reciprocity_4}, \hat{\beta}_{transitivity_1}, \dots, \hat{\beta}_{transitivity_4}\}$ according to three different step-wise models (with $K = 4$) that are randomly chosen from the bag of the estimated models and each following one of the three interval types (by column: increasing, equal and decreasing intervals). The bold black line represents the step-wise function for each endogenous effect in the model and the vertical dashed lines indicate the time bounds characterizing the intervals.

where β_0 represents the logarithm of the baseline rate of requests and the remaining effects describe the estimated step-wise trends for the three network statistics. Inertia quantifies the persistence of the sender in targeting its requests to the same receiver, for instance because the receiver is an actor with some socio-political relevance like a legal figure or authority. Reciprocity describes the level of reciprocation of the sender towards the receiver based on the past volume of interactions that the receiver addressed to the sender. Transitivity closure quantifies the level of information mediation by means of the volume of triads that can be potentially closed by the occurrence of event e' . We assume that at every point in time, every possible dyad is at risk of occurring,

hence the risk set consists of $|\mathcal{R}| = N \times (N - 1) = 90$ dyads.

3.6.2 Predefined step-wise decay models

As the maximum time (γ_K) that past events may affect current relational events we consider 180 days (roughly half an year). Furthermore, we consider three different predefined step-wise memory decay functions by dividing the past in $K = 4$ intervals with either increasing widths, equal widths, or decreasing widths (as described in Section 3.5.1). Figure 3.10 shows the estimated step-wise decay functions for inertia, reciprocity, and transitivity given the three different interval configurations.

As is to be expected, the three models result in different estimated (discretized) shapes of memory decay. For instance, for Transitivity Closure we see that decreasing intervals and increasing intervals produce contrasting decays where the decays not only follow different shapes, but the magnitudes of the effect are different as well. The magnitudes of the effects are similar for the "equal" and "decreasing" intervals, whereas for "increasing" interval widths the magnitudes are quite different from the models with "equal" and "decreasing" widths.

In sum, step-wise models with predefined interval configurations provide us with a very rough idea of how fast memory decays in a given relational event network. However, predefined step-wise memory decay models provide only limited insight into the full shape of memory decay along transpired time, or, for example, whether an (approximated) exponential decay is more likely than a (approximated) smooth one-step decrease. To learn this from an observed relational event network, we need the proposed weighting system for a bag of step-wise models together with a Bayesian model averaging approach. We consider this next.

3.6.3 Approximately smooth memory decay models

For our bag of step-wise models, three sets of 501 intervals were generated for $K = \{3, 4, 5\}$ steps (250 intervals with increasing size, 250 intervals with decreasing size, 1 with equal size). Thus in total, 1503 step-wise models were considered. We chose to use around 500 models per K since we noticed that the overall number of random intervals (1503) already provides stable final results. The estimation of the whole bag of models required about 6.5 hours: for each

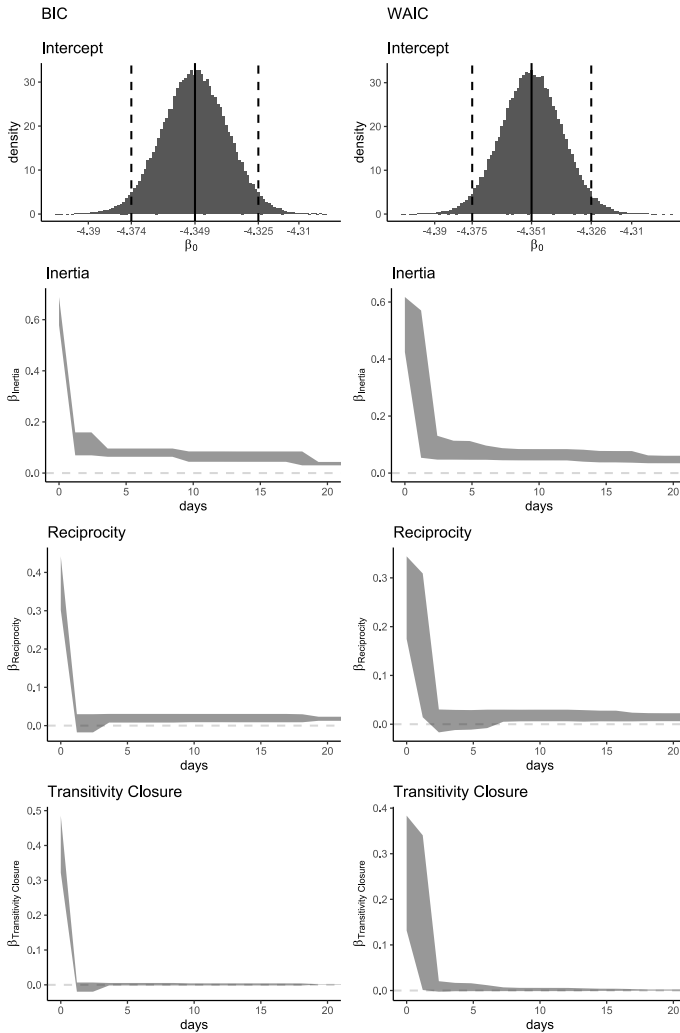


Figure 3.11: Posterior estimates resulting from the BMA with BIC (left) and WAIC (right) weights, from the top to the bottom: posterior distribution for the intercept (β_0), posterior trends for inertia, reciprocity and transitivity closure. The gray area (dashed lines for the intercept) is generated by the posterior highest density intervals calculated until 20 days (maximum value plotted on the x-axis).

step-wise model the computation of the endogenous statistics as well as the estimation of parameters was parallelized on 8 threads¹.

Figure [3.11](#) shows the posterior trends resulting from two Bayesian Model

¹The device used for running the whole-bag estimation had a CPU Intel(R) Core(TM) i7-8750H, Processor Base Frequency 2.20GHz, 8Gb of RAM.

Averaging approaches: one with BIC weights (left panels) and one with WAIC weights (right panels). Because most of the decay occurs in the first twenty days, only this period is plotted in the figure. The intercept β_0 is the only parameter without a decay by definition and the posterior point estimate of the baseline event rate is $\exp\{\hat{\beta}_0\} \approx 0.0129$ (similar for both BIC and WAIC weights; upper panels).

Since the network consists of nodes that represent collectives of individuals, it is important to interpret the estimated memory decay functions as referring to the memory of groups, rather than of individuals. Focusing on the results for the WAIC weights in Figure 3.11 (right panels), all the three trends show a clear approximately exponential memory decay. The drastic decrease near zero suggests that recent requests have a much higher impact on the event rate than less recent ones. Therefore, the trend observed for inertia indicates a tendency of actors to keep sending requests to the same recipient of their most recent requests. This reflects "short-lived inertia" (driven by the requests that happened in a fairly recent past) rather than "long-lived inertia" (where requests that have occurred over a much longer time span continue to be repeated).

For reciprocity, we see that memory drops a bit faster than for inertia and stabilizes around a low value that decreases further, indicating that actors reciprocate on requests received in the very recent past, but requests that were not responded to quickly are soon "forgotten" and are unlikely to be responded to. Norms of reciprocity are clearly not enduring and non-reciprocated requests disappear from social memory very quickly. Finally, transitivity is similarly driven by very recent interactions. Considering that dyadic requests only briefly trigger the tendency to respond, it makes sense that having common past communication partners also mainly matters if those joint interactions date back to only recent history rather than to a period somewhat longer ago.

Together, the results paint a picture of a "delusion of the day" kind of politics. Interactions between these institutional actors appears to be driven by current events in the country, where response to actuality appears more predictive of future interactions than long-term governed interaction. While this may be typical of governmental interactions, the effect may be strengthened by the fact that the data come from news paper articles. News paper articles will generally only report publicly visible interaction (hence, journalists may miss interaction that occurs behind closed doors or interactions that are not made public) and will tend to focus mainly on what is of interest "today." That said, it does make

a lot of sense to find that governmental parties seem to base their interactions mainly (but not exclusively) on what is going on in the present and the very recent past, and focus less on what happened longer ago and may be less salient in the public's eye.

The resulting trends obtained from the BIC weights approximately follow the same decays as the WAIC. However, we see that the BIC weights show an approximate step-wise trend because the BIC becomes increasingly large for that step-wise model that is the closest to the true (smooth) model (in terms of Kullback-Leibler distance (Grünwald & van Ommen, 2017)). Thus, the weight of that step-wise model dominates the weights of all other step-wise models. This illustrates that the BIC is useful for finding the best fitting step-wise model, which, in this case, has increasing interval widths over the transpired time, forming roughly an exponential decay. On the other hand, the BIC is less useful for finding an approximate smooth decay trend. For this purpose we recommend the WAIC.

3.6.4 Assessing the predictive performance: a comparison with parametric memory decays

The results show that memory decays approximately exponentially in this dataset. Next, we compare the performance of the fitted semi-parametric model with other relational event models that either do not contemplate a memory decay (REM without memory) or fix it to some predefined parametric trend (step-wise or exponential):

- **REM without memory:** this is a basic relational event model where endogenous statistics such as inertia, reciprocity, and transitivity closure are embedded in the linear predictor as a function of the total volume of past events without any memory decay. For the REM model without memory, the log-rate at any time $t \in [t_1, t_M]$ and for any request e' in the risk set \mathcal{R} is:

$$\begin{aligned} \log \lambda(s_{e'}, r_{e'}, E_t, \beta) = & \beta_0 + \beta_{\text{inertia}} \text{inertia}(s_{e'}, r_{e'}, t) + \\ & + \beta_{\text{reciprocity}} \text{reciprocity}(s_{e'}, r_{e'}, t) + \\ & + \beta_{\text{transitivity closure}} \text{transitivity closure}(s_{e'}, r_{e'}, t) \end{aligned} \quad (3.26)$$

- **REM with exponential decay:** We specify three models with endogenous

statistics such that events follow an exponential weight decay. The weight decay at t_m for any event occurred at $t_{e'} < t_m$ is

$$\frac{\ln(2)}{\theta_{\text{half-life}}} \exp \left\{ -(t_m - t_{e'}) \frac{\ln(2)}{\theta_{\text{half-life}}} \right\}$$

(Brandes et al., 2009), where $\theta_{\text{half-life}}$ is fixed, respectively, to 7 days, 30 days, and 90 days. For these models, the log-rate at any time $t \in [t_1, t_M]$ and for any request e' in the risk set \mathcal{R} is:

$$\begin{aligned} \log \lambda(s_{e'}, r_{e'}, E_t, \beta) = & \beta_0 + \beta_{\text{inertia}} \text{weighted-inertia}(s_{e'}, r_{e'}, t, \theta_{\text{half-life}}) + \\ & \beta_{\text{reciprocity}} \text{weighted-reciprocity}(s_{e'}, r_{e'}, t, \theta_{\text{half-life}}) + \\ & + \beta_{\text{transitivity closure}} \text{weighted-transitivity closure}(s_{e'}, r_{e'}, t, \theta_{\text{half-life}}) \end{aligned} \quad (3.27)$$

These models are named *Exp 7*, *Exp 30* and *Exp 90* in Appendix A.4. The idea is similar to the approach of Brandenberger (2018a) who also considers exponential decay models and uses different predefined values for the half-life parameter.

- **REM with step-wise decay:** We specify three step-wise models with the following widths:
 - $\gamma_{\text{days}} = \{0, 90, 180\}$ (two intervals with equal size);
 - $\gamma_{\text{days}} = \{0, 7, 30, 90, 180\}$ (four intervals with increasing size);
 - $\gamma_{\text{days}} = \{0, 1.32, 14, 46.2, 180\}$ (four intervals, using the widths of the model with the best WAIC found with the semi-parametric approach).

The step-wise models above are named respectively *StepEqual*, *StepIncr* and *bestWAIC* in Appendix A.4. The three models have $\gamma_{\text{max}} = 180$ days. The log-rate at any time $t \in [t_1, t_M]$ and for any request e' in the risk set \mathcal{R}

Chapter 3

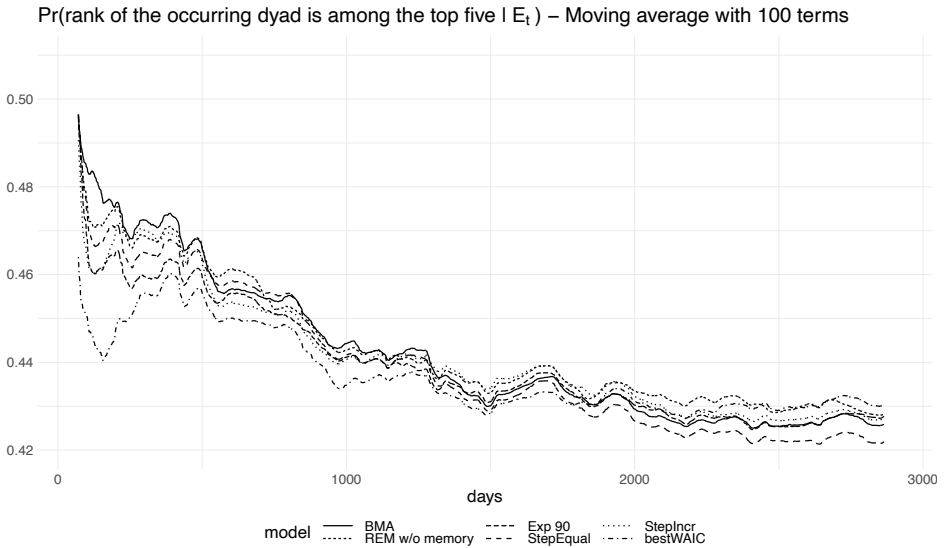


Figure 3.12: Probability of observing the rank of the occurring dyads being among the first five most likely dyads (moving average with 100 terms, *Exp 7* and *Exp 30* performed worse than the rest of the models and were removed from the plot).

is:

$$\begin{aligned} \log \lambda(s_{e'}, r_{e'}, E_t, \beta) = & \beta_0 + \sum_{k=1}^K \beta_{\text{inertia}_k} \text{inertia}_k(s_{e'}, r_{e'}, t) + \\ & + \sum_{k=1}^K \beta_{\text{reciprocity}_k} \text{reciprocity}_k(s_{e'}, r_{e'}, t) + \\ & + \sum_{k=1}^K \beta_{\text{transitivity closure}_k} \text{transitivity closure}_k(s_{e'}, r_{e'}, t) \end{aligned} \quad (3.28)$$

where K is the number of intervals in the model. Endogenous statistics in step-wise models are calculated as explained in Section 3.3.

In Appendix A.4, we include a table with the maximum likelihood estimates and standard errors for each model.

In Figures 3.12 and 3.13, we examine two plots that assess the predictive performance of the models. Figure 3.12 displays the probability of the observed

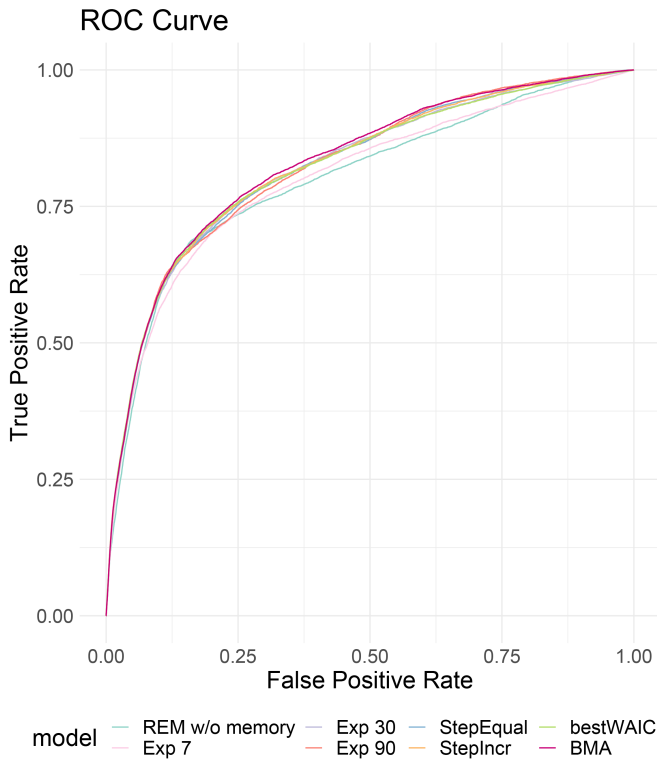


Figure 3.13: ROC curve of each model in the comparison.

dyads having rank less or equal than five, calculated as

$$\sum_{m=1}^Z \mathbb{I}(\text{rank}(e_m) \leq 5) / Z \quad \text{with } Z = 2, \dots, M$$

where $\text{rank}(e_m)$ returns the rank of the predicted probability for event e_m and $M = 7567$ is the number of events in the sequence. Thus, at each time point, given the sequence of already occurred events (including the occurring event at t_m), the count of predicted ranks being less or equal than five (out of the 90 dyads that were at risk at each time point) is divided by the number of events in the partial event sequence. We calculate this probability for all models under comparison. We consider a moving average with 100 events to better visualize the overall predictive trends. We excluded models *Exp 7* and *Exp 30* from the figure because they performed clearly worse than the rest of the models (this keeps the figure more readable).

The plotted trends show how well the models perform over time. The solid line represents the performance of the BMA model resulting from the semi-parametric approach introduced in this chapter. In comparison to the other decay models, its performance maintains a level that is, on average, higher than most of the models in the comparison. This illustrates that a model where the shape of the decay is learned from the data on average results in better predictions and better model fit than competing models where the decay is prespecified based on rough heuristic arguments. Finally it is interesting to observe that the REM without memory also performs quite competitively.

We note that the aim of our approach is not to generate a model that necessarily outperforms other models in predictive accuracy. Although the model is expected to generally do equally well or better than most competing models, an important aspect of the approach is that it allows a researcher to get a good idea of how long past events maintain their influence. This allows a researcher to then specify better further inferential models (informed by the decay shape that is found from the semi-parametric model). Perhaps more importantly, empirical results of exactly how the past keeps influencing the present and the future are essential for theory development. Considering the dearth of time-sensitive social theory, approaches that can uncover the empirical pattern of time can be highly informative for theorists to develop truly time-sensitive social theories upon. Of course, this requires the application of the model to a wider set of data than just our illustrative data set.

We plot the ROC curves in Figure 3.13; again we see that the BMA model on average performs best. Here, the REM without memory performs relatively poorly. The no-memory REM under-predicts actually occurring events and can only achieve high accuracy by predicting a relatively large number of events that actually do not occur. The memory-based models have a better overall trade-off between incorrectly and correctly predicted events, even considering the simplicity (the models are fully based on only inertia, reciprocity, transitivity closure, and an intercept) of the model for such complex interaction patterns among governmental actors in India.

3.7 Discussion

In this chapter, we presented different methods for learning how past interactions between social actors affect future interactions in the network. We first

considered a K -step-wise model that approximated memory decay with a discrete step-wise trend. This model can be estimated using existing software functions for relational event analysis. The proposed Bayesian model averaged memory decay estimator will be made available in a new R package².

The next key contribution is a novel Bayesian model averaging approach to estimating memory decay in a relational modeling framework where events are assumed to continuously change in importance as the time since the event increases. The promising aspect of this semi-parametric approach lies in its ability to learn the shape of the memory decay without making any parametric assumption about it. Furthermore, by building on the step-wise model, the proposed method is computationally feasible. We considered two weighting systems for Bayesian model averaging of a bag of step-wise models: the BIC and the WAIC. As was illustrated, the BIC is useful for finding *the* one best fitting step-wise model for a given empirical relational event history. The BIC, however, is not suitable for finding an approximate smooth trend of the memory decay, as all weight is placed on the single step-wise model that is closest to the true smooth decay model. This issue does not occur for the WAIC as the Bayesian model average of many step-wise models results in a smooth trend.

The semi-parametric approach on average provided better predictive performance than other approaches where the weight decay was set using predefined parameters. This illustrates the usefulness of relaxing the assumption of predefined decay functions when making predictions and doing inferences. Moreover, the semi-parametric approach can uncover exactly how and for how long past events matter and can show if this is perhaps different between reciprocity and transitivity (or other statistics). A researcher can use the semi-parametric approach to first run several relatively simple models that can inform the researcher about the memory decay shapes that are present in the data at hand. Following that, the researcher can then specify further, more complex, models that utilize some predefined memory structure that is based on the shape found by the semi-parametric approach. This allows a researcher to run quite complex relational event models, without the computational burden of repeating the memory decay model several times for each new model that is specified, while, at the same time, taking into account the empirically extracted memory decay function for the dataset at hand.

²R package `bremory` (Arena, 2022a).

Chapter 3

In addition, researchers can use the methodology to uncover empirical trends of how past events matter as time passes by. Once this has been applied to enough datasets, these findings can inform solid theory development on how the past matters for the future. There is barely any social theory that is able to systematically explain and predict how present social interaction affect future social interactions and for how long exactly, whether the effects are linear or non-linear (and, in which case: following which shape?), and which conditions have an effect on that. Although social scientists acknowledge that time and timing matters for social reality (e.g., Ancona et al. (2001), Kozlowski et al. (2016), R. T. A. J. Leenders et al. (2016), Mitchell and James (2001), and Monge (1990)), the empirical means to uncover actual memory shapes or the empirical means to test potential theoretical expectations about the course of time has lacked. We believe that our approach has the ability to support these efforts.

In this chapter, we assume that all events are random, in the sense of having some probability of occurrence at any time. Some events, however, are not random and follow a fixed deterministic pattern. Marcum and Butts (2015) refer to these events as “clock events”. Examples include standardized lunch times (“every day we eat together in the cafeteria between 1200h and 1230h”), fixed office hours, the end of the workday at 1700h, et cetera. These deterministic events can affect interaction rates directly, but can also affect memory decay. For example, consider a workplace where work ends strictly at 1700h. If it happens to be the norm to follow up on a request from a colleague within half an hour (and older requests “drop from the radar”), requests that come in at 1645h should be handled within fifteen minutes and may be forgotten as the clock turns 1700h. In this case, the deterministic end-of-workday event directly affects the memory decay. In situations where clock events occur, it would be interesting to incorporate them into the modeling approach. At the very least, the researcher should be aware of them, so as to not have the memory shapes be affected by the clock events without the researcher realizing it.

The empirical example presented in this chapter involves a relatively small network. It is important to note however that the methodology can be used for larger networks as well, even though the computation can be expensive in that case. We leave computational optimization of the approach for larger networks for future work.

Another important direction for future research would be to apply the method to different event types or sentiments. For instance, one expects negative events

(e.g., a country threatening another country, a pupil insulting a peer, a teacher rebuking a student) to have a memory decay that is slower and more persistent than for positive events (e.g., a teacher praising a student, a country cooperating with another country) (Brass and Labianca, 1999; Labianca and Brass, 2006). This difference may apply as well to other event types from which possible different memory shapes might emerge. For example, it might be that email interaction is more fleeting than face-to-face interaction. This is especially relevant in the understanding of projects where some project members may be co-located and have ample face-to-face interaction, while other members of the project team may reside in different locations which makes technology-enabled communication with them more pertinent. The team leader may give a similar message to a co-located project member (using face-to-face interaction) as to a physically-distant project member (sending an email), where the two communication media may have differential memory effects. Having a modeling approach like the semi-parametric model from this chapter allows researchers to study conditions that affect memory decay patterns differently.

Furthermore, in the case of more dynamic situations, e.g., when the network switches between different states or regimes, memory decay may also change accordingly. For example, in emergency situations, recently past events may play an even larger role on interaction dynamics than long past events compared to the period of time before the emergency happened. Consequently, we would want to learn the change of the shape (and length) of memory decay across different states in dynamic environments.

In our approach, we do not prespecify the shape of the memory decay. However, with the choice for BIC or WAIC and with the choice for increasing / decreasing / equal intervals, some shapes are more likely to be found than others. We have illustrated how a researcher can compare these various choices against each other and pick that specification that fits the data best (according to predictive fit or some other criterion). However, a substantively very meaningful next step would be to examine when it is more plausible for memory decay to follow a step-wise or a continuous shape. It is worth it to systematically examine which social mechanisms are likely to lead to step-wise temporal effects and which mechanisms are not. This would both assist further model building and the further development of time-sensitive social theory.

We expect that the acquired ability of both estimating social memory decay processes and testing for the various conditions that might shape them can

Chapter 3

be a crucial step towards a more accurate understanding of network dynamics developing at a local as well as at a global level.

HOW FAST DO WE FORGET OUR PAST
SOCIAL INTERACTIONS?
UNDERSTANDING MEMORY RETENTION
WITH PARAMETRIC DECAYS IN
RELATIONAL EVENT MODELS

4

Abstract

In relational event networks, endogenous statistics are used to summarize the past activity between actors. Typically it is assumed that past events have equal weight on the social interaction rate in the (near) future regardless of the time that has transpired since observing them. Generally, it is unrealistic to assume that recently past events affect the current event rate to an equal degree as long-past events. Alternatively one may consider using a prespecified decay function with a prespecified rate of decay. A problem then is that the chosen decay function could be misspecified yielding biased results and incorrect conclusions. In this chapter, we introduce three parametric weight decay functions (exponential, linear, and one-step) that can be embedded in a relational event model. A statistical method is presented to decide which memory decay function and memory parameter best fit the observed sequence of events. We present simulation studies that show the presence of bias in the estimates of effects of the statistics whenever the decay, as well as the memory parameter, are not properly estimated, and the ability to test different memory models against each other using the Bayes factor. Finally, we apply the methodology to two empirical case studies.

4.1 Introduction

In relational event networks, the past relational event history between the actors can have an enormous impact on future relational events (Butts, 2008). Research has shown that the past can generally be well-summarized using so-called endogenous statistics to model the events to be observed. These endogenous statistics typically quantify the activity between actors in the past. For example, the endogenous statistic Inertia of actor i towards j for event m is generally computed as the total volume of past events from i to j until the previous event, i.e.,

$$\text{inertia}(i, j, t_m) = \sum_{e' \in E_{t_{m-1}}} \mathbb{I}(s(e') = i, r(e') = j), \quad (4.1)$$

where $E_{t_{m-1}}$ denotes the event history until event $m - 1$, $s(e')$ is the sender of event e' , and $r(e')$ is the receiver of event e' . Thereby the assumption is that the (logarithm of the) relational event rate between two actors depends proportionally on the number of past events between these actors. Other examples of endogenous statistics include reciprocity, transitivity or even more complex higher-order dynamic patterns.

By defining the endogenous statistics as the total number of past events the assumption is that all past events equally contribute to the event rate of the dyads in the subsequent period. This however may not be likely in real-life social networks. As an example let us consider a group of friends who send text messages to each other. At some point let us assume that James sent about 24 messages to Keira, and that Vicky also sent about 24 messages to Roberto (since the observational period). Out of the 24 messages sent by James to Keira, let us assume that 22 were sent more than one month ago, and 2 messages were sent two weeks ago. Out of the 24 messages sent by Vicky to Roberto, all 24 messages were sent in the last 5 days. Now the question is whether it is more likely that James will send a message to Keira next, or that Vicky will send a message to Roberto next? Under the assumption that sending messages is mainly driven by inertia, and inertia is computed as the total volume of past messages between actors (which is equal for both dyads), there would be an equal probability that for James to send a message to Keira, as for Vicky to send a message to Roberto. Given the fact that Vicky has been much more active to send message to Roberto in the recent past however it may be much more plausible that Vicky will send a message to Robert next than James to Keira. This would imply

that recently past relational events have a stronger impact on what happens next than long-past events. Thus one could say that recently past social events are fresh in the memory of actors while long past events may not.

Alternatively it can be assumed that the weight of past events decays according to an exponential function of the transpired time of the past event (Brandes et al., 2009). Indeed, it is likely that over time actors differently weigh past events according to their time recency. The time recency of an event is defined as the time transpired at the present time point since its occurrence. This measure increases over time after the event happens and can be a crucial information in understanding whether and how the weights of events decay as time goes by and their time recency decreases. This would provide value insight to learn how the past affects the future in relational event networks. As proposed in Brandes et al. (2009) the formula for inertia is the following,

$$\text{exponential-inertia}(i, j, t_m) = \sum_{\substack{e' \in E_{t_{m-1}} \\ s(e')=i \wedge r(e')=j}} \frac{\ln(2)}{\theta_{\text{half-life}}} \exp \left\{ -(t_m - t_{e'}) \frac{\ln(2)}{\theta_{\text{half-life}}} \right\} \quad (4.2)$$

where the weight of each event $(s(e'), r(e')) = (i, j)$ in the current history of events ($E_{t_{m-1}}$) follows an exponential decay governed by the half-life parameter $\theta_{\text{half-life}}$, which is assumed to be known. The transpired time of the event e' , measured as $t_m - t_{e'}$, is updated at each time point. Because it increases over time, the transpired time decreases the event weight over time. The speed of such decrease depends on the value of the half-life parameter ($\theta_{\text{half-life}}$) that describes the waiting time before the weight of the past event (i, j) halves. Thus, the larger the $\theta_{\text{half-life}}$, the slower will be the decrease in weight and, in turn, long-passed events will keep having a high contribution in the calculation of the statistic, reflecting a long-lasting memory of actors. When a researcher changes the parameters governing the decay, the model statistics (such as the value of inertia) change with it and, in turn, their effect on the event rate changes as well. For this reason, the use of such prespecified half-life parameters should be used with care. This is even more the case because the weight of past events may even decrease with a different shape than with an exponential shape in real life networks. The change of effects due to a change in the memory parameter was already explored by Brandenberger (2018b), where the author shows the different estimated model effects resulting from prespecifying different half-life values.

Another approach to modeling weight decay in relational event data was proposed by Perry and Wolfe (2013), where the past history of events at each time point is divided according to a set of $K + 1$ increasing time widths $\gamma = (\gamma_0, \gamma_1, \dots, \gamma_K)$ and endogenous statistics are calculated within each of the K resulting intervals. For instance, inertia for the k -th interval is calculated as,

$$\text{interval-inertia}(i, j, t_m, k) = \sum_{\substack{e' \in E_{t_{m-1}:} \\ (t_m - t_{e'}) \in (\gamma_{k-1}, \gamma_k]}} \mathbb{I}(s(e') = i, r(e') = j) \quad \text{for } k = 1, \dots, K \quad (4.3)$$

Therefore, the effect of inertia in each interval is estimated and finally described by the vector of effects $\beta_{\text{inertia}} = (\beta_{\text{inertia}_1}, \dots, \beta_{\text{inertia}_K})$. No assumptions are made about the steps (which may either decrease, increase, etc.) and the estimation can be done relatively easy using existing software. Note that there is a clear relation between this step-wise approach and the above weighted approach according to the following equation

$$\begin{bmatrix} \beta_{\text{inertia}_1} \\ \vdots \\ \beta_{\text{inertia}_K} \end{bmatrix} = \begin{bmatrix} \beta_{\text{inertia}} w_1 \\ \vdots \\ \beta_{\text{inertia}} w_K \end{bmatrix} \quad (4.4)$$

where on the right side: (w_1, \dots, w_K) is the step-wise function of the weights of the events which is based on the widths $(\gamma_0, \dots, \gamma_K)$ and assumes that events belonging to the same interval have the same weight, β_{inertia} is the effect of the network dynamic on the event rate. For an extended description of the relation in (4.4) see Appendix B.1. Also the step-wise approach has potential limitations. First, in some applications it may not be natural to assume that the relative importance of a past event is relatively high and one second later (say) its importance drops considerably which is the case in such step-wise models. Second, it is generally unclear how the intervals should be chosen such that the memory (decay) in the data is accurately captured (see also Arena, Mulder, and Leenders (2022) for a related discussion). Finally note that by considering many different intervals for all endogenous effects, the number of unknown parameters can unduly blow up resulting in a tremendous increase of our uncertainty about the model parameters.

In this chapter an alternative methodology is proposed to better learn about past events affecting future events. We assume a continuous, parameterized

decay function which can either be exponential, linear, or step-wise. Each of these functions has a single memory parameter that is optimized using the observed data. A Bayesian test is proposed to determine which decay function (exponential, linear, or step-wise) fits the data best. Thereby, the methodology builds on previous approaches by (i) allowing the weight of past events to decrease in continuous time (as in Brandes et al. (2009)) but at the same time estimate the rate of the decay from the data, and (ii) finding the best fitting shape of the weight decay (as in Perry and Wolfe (2013)) without overparameterizing the model.

Related to the current work, the time sensitivity in relational event modeling has also been discussed in various other studies. The effect of time recency of past interactions was discussed by Tranmer et al. (2015), and a weekend effect was investigated by Amati et al. (2019) in a network of health care organizations, in which authors show the different network mechanisms that can be observed between week days and weekends. In another work, Bianchi and Lomi (2022) study short-term and long-term effects in network dynamics and provide examples on a high-frequency network (financial markets) as well as on a low-frequency network (patient-sharing relations among health care organizations). Furthermore, methods for estimating time-varying networks effects were proposed by Mulder and Leenders (2019), Meijerink-Bosman, Back, et al. (2022) and Meijerink-Bosman, Leenders, et al. (2022) using moving window approaches, and by Fritz et al. (2021) using B-splines.

Furthermore, some work has been done on external decays and on the presence of right-censoring. Stadtfeld and Geyer-Schulz (2011) discussed the problem of using external decay functions in a discrete state space and examined the use of exponential decays combined with an arbitrary threshold on the decay. They observed that despite external decays as well as events of other types might affect the network of events under study, if the Markov process transition rates are defined over very short time spans, the impact of such factors would only be marginal. In another work, Stadtfeld and Block (2017) discussed about the hurdle of right-censoring in relational event networks and proposed a discrete time window approach that overcomes the issues generated from right-censored events.

The remainder of this chapter is organized as follows: in Section 4.2 we introduce parametric memory decay functions and define three potential decays: (i) the one-step decay, (ii) the exponential decay and (iii) the linear decay. Then, in

Section 4.3, we look into the methodological consequences of treating the memory parameters as parameters to be estimated from the data, introduce the use of the profile log-likelihood in relational event models, and finally propose some possible optimization methods which aim to find the maximum likelihood estimate for the memory parameter. In Section 4.4 and 4.5 we show the results on simulated relational event histories as well as on two real case studies.

4.2 Parametric functions for modeling memory decay

Recently-occurred events generally have a larger impact on the next relational event that will occur in a social network than long-past events. To model this we define a weighting function, which is denoted by $w(\gamma_e(t), \theta)$ where:

1. $\gamma_e(t) = t - t_e$ is the transpired time of event e at time t with $t > t_e$;
2. θ is a memory parameter with support $S(\theta) \in \mathbb{R}$, which determine the resulting shape of the decay;
3. the outcome of the weight is a non-negative real number, that is $w(\gamma_e(t), \theta) \in \mathbb{R}_0^+$.

The weight of a past event can reflect to what degree a past event is remembered, and thus, the weighting function can be viewed as an operationalization of the memory decay of actors about past events. For this reason we shall use the term weighting function and memory decay function interchangeably in this chapter.

The above weighting function is then used for computing the endogenous statistics which summarize the past event history at time t . For example, inertia, which is normally computed as the total count (volume) of past events between two actors (i, j) , is now computed as a weighted count of past events weighted according to the chosen weighting function with memory parameter θ , i.e.,

$$\text{weighted-inertia}(i, j, t_m, \theta) = \sum_{e' \in E_{t_{m-1}}} \mathbb{I}(s(e') = i, r(e') = j) w(\gamma_{e'}(t_{m-1}), \theta) \quad (4.5)$$

Note that the transpired time is computed from the time of the previous event t_{m-1} , which is when the waiting time starts for observing the m -th event. During

the waiting time, the weight is assumed to stay constant so that the assumption of constant hazards between events is not violated.

In contrast to previous approaches we assume the memory parameter to be unknown. Regarding the memory function, many possible shapes could be considered. To keep the model computationally feasible however, three parametric functions of the memory decay are considered in this chapter: a one-step decay function, an exponential decay function, and a linear decay function.

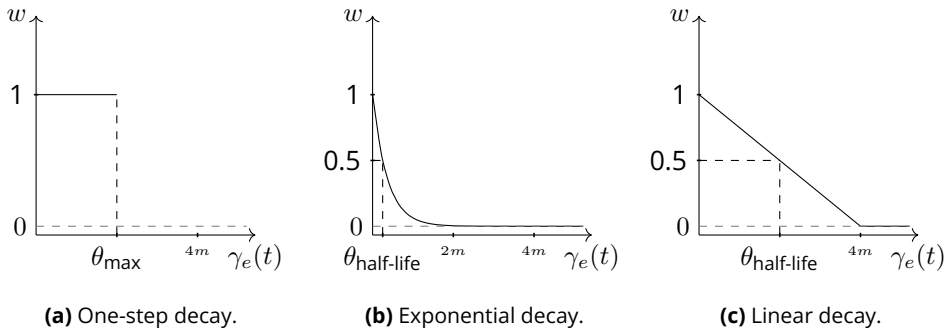


Figure 4.1: Three examples of memory decay: (a) one-step decay where $\theta_{\max} \approx 2$ months and the height of the step is fixed to 1; (b) exponential decay where $\theta_{\text{half-life}} \approx 7$ days; (c) linear decay where $\theta_{\text{half-life}} \approx 2$ months.

One-Step decay

The one-step decay is defined as

$$w_{\text{step}}(\gamma_e(t), \theta_{\max}) = \begin{cases} 1 & \text{if } \gamma_e(t) \leq \theta_{\max} \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

where $\theta_{\max} \in (0, +\infty)$ is the memory parameter and the decay describes a one-step function. Thus, events will contribute to the statistic only if their transpired time is less than the threshold θ_{\max} . Moreover, the model is simplified to the case where the weight is unitary. Note that the interpretation of a coefficient of an endogenous statistic that is computed using a one-step memory function is similar to the interpretation of coefficients of count statistics which ignore memory decay. The only difference is that in the step-wise model only the events with transpired time γ that do not exceed the threshold value θ_{\max} contribute to the rate with a value equal to the corresponding coefficient of the parameter.

Chapter 4

An example of the shape of the one-step decay is shown in Figure 4.1a where $\theta_{\max} \approx 2 \text{ months}$. In this case, weighted inertia between actors i and j would be equal to the total number of past events between i and j in the last 2 months.

Substantively a one-step decay may be appropriate in social networks where actors only have a relatively short-term memory. It may then be reasonable to assume that only the past events within this short window affect the endogenous statistics, and that the past events in this window affect the endogenous variables (approximately) equal. Computationally the one-step model is convenient as we would only need to look back until θ_{\max} to compute the endogenous statistics.

The one-step function was used by Mulder and Leenders (2019) using a pre-specified memory length. Mulder and Leenders (2019) also assumed that network parameters may change over time. This was achieved by estimating the network parameters within a time window which was set equal to the chosen memory length while moving the window over the observed event history. In the current chapter we do not consider a model where network parameters can change over time. Instead we assume that the network parameters are homogeneous over time but, in the case of a one-step decay, we do assume that the past events affect the endogenous statistics until a certain threshold value (i.e., θ_{\max}), which is assumed unknown.

Exponential decay

The functional form for the exponential decay is

$$w_{\text{exp}}(\gamma_e(t), \theta_{\text{half-life}}) = \exp \left\{ -\gamma_e(t) \frac{\ln(2)}{\theta_{\text{half-life}}} \right\} \quad (4.7)$$

for $\gamma_e(t) \in (0, +\infty)$ where $\theta_{\text{half-life}} \in (0, +\infty)$ is the memory parameter that measures the minimum elapsed time after which the event weight is halved. In this formulation we let the weights start decaying from 1 instead of $\frac{\ln(2)}{\theta_{\text{half-life}}}$ as it is defined in Brandes et al. (2009) and this will only affect the scaling of the effects β . One of the possible shapes of an exponential decay is shown in Figure 4.1b where $\theta_{\text{half-life}} \approx 7 \text{ days}$.

The interpretation of a coefficient of an endogenous statistic that is computed using an exponential decay function is slightly more complicated than for a regular count statistic because the contribution of each past event to the rate (and

the hazard) depends on the transpired time since the event was observed. For example, when the coefficient of inertia is equal to 3 and the decay function in Figure 4.1b is considered with a half-life of 7 days which starts at 1, the last event that was observed has a maximal contribution to the rate (and hazard) equal to 3. Furthermore, the contribution of events that were observed approximately 7 days ago contribute with approximately 1.5, and events that were observed approximately 14 days ago contribute with approximately .75 to the rate and hazard.

Theoretically an exponential memory decay implies that the weight reduces to half its value in a fixed amount of time, regardless of the current weight. Furthermore the model assumes that past events are never “forgotten” as in the one-step model. Depending on the context this may be realistic. Computationally the exponential decay is somewhat demanding as it requires one to look back at the entire past history for computing endogenous statistics. However eventually the weights become negligible, and thus, can be approximated as zero.

The exponential model was proposed by Brandes et al. (2009) to model relational events between political actors (e.g., countries) during conflicts. Instead of estimating the half-life parameter from the observed data, the model was fitted using different prespecified half-life parameters. This yielded fairly consistent results in their empirical applications. It is yet unknown whether this result holds in general. This will be explored later in this chapter when fitting the model using misspecified memory parameters.

Linear decay

The linear decay function is defined as

$$w_{\text{linear}}(\gamma_e(t), \theta_{\text{half-life}}) = \left(1 - \frac{1}{2\theta_{\text{half-life}}} \gamma_e(t)\right) \mathbb{I}(\gamma_e(t) \leq 2\theta_{\text{half-life}}) \quad (4.8)$$

for $\gamma_e(t) \in (0, +\infty)$ and with $\theta_{\text{half-life}} \in (0, +\infty)$, which quantifies the time until the weight is halved, similar as in the exponential decay in (4.7). Unlike the exponential decay on the other hand the weight becomes 0 after the transpired time reaches $2\theta_{\text{half-life}}$, similar as the one-step model. In this sense the linear decay model can be seen as a middle ground between the one-step decay and the exponential decay function. An example of a linear weight decay is shown

in Figure 4.1c for $\theta_{\text{half-life}} = 2$ months.

The interpretation of a coefficient of an endogenous statistic that is computed using a linear decay function may be slightly more complicated than for a regular count statistic (because we need to take the transpired time of past events into account) but possibly the interpretation is easier than for statistics with the exponential decay function because linear trends are relatively easy to understand. For example, if inertia would be equal to 3 and the decay function in Figure 4.1c is considered having a half-life of 2 months, the last event that was observed has a maximal contribution to the rate (and hazard) equal to 3, and events that occurred approximately 1 month, 2 months, 3 months and 4 months or more contribute to the event rate with 2.25, 1.5, 0.75, and 0.

The linear decay model may be appropriate where the contribution of past events to the endogenous statistics (and thus to the logarithm of the event rate) is an approximately linear function of the transpired time, which, at some point, becomes approximately zero. Similar as the one-step model, the model is computationally convenient because one would not need to take the entire past history into account in the computation of the endogenous statistics. It may be somewhat less realistic however that the decay is assumed to be exactly 0. To our knowledge a linear decay model has not yet been considered for relational event modeling.

Normalizing decay functions and updating statistics

All the three weight decays start at 1, decay towards zero but are not normalized. However, they can be normalized by multiplying them with a normalizing constant of $\frac{\log(2)}{\theta}$ for the exponential decay and $\frac{1}{\theta}$ for the one-step and the linear decay. The effect of the normalization of the weights directly translates into a re-scaling of the effect β of each endogenous statistic on the event rate, without changing nor re-scaling the estimate of the memory parameter. Normalization is recommended whenever it is needed to compare effects β across different parametrizations or across network dynamics with different memory parameter but following the same parametrization.

When endogenous statistics at each time point are updated according to one of the weight decays introduced in this section, the transpired time of events in the history is updated with respect to the time point that precedes the present one. For instance, if we need to update statistics at time t_m , the history of events

that we are going to consider will be $E_{t_{m-1}}$, that is the collection of events from the onset until and including the event occurred at t_{m-1} and the time we consider to compute the time transpired of each event in the history will be the time of the last event in $E_{t_{m-1}}$, that is t_{m-1} . Therefore, since statistics are assumed to be updated at the last observed time point and not during the waiting time between two subsequent events, no right-censoring has to be taken into account in our analysis and the assumption of constant hazards during waiting times is not violated.

4.3 The Profile log-likelihood in REM

The functions to model memory decay presented in Section 4.2 are three examples of univariate decays that can be embedded in the likelihood function of a Relational Event Model (REM; Butts (2008)) as well as in an Actor-oriented Model (DyNAM; Stadtfeld and Block (2017)). In these decay functions the memory parameter has support in $(0, +\infty)$. Thus, with the purpose of avoiding a constrained optimization for the memory parameter, we can reparametrize it as $\theta = \exp\{\psi\}$, where $\psi \in \mathbb{R}$ is the natural logarithm of the memory parameter θ .

We now consider a sequence E_{t_M} of M relational events occurring among N actors where the likelihood function of a REM, which depends on the memory decay parameter ψ , is written as

$$\mathcal{L}(\beta, \psi; E_{t_M}) = \prod_{m=1}^M \left[\lambda(s_{e_m}, r_{e_m}, X_{e_m}, E_{t_{m-1}}, \beta, \psi) \prod_{e' \in \mathcal{R}} \exp\{-\lambda(s_{e'}, r_{e'}, X_{e'}, E_{t_{m-1}}, \beta, \psi) (t_m - t_{m-1})\} \right] \quad (4.9)$$

where at each time point a vector of endogenous and exogenous statistics in X_{e_m} is available for every possible dyad in the risk set (\mathcal{R}). Although we assume a time-invariant risk set the method can straightforwardly be applied to dynamic risk sets. Parameters β describe the effect of the statistics on the event rate and ψ represents the logarithm of the memory parameter under a specific memory decay, which is assumed to be the same for all the endogenous statistics. In the context of maximization of the likelihood function we are interested in finding the vector of parameters (β, ψ) that maximizes the likelihood given

the observed sequence of events which is equivalent to minimizing the negative log-likelihood:

$$\arg \min_{(\beta, \psi)} \{-\ln \mathcal{L}(\beta, \psi; E_{t_M})\} \quad (4.10)$$

The optimization problem in (4.10) is generally solved by calculating the derivatives of the function up to and including the second order. In the case of a REM with an unknown memory parameter, endogenous statistics are no more sufficient for the estimation of the corresponding vector of effects β , because their value changes depending on the value of the memory parameter ψ . Thus, only the sequence of events can be referred to as sufficient statistic both for the endogenous effects in β and for ψ . Moreover, derivatives for the memory parameter can either increase the computational burden or fail to exist (for instance, in the one-step decay function). In light of this, we can take advantage of the negative profile log-likelihood for a given memory parameter and investigate whether the memory decay assumption is supported from the data and where the minimum potentially lies in. The profile negative log-likelihood for ψ can be written as,

$$-\ln \mathcal{L}_p(\psi) = \min_{\beta} \{-\ln \mathcal{L}(\beta, \psi; E_{t_M})\} \quad (4.11)$$

where the value of $-\ln \mathcal{L}_p(\psi)$ is obtained as the minimum value of the negative log-likelihood where the memory parameter is fixed and the optimization is carried over β (as in a regular REM). Equation (4.11) comes down to one optimization for each fixed value of $\psi \in \mathbb{R}$. If there exists a minimum for $-\ln \mathcal{L}_p(\psi)$, that value will correspond to the global minimum of both ψ and the optimized vector β , thus they will be a solution for the optimization of the negative log-likelihood $-\ln \mathcal{L}(\beta, \psi; E_{t_M})$.

An example of the negative profile log-likelihood based on one randomly simulated relational event history with an exponential memory decay is shown in Figure 4.2 where the function reaches its minimum close to the true value of the log-half-life parameter ($\psi = \ln(4) \approx 1.386$, indicated by the dashed vertical line). The slight deviation from the true value can be explained from random sampling (explored in more detail in the next section).

A drawback of the optimization of the negative Profile log-likelihood is that such methods do not provide a measure for the standard error of the memory parameter nor its covariances with the vector of effects β . A way to estimate the accuracy of the estimate for the memory parameter and the related covariances

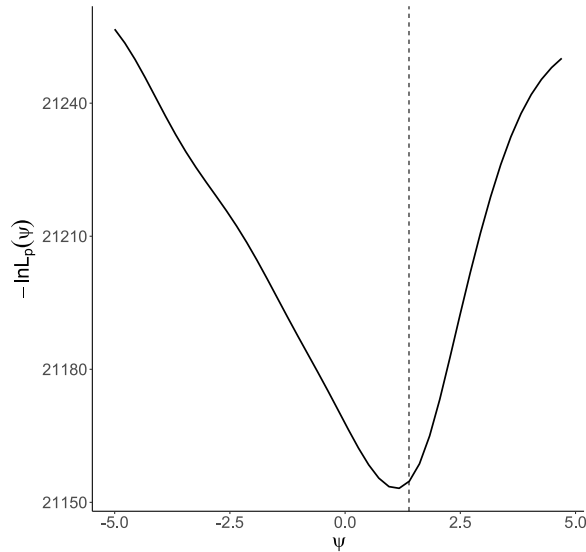


Figure 4.2: Negative profile log-likelihood for the log-half-life parameter (exponential memory decay with true value $\psi = \ln(4) \approx 1.386$, dashed vertical line) for one randomly simulated event history.

consists in embedding the weight decay function in the model and in optimizing the complete log-likelihood in (4.9). However, this approach requires the weight decay function to be differentiable at least twice.

Even though we cannot obtain standard errors for the memory parameters in a straightforward manner, the profile log-likelihood can be used to quantify our relative uncertainty (from a Bayesian perspective) about different models that assume different values for the memory parameter as in a model selection problem. For example, when looking at the example data that was used in Figure 4.2, we could think of a set of models $\mathcal{M}_1 : \psi = -5.0$, $\mathcal{M}_2 : \psi = -4.9$, $\mathcal{M}_3 : \psi = -4.8, \dots, \mathcal{M}_{101} : \psi = 5$. The Bayesian information criterion for model \mathcal{M}_t (BIC; Schwarz (1978)) is then defined by

$$BIC(\mathcal{M}_t) = k \log(M) - 2 \ln \mathcal{L}(\hat{\beta}, \psi; E_{t_M}), \quad (4.12)$$

where $\hat{\beta}$ is the MLE assuming the memory parameter of length k , and ψ is given under model \mathcal{M}_t . Thus, this BIC is directly available using standard statistical software. Consequently the Bayes factor between one model, say, \mathcal{M}_{t_1} assuming a certain value for the memory parameter against another model, say, \mathcal{M}_{t_2} ,

assuming another value (possibly assuming another memory function as well), is then given by

$$BF(\mathcal{M}_{t_1}, \mathcal{M}_{t_2}) = \exp\{BIC(\mathcal{M}_{t_2})/2 - BIC(\mathcal{M}_{t_1})/2\}, \quad (4.13)$$

which quantifies the relative evidence in the data in favor of \mathcal{M}_{t_1} against \mathcal{M}_{t_2} . Thus, via this route we can even test non-nested models having different memory functions and assuming different memory parameters.

4.4 Simulations: synthetic relational event histories with memory decay

Numerical simulations were conducted to explore the bias and the change in fit observed when memory decay values and/or decay parametrization are misspecified. Furthermore, we explored the performance of the Bayes factor to test between models with different memory decays. Finally, a simulation was carried out to investigate the behavior of the estimates in the scenario where the assumption of piece-wise-constant hazard is no longer met. The simulation studies under four different populations will be referred to as Simulation 1, 2, 3 and 4.

Simulation 1: Exponential memory decay

In Simulation 1, 100 relational event histories are generated, each with $M = 5,000$ events occurring among $N = 20$ actors. The log event rate for any dyad $(i, j) \in \mathcal{R}$ at time t is specified as follows:

$$\begin{aligned} \ln \lambda(i, j, t) = & \beta_{\text{Intercept}} + \beta_{\text{Dyadic}_1} \text{Dyadic}_1(i, j) + \\ & + \beta_{\text{Dyadic}_2} \text{Dyadic}_2(i, j) + \beta_{\text{Inertia}} \text{weighted-Inertia}(i, j, t, \theta_{\text{half-life}}) + \\ & + \beta_{\text{Reciprocity}} \text{weighted-Reciprocity}(i, j, t, \theta_{\text{half-life}}) + \quad (4.14) \\ & + \beta_{\text{TClosure}} \text{weighted-TClosure}(i, j, t, \theta_{\text{half-life}}) + \\ & + \beta_{\text{ABAY}} \text{ABAY}(i, j, t) \end{aligned}$$

where Dyadic_1 and Dyadic_2 are two exogenous variables that are time-invariant and asymmetric (i.e. $\text{Dyadic}_1(i, j) \neq \text{Dyadic}_1(j, i)$). Weighted inertia, weighted reciprocity, and weighted transitivity closure (TClosure, based on the definition

presented in Arena, Mulder, and Leenders (2022)) are endogenous statistics based on a weighted count using an exponential memory decay with $\theta_{\text{half-life}} = 4$ (with $\psi = \ln(\theta_{\text{half-life}}) \approx 1.386$). ABAY is an endogenous turn-continuing participation shift (Butts, 2008) which does not follow any memory decay. The vector of true parameters is $(\beta_{\text{Intercept}} = -3.5, \beta_{\text{Dyadic}_1} = 0.5, \beta_{\text{Dyadic}_2} = -0.3, \beta_{\text{Inertia}} = 0.2, \beta_{\text{Reciprocity}} = 0.3, \beta_{\text{TClosure}} = 0.1, \beta_{\text{ABAY}} = 0.2)$.

Simulation 2: Linear memory decay

In Simulation 2, 100 relational event histories are generated, each with $M = 5,000$ events occurring among $N = 20$ actors. The log event rate for any dyad $(i, j) \in \mathcal{R}$ at time t is specified as in (4.14). However, in this simulation the memory decay for weighted inertia, weighted reciprocity, and weighted transitivity closure follows a linear decay with $\theta_{\text{half-life}} = 4$ (with $\psi = \ln(\theta_{\text{half-life}}) \approx 1.386$). The vector of true parameters is the same as the one used in Simulation 1 except for the Intercept which is $\beta_{\text{Intercept}} = -3$.

Simulation 3: One-step memory decay

The same configuration is considered as for Simulation 2 but with a one-step memory decay for weighted inertia, weighted reciprocity and weighted transitivity closure using threshold $\theta_{\text{max}} = 4$ (with $\psi = \ln(\theta_{\text{max}}) \approx 1.386$).

Simulation 4: Exponential memory decay and decreasing hazard

In this simulation, 100 relational event histories are generated, each with $M = 5,000$ events occurring among $N = 20$ actors. To explore the effect of violations of the piece-wise constant hazard assumption, the waiting times are generated from a Weibull distribution where the shape parameter is assumed equal to 0.5. With such value of the shape parameter, hazards decrease over the waiting times. The scale parameter of the Weibull is still a function of the rates and the log event rate for any dyad $(i, j) \in \mathcal{R}$ at time t is specified as in (4.14) with the exception of the Intercept that is assumed $\beta_{\text{Intercept}} = -10$. The weight decay of the endogenous statistics follows the same exponential decay as in Simulation 1 with a half-life parameter $\theta_{\text{half-life}} = 4$ (with $\psi = \ln(\theta_{\text{half-life}}) \approx 1.386$).

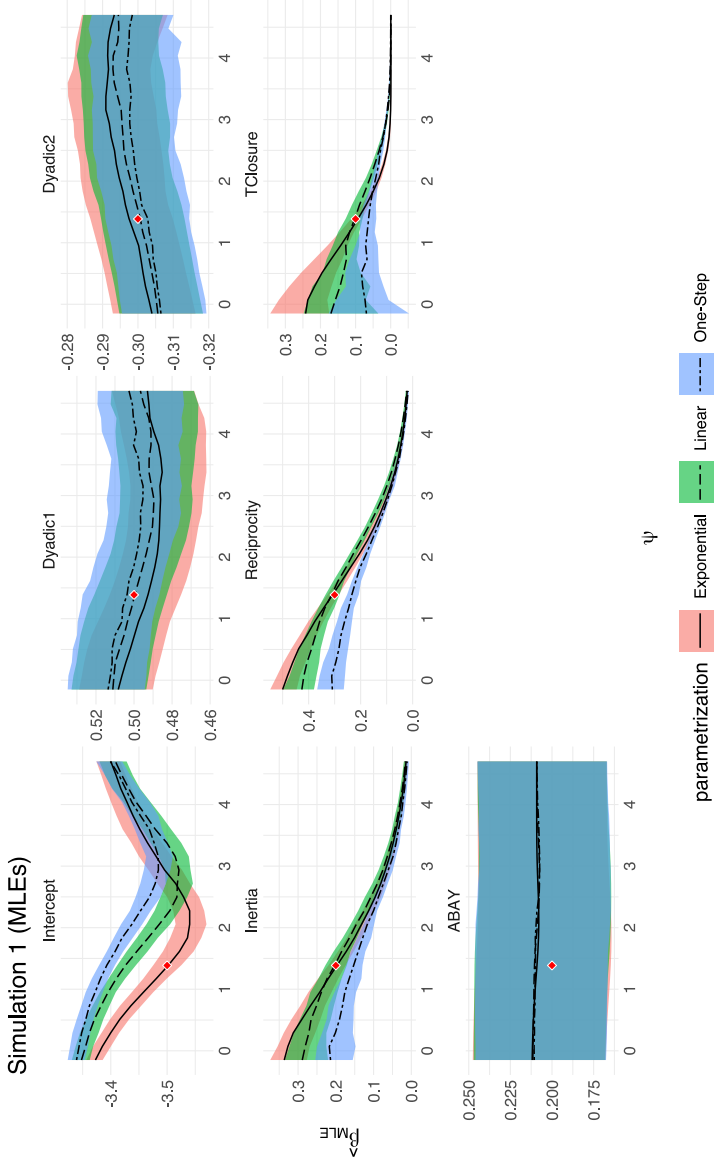


Figure 4.3: Simulation 1 (the true decay is exponential). Trend of the maximum likelihood estimates over the logarithm of the memory parameter, ψ , and under each of the three memory decays (exponential, linear and one-step). The shaded regions delimit the first and the third quartile of the distribution (based on 100 simulated event sequences) of the estimated effect β over ψ . The black lines show the trend of the median of each effect across the 100 simulations, and they have a different line type according to each parametrization. The diamond-shaped point marks the coordinates of the true memory parameter ($\ln(4) \approx 1.386$) and the true value of each specific effect.

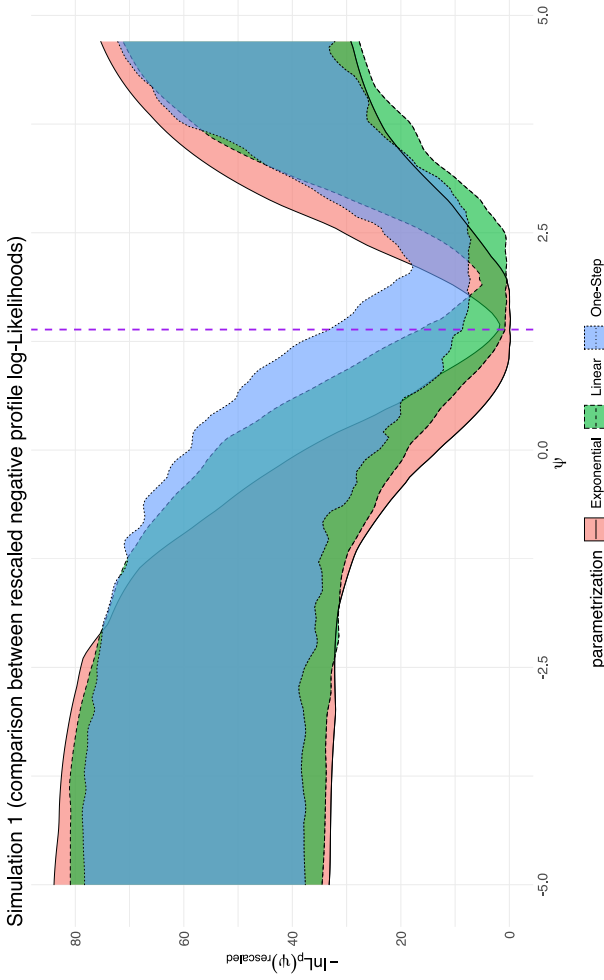


Figure 4.4: Simulation 1 (comparison between rescaled negative profile log-likelihoods across simulations). The y-axis is the $-\ln(L_p)$ that is rescaled based on the global minimum across the three parametrizations and the local minimum within each parametrization. The figure shows three regions with different line types, one per each parametrization. Each region represents the (rescaled) value assumed by the 95% of the simulations in one parametrization across different values of the memory parameter (here on its logarithmic scale on the x-axis). The vertical dashed bold line marks the true value for the logarithm of the memory parameter ($\psi = \ln(4) \approx 1.386$). The three weight decays result in showing about the same evidence towards small values of the memory parameter (negative values on the logarithmic scale) as well as towards larger values (greater than 3.0 on the logarithmic scale). However, when in the neighborhood close to the true value of the memory parameter, the tree parametrizations show a diverging evidence, with the Exponential model being the lowest, which is the true parametrization used in the generation of the 100 event sequences.

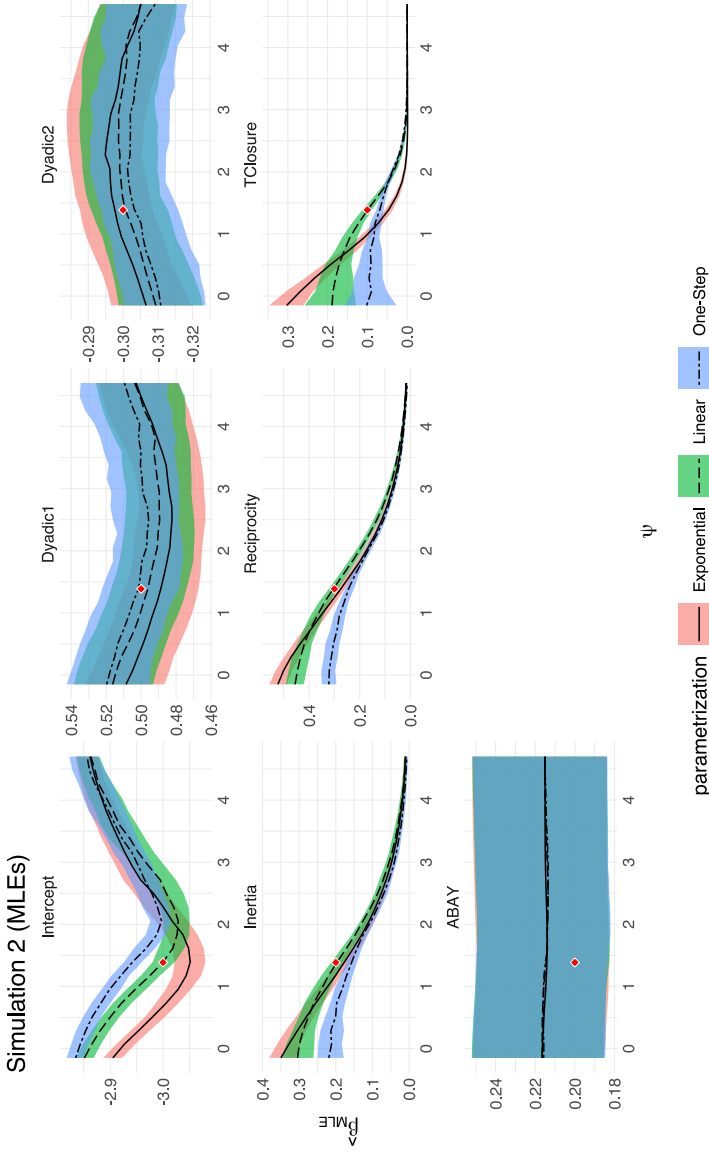


Figure 4.5: Simulation 2 (the true decay is linear). Trend of the maximum likelihood estimates over the logarithm of the memory parameter, ψ , and under each of the three memory decays (exponential, linear and one-step). The shaded regions delimit the first and the third quartile of the distribution (based on 100 simulated event sequences) of the estimated effect β over ψ . The black lines show the trend of the median of each effect across the 100 simulations, and they have a different line type according to each parametrization. The diamond-shaped point marks the coordinates of the true memory parameter (ln(4) \approx 1.386) and the true value of each specific effect.

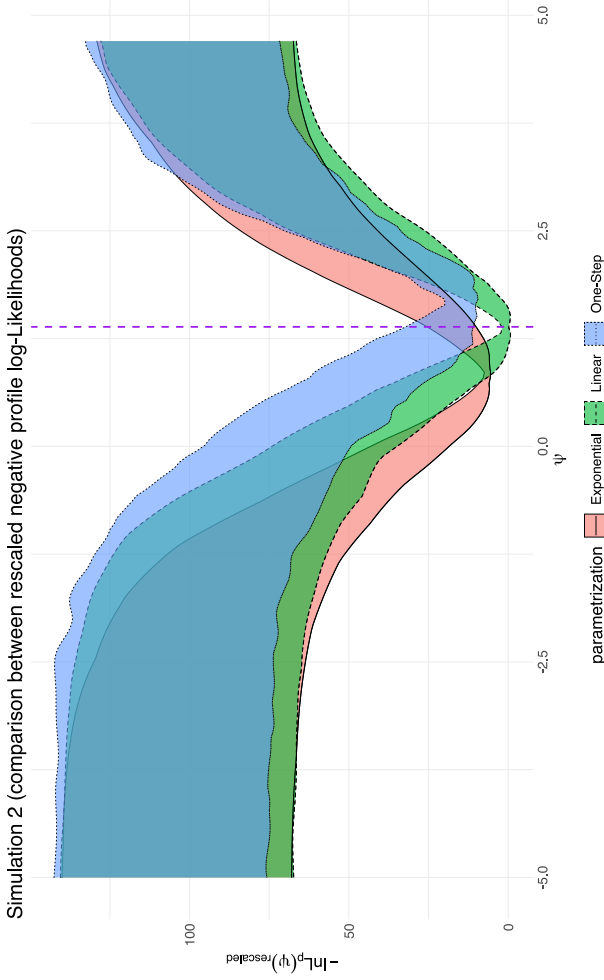


Figure 4.6: Simulation 2 (comparison between rescaled negative profile log-likelihoods across simulations). The y-axis is the $-\ln(L_p)$ that is rescaled based on the global minimum across the three parametrizations and the local minimum within each parametrization. The figure shows three regions with different line types, one per each parametrization. Each region represents the (rescaled) value assumed by the 95% of the simulations in one parametrization across different values of the memory parameter (here on its logarithmic scale on the x-axis). The vertical dashed bold line marks the true value for the logarithm of the memory parameter ($\psi = \ln(4) \approx 1.386$). The three weight decays result in showing about the same evidence towards small values of the memory parameter (negative values on the logarithmic scale) as well as towards larger values (greater than 3.0 on the logarithmic scale). However, when in the neighborhood close to the true value of the memory parameter, the tree parametrizations show a diverging evidence, with the Linear model being the lowest, which is the true parametrization used in the generation of the 100 event sequences.

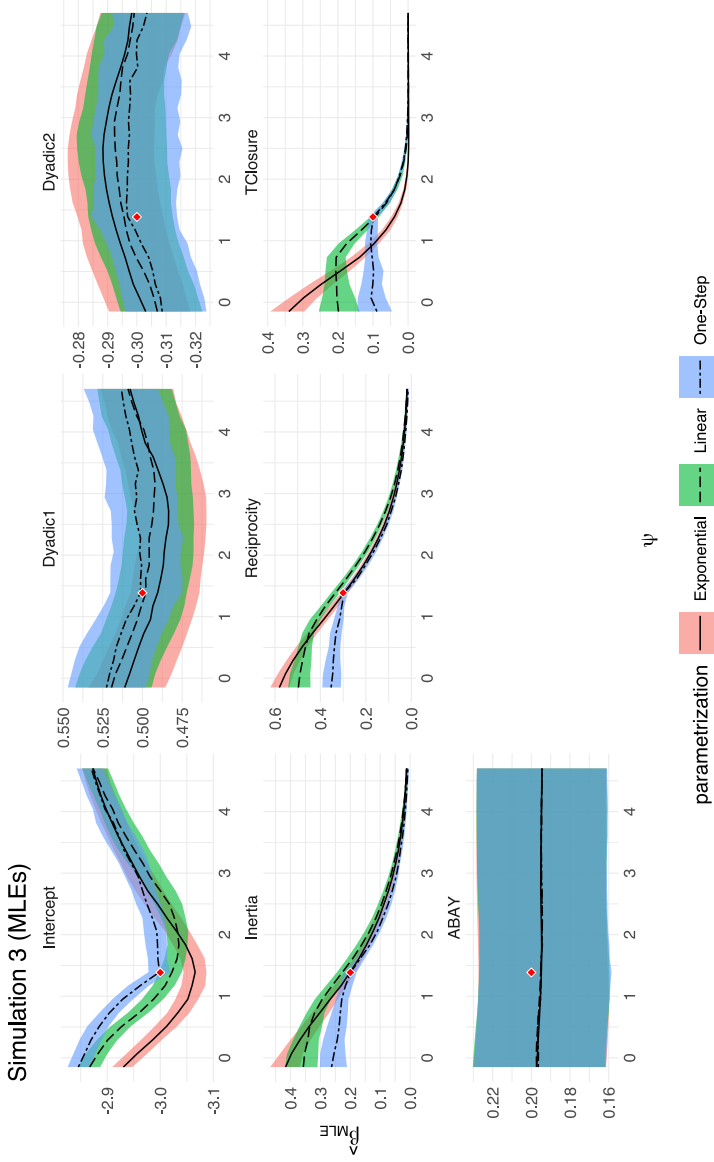


Figure 4.7: Simulation 3 (the true decay is one-step). Trend of the maximum likelihood estimates over the logarithm of the memory parameter, ψ , and under each of the three memory decays (exponential, linear and one-step). The shaded regions delimit the first and the third quartile of the distribution (based on 100 simulated event sequences) of the estimated effect β over ψ . The black lines show the trend of the median of each effect across the 100 simulations, and they have a different line type according to each parametrization. The diamond-shaped point marks the coordinates of the true memory parameter ($\ln(4) \approx 1.386$) and the true value of each specific effect.

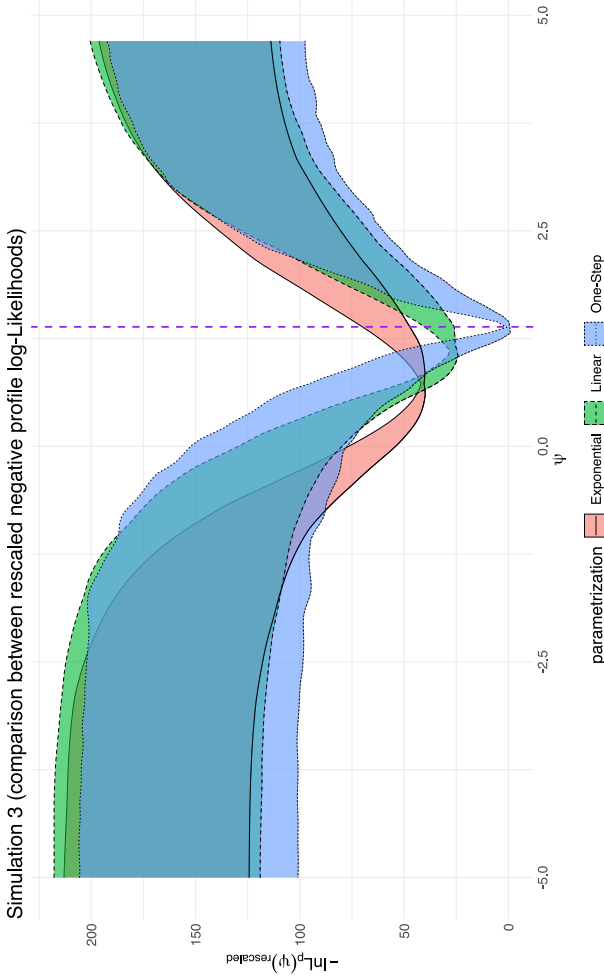


Figure 4.8: Simulation 3 (comparison between rescaled negative profile log-likelihoods across simulations). The y-axis is the $-\ln(L_p)$ that is rescaled based on the global minimum across the three parametrizations and the local minimum within each parametrization. The figure shows three regions with different line types, one per each parametrization. Each region represents the (rescaled) value assumed by the 95% of the simulations in one parametrization across different values of the memory parameter (here on its logarithmic scale on the x-axis). The vertical dashed bold line marks the true value for the logarithm of the memory parameter ($\psi = \ln(4) \approx 1.386$). The three weight decays result in showing about the same evidence towards small values of the memory parameter (negative values on the logarithmic scale) as well as towards larger values (greater than 3.0 on the logarithmic scale). However, when in the neighborhood close to the true value of the memory parameter, the tree parametrizations show a diverging evidence, with the One-Step model being the lowest, which is the true parametrization used in the generation of the 100 event sequences.

4.4.1 Exploring bias based on a misspecified memory decay

The first purpose of the first three simulation studies is to understand whether and to what degree maximum likelihood estimates of the effects of the statistics in a REM are affected by the value of the memory parameter. This is important as memory decay is often prespecified in an ad hoc manner.

In Figures 4.3, 4.5 and 4.7, the trend for each estimated effect over the logarithm of the memory parameter, ψ , is shown under the three memory decays (exponential, linear and one-step). The shaded areas delimit the first and the third quartile of the distribution (based on the 100 simulations) of the estimated effect at any given value of ψ . The black lines show the trend of the median of each effect over the 100 simulations, and they have a different line type according to each parametrization. The diamond-shaped point marks the coordinates of the true memory parameter ($\ln(4) \approx 1.386$) and the true value of each specific effect. In all the simulations we see that all the endogenous variables which were assumed to follow a memory decay (Inertia, Reciprocity and Transitivity Closure) as well as the Intercept are considerably affected by bias in the case of a misspecified memory parameter. Only if (i) the memory model assumed is the correct one and (ii) the memory parameter is around its maximum likelihood estimate, the distribution of the estimates across the simulations tend to concentrate around the true β . As a consequence of this, it is evident that by not accounting for the memory parameter in the maximum likelihood optimization of a REM as well as not investigating different memory decays will likely lead the researcher to biased estimates.

Furthermore, Figure 4.4, 4.6 and 4.8 show a comparison between rescaled negative profile log-likelihoods across 100 simulations within each simulation study (Simulation 1, 2 and 3). In each simulation study, each of the 100 simulated event sequences were optimized under each of the three parametrizations of the memory decay (Exponential, Linear and One-Step). Therefore, for each event sequence a negative profile log-likelihood is found for each parameterization.

The set of negative profile log-likelihoods under each parametrization and per each event sequence are then rescaled based on the global minimum across the three parametrizations and the local minimum within each parametrization, resulting in the new scale on the y-axis ($-\ln(L_p)_{\text{rescaled}}$). Each Figure shows three regions with different line types, one per each parametrization. Each region

represents the (rescaled) value assumed by the 95% of the simulations in one parametrization across different values of the log-memory parameter (on the x-axis). The vertical dashed bold line marks the true value for the logarithm of the memory parameter ($\psi = \ln(4) \approx 1.386$).

The results suggest that the proposed method using the profile log likelihood results in accurate estimates of the memory parameter in a well-specified model. Moreover, the true data generating model results in the best fit overall. Finally we see that in all three simulation studies, the three parametrizations result in roughly the same fit when towards small values of the memory parameter (negative values on the logarithmic scale) as well as towards larger values (greater than 3.0 on the logarithmic scale). This implies that in the case of a complete mismatch of the true decay parameter and the decay parameter that is used for model fitting, it does not matter which decay function would be used.

When including memory decay parameters in Relational Event Models it is also important to verify whether the assumption of proportional hazards is violated or not. In order to accomplish this, we analyzed the Schoenfeld's residuals (Schoenfeld, 1982) calculated for those endogenous statistics which are assumed to follow a weight decay (Inertia, Reciprocity and Transitivity Closure). In each of the three simulations, residuals in the 100 replicates distributed around zero and showed no trend over time, which implies that the assumption of proportional hazard was not violated.

4.4.2 Testing different decay functions via the Bayes Factor

The second purpose of simulation studies 1, 2 and 3 is to explore the performance of the Bayes Factor (BF) to test different memory models. We measured the relative evidence in favor of the true model given the simulated relational event sequence. In each of the three simulations, for every generated event sequence, we computed the Bayes Factor of the model of the true weight decay function against the best model under the remaining other two decays using Equation (4.13) for each simulated dataset. The formulation of the Bayes Factor is such that $BF(\mathcal{M}_1, \mathcal{M}_2) > 1 (< 1)$ implies evidence for \mathcal{M}_1 (\mathcal{M}_2). If the Bayes factor is on the log scale the cut-off value equals 0. By investigating the distribution of the Bayes Factor across the 100 sequences for all the simulations we get insights how well the Bayes factor can distinguish between different memory models. Figure 4.9 plots the distribution of the Bayes factors.

Chapter 4

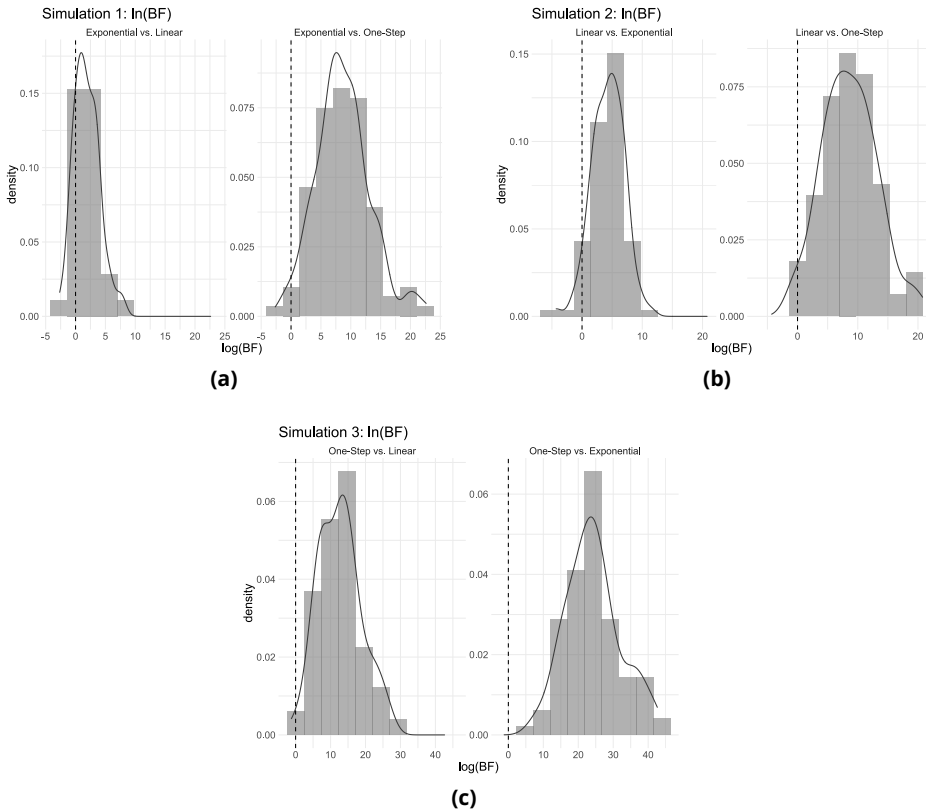


Figure 4.9: Distribution of $\ln(BF)$ (where $\ln(BF) > 0$ translates to as evidence in favor of the true model) : (a) in Simulation 1 the true weight decay is exponential, thus the two Bayes Factors are $BF(\mathcal{M}_{\text{Exponential}}, \mathcal{M}_{\text{Linear}})$ and $BF(\mathcal{M}_{\text{Exponential}}, \mathcal{M}_{\text{One-Step}})$ and the number of simulations where $\ln(BF) > 0$ is respectively 80 and 98 out of 100; (b) in Simulation 2 the true weight decay is linear, thus the two Bayes Factors are $BF(\mathcal{M}_{\text{Linear}}, \mathcal{M}_{\text{Exponential}})$ and $BF(\mathcal{M}_{\text{Linear}}, \mathcal{M}_{\text{One-Step}})$ and the number of simulations where $\ln(BF) > 0$ is respectively 95 and 98 out of 100; (c) in Simulation 3 the true weight decay is one-step, thus the two Bayes Factors are $BF(\mathcal{M}_{\text{One-Step}}, \mathcal{M}_{\text{Linear}})$ and $BF(\mathcal{M}_{\text{One-Step}}, \mathcal{M}_{\text{Exponential}})$ and the number of simulations where $\ln(BF) > 0$ is respectively 99 and 100 out of 100.

In each of the three simulations, the distributions of the $\ln(BF)$ concentrates on positive values (> 0), with at least the 95% of the generated sequences supporting the true memory decay and values of the Bayes Factor (on its logarithmic scale) show a somewhat strong evidence as well. The worst performance was observed in the case of Simulation 1 (exponential decay) when it was compared with a linear decay, where the Bayes factor pointed towards the linear model in 20% of the simulated data sets. This result shows that the linear de-

Simulation 4 (distribution of MLEs across 100 simulations)

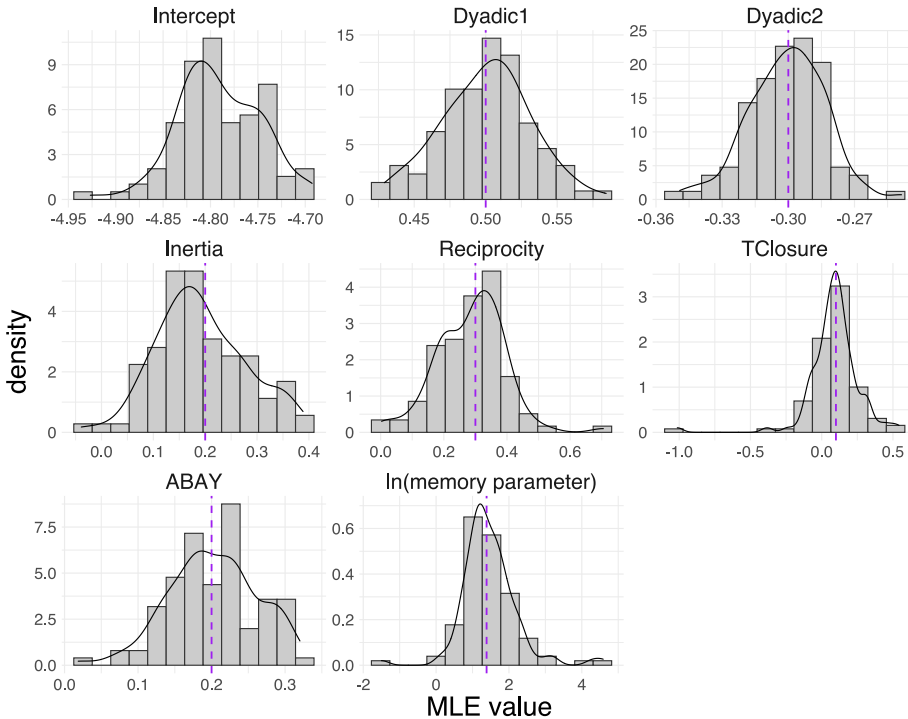


Figure 4.10: Simulation 4 (the true waiting time in the 100 simulated event sequences is distributed as a Weibull with shape parameter equal to 0.5). Distribution of the maximum likelihood estimates of the effects of the statistics as well as the memory parameter in a REM (with piece-wise constant hazard assumption). Each vertical dashed line corresponds to the true value of each effect.

can potentially mimic an underlying exponential decay. Of course note that because the Bayes factor is consistent, the error probability would go to 0 when increasing the sample size of the simulated event history and the evidence for the true model would go to infinity.

4.4.3 Exploring estimation errors due to a misspecified hazard function

In Simulation 4, we explore the potential estimation error of the proposed relational event model with an exponential memory decay while the data were generated using Weibull waiting times (which violate the piece-wise-constant hazard assumption). Figure 4.10 shows the distribution of the endogenous and

exogenous REM parameters β based on the optimized memory parameter ψ for each generated dataset. The Figure shows that the distribution of the maximum likelihood estimate of each effect is generally centered around its true value which suggests practically no clear estimation error, except for the intercept which roughly ranges between -4.95 and -4.70 while the true value was -10.0 . This suggests that the decreasing hazard under the data generating model is mainly picked up by the intercept of the fitted model while leaving the other model parameters generally unchanged on average. This suggests that the estimated network parameters are safe to interpret in the case of a misspecified model due to a decreasing hazard underlying the data.

4.5 Investigate memory decay in empirical relational event networks

In this section we apply the methodology to the following two real-life case studies:

- A network of Indian socio-political actors sending demands to one another;
- A network of students sending text messages among each other.

The goal was to learn which memory decay function best describes the weight decrease of past events when modeling future events using endogenous network statistics. We also illustrate the impact of misspecified memory parameters on the network coefficients. Moreover, the fit and predictive performance of the best fitting memory decay model was compared with a model which ignores memory decay to study the importance of memory decay in empirical relational event networks. Finally we provide insights about the computational costs of the approach for relational events with different numbers of actors and different numbers of events.

4.5.1 Demands among Indian socio-political actors

The Indian data were retrieved from the ICEWS (Integrated Crisis Early Warning System) (Boschee et al., 2015). This database is available in the Harvard Dataverse repository and it collects relational events that describe interactions (found in news articles) occurring between socio-political actors all over the world. We focus our analysis on the sequence of requests that were recorded

	Decay			w/o memory
	Exponential	Linear	One-Step	
BIC	56753.55	56775.5	56870.08	60659.31
$\ln(BF)$	-	10.97	58.27	1952.88

Table 4.1: (Indian data) BIC of the best model (where the memory parameter is optimized) under each of the three memory decays (exponential, linear and one-step) and for the model w/o memory. The lowest BIC is the one of the exponential model (56753.55), and the two log-Bayes-Factor are calculated based on the following model comparisons: $BF(\mathcal{M}_{\text{Exponential}}, \mathcal{M}_{\text{Linear}})$, $BF(\mathcal{M}_{\text{Exponential}}, \mathcal{M}_{\text{One-Step}})$ and $BF(\mathcal{M}_{\text{Exponential}}, \mathcal{M}_{\text{w/o memory}})$.

within the Indian territory. In the original data, such requests were further classified in humanitarian, military or economic ones but we avoid such distinction in our analysis.

The relational event sequence consists of $M = 7567$ demands recorded between June 2012 and April 2020 and sent among the ten most active actor types: citizens, government, police, member of the Judiciary, India, Indian National Congress Party, Bharatiya Janata Party, ministry, education sector, and "other authorities". The time variable is recorded at a daily level, therefore events that co-occurred are considered evenly spaced throughout a day and the memory parameter is measured in days. The logarithm of the rate (λ) for the demand sent by actor i to actor j at time t is modeled as

$$\begin{aligned} \ln \lambda(i, j, t) = & \beta_{\text{Intercept}} + \beta_{\text{Inertia}} \text{weighted-Inertia}(i, j, t, \psi) + \\ & + \beta_{\text{Reciprocity}} \text{weighted-Reciprocity}(i, j, t, \psi) + \\ & + \beta_{\text{TClosure}} \text{weighted-TClosure}(i, j, t, \psi) + \beta_{\text{ABAY}} \text{ABAY}(i, j, t) \end{aligned} \quad (4.15)$$

where weighted-Inertia, weighted-Reciprocity and weighted-Transitivity Closure (TClosure) are assumed to follow a weight decay governed by ψ , the logarithm of the memory parameter.

We first investigated the three weight decays presented in Section 4.2 (exponential, linear and one-step decay) by optimizing the negative log-likelihood for each model (a plot of the $-\ln L_p(\psi)$ is shown in Figure 4.11). Finally we chose the best fitting model among the three models, that is the one with the lowest BIC. In Table 4.1 the BIC's of the three best models are reported along with the

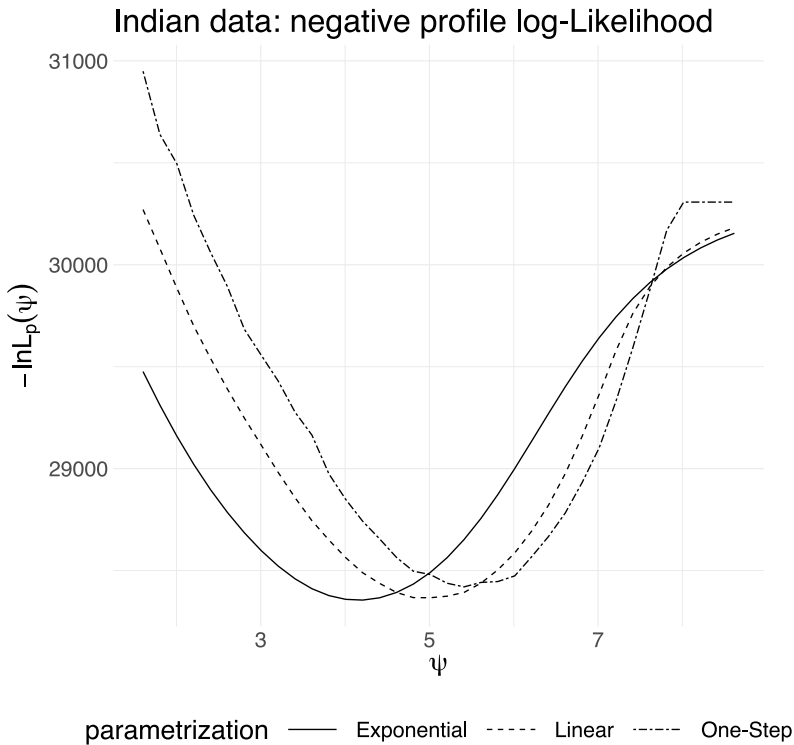


Figure 4.11: (Indian data) negative profile log-likelihood ($-\ln L_p(\psi)$) under each of the three memory decays (exponential, linear and one-step).

BIC of a model in which no memory decay was specified. In the same table, the log-Bayes-Factor is calculated by considering as reference the model with the lowest BIC among the three, that is the exponential one. We see that there is convincing evidence in the data in favor of the exponential decay model as the data were approximately $\exp(10.97)$, $\exp(58.27)$, and $\exp(1952.88)$ times more likely under the exponential model than under the linear decay model, the one-step model, and the model without memory, respectively. Thus, we choose to continue the analysis with the exponential decay model. In Figure 4.12 the trend of the MLEs is plotted over the logarithm of the memory parameter (ψ) for the exponential decay model. Again, we see a considerable impact of the choice of the memory parameter which suggests that choosing the memory parameter in an ad hoc manner is not advised. For example, we see that transitivity closure can vary from approximate 0 to more than 1 within the considered range of the memory parameter.

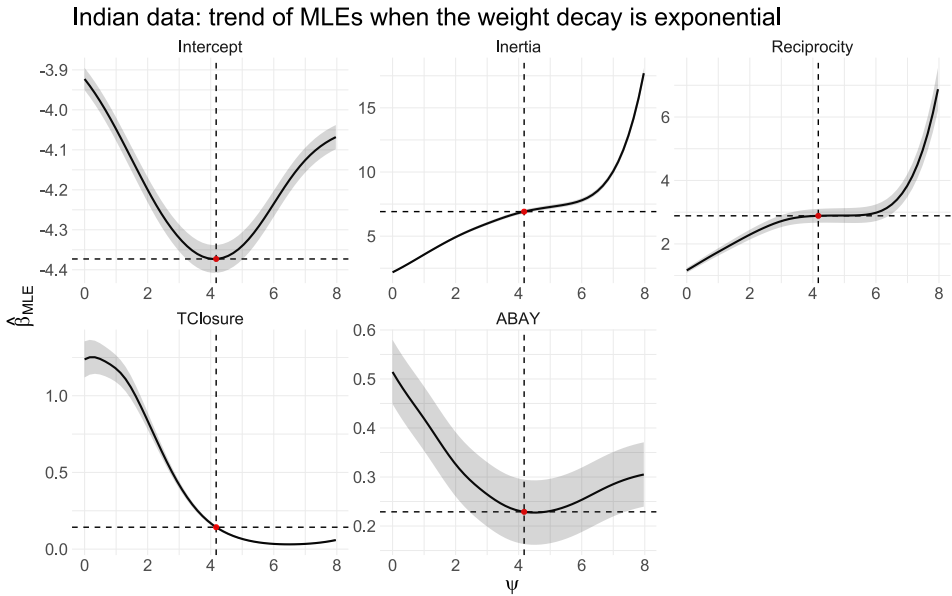


Figure 4.12: (Indian data) trend of the maximum likelihood estimates (MLEs) for the exponential decay over ψ (logarithm of the memory parameter). The dashed black lines in each plot mark the estimate for the log-memory-parameter $\hat{\psi}_{MLE}$ (vertical lines) and the estimates of the effects β (horizontal lines) at the corresponding $\hat{\psi}_{MLE}$. The shaded regions are the 95% confidence intervals for the effects β estimated at any value of ψ .

The estimated half-life of the exponential memory decay in this network is $\exp(\hat{\psi}) \approx 64$ days. Thus the weight of past requests tends to halve after about 2 months. This case study has been already shown in Arena, Mulder, and Leenders (2022) where a semi-parametric strategy was applied to model memory decay by means of an ensemble of many step-wise decay models. In that analysis, which does not make parametric assumptions about memory decay function (and is therefore computationally much more expensive), the shape of the decay also followed an approximate exponential shape, which is the same as we find here using the parametric approaches presented in this chapter.

In Table 4.2, the estimates of the effects β at the optimized memory parameter are reported. When interpreting these coefficients it is important to note that the memory function was normalized such that the surface underneath the line equals 1. Given the half-life parameter of 64 days, this implies that the function in Figure 4.1b would be multiplied with $\ln(2)/64$. Thus, given the estimated inertia effect of 6.9, the contribution of the last observed event to the last observed dyad is equal to a factor of $\exp(6.9 \times (\ln(2)/64)) \approx 1.08$, i.e., an increase

	$\hat{\psi}$	Intercept	Exponential decay			
			Inertia	Reciprocity	TClosure	ABAY
$\hat{\beta}$	4.156	-4.373	6.912	2.888	0.146	0.229
$se(\hat{\beta})$	-	0.018	0.074	0.111	0.005	0.033

Table 4.2: (Indian data) Maximum likelihood estimates for the exponential decay. The estimate of the logarithm of the memory parameter is 4.156, that is an half-life of $\exp(4.156) \approx 64days$. Estimates of effects β are all significant.

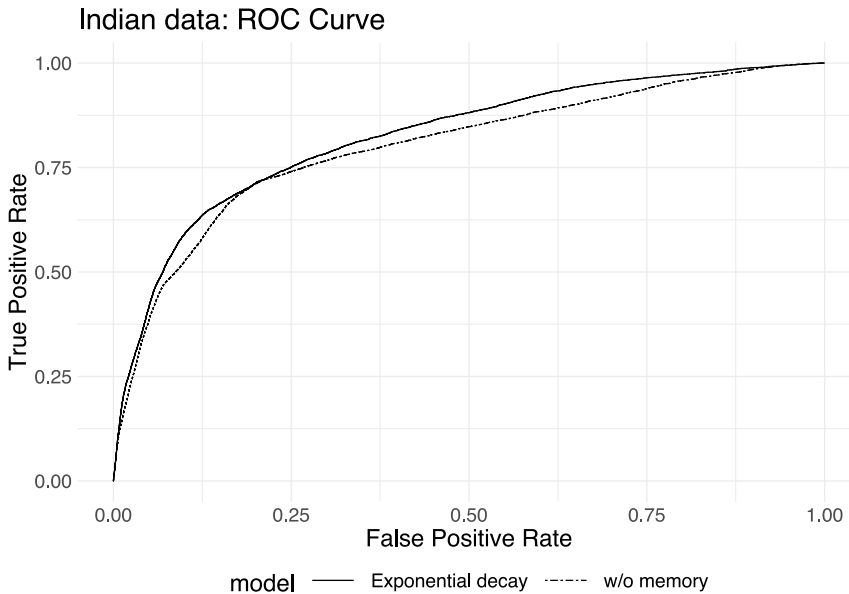


Figure 4.13: (Indian data) ROC curve of model with exponential memory decay and model without memory

of 8% (which is the maximal contribution), and if an event was observed for this dyad approximately 64 days ago, this would have resulted in a contribution to the rate with a factor of $\exp(6.9 \times (\ln(2)/64) \times 0.5) \approx 1.04$, i.e., an increase of 4%.

To illustrate the importance of modeling memory decay, we evaluated the predictive performance of the best fitting relational event model with an exponential decay function and compared it with a relational event model which ignores memory decay by giving all past events an equal weight. The plot in Figure 4.13 shows the ROC curves of both models which clearly shows the superior performance of the model with an exponential memory decay function.

Network of sms	# Actors	# Events
1 cluster	23	3678
2 clusters	53	7311
8 clusters	199	13943

Table 4.3: (Sms data) Dimensions of the three sub-networks used in the example.

4.5.2 Text messages among students

The sms data consist of a sequence of text messages sent among a group of university students (freshmen) during a period of four weeks. The original event sequence is part of the interaction data collected in the Copenhagen Networks Study (Sapiezynski et al., 2019) and it consists of 568 students and 24333 events (number of text messages).

We ran the same analysis on three sub-sequences of events (increasing in both number of actors and number of events) so as to have a better understanding of the computational complexity of the methodology presented in this chapter as well as to explore the method for networks of different sizes, both in terms of the number of actors and the number of events. For the selection of the three sub-sequences: (i) we ran a clustering algorithm that works on the optimization of a modularity score (Clauset et al., 2004; Csardi & Nepusz, 2006), (ii) we sorted the clusters based on the length of the event sequences, from the longest to the shortest, and (iii) we considered three sub-networks where the first was based on the first cluster of actors, the second was based on the first two clusters and, finally, the third was based on the first eight clusters. Each sub-sequence of events also includes the interactions between actors belonging to a different cluster. In Table 4.3, we show the size of each network in terms of number of students (# Actors) and text messages sent (# Events).

In all the three selected relational event sequences, the time variable is available as timestamp which is converted to hours transpired since the beginning of the observation time. Thus the memory parameter will be measured in hours as well. In addition to the event sequence we also know the gender of the students and whether they are friends on Facebook or not. With these two information we specified two dyadic variables: (1) SameGender which assumes the value 1 if the two actors interacting have the same gender, 0 otherwise; (2) FBfriends that assumes the value 1 where the sender and the receiver of the text message are

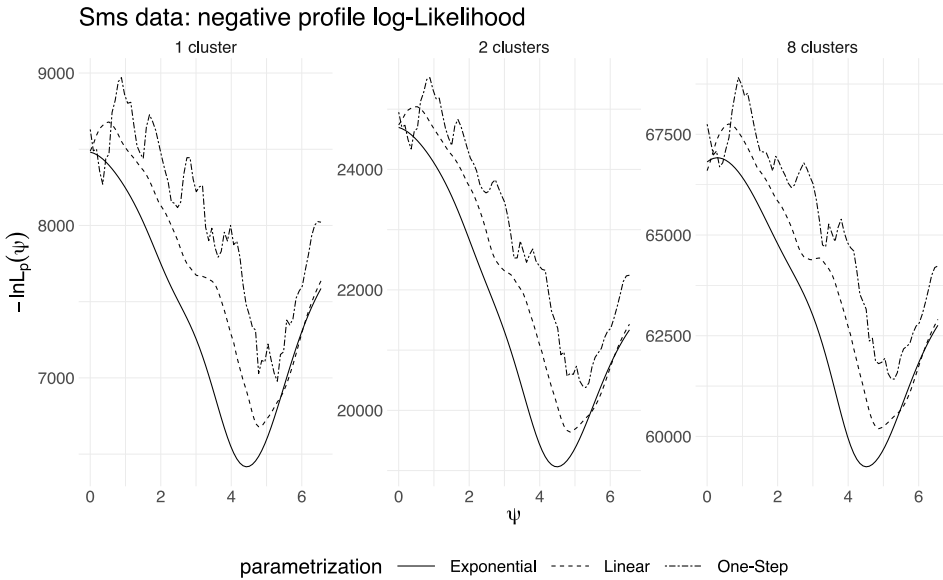


Figure 4.14: (Sms data) negative profile log-likelihood ($-\ln L_p(\psi)$) under each of the three memory decays (exponential, linear and one-step) and for each sub-network (1 cluster, 2 clusters and 8 clusters).

		Decay			w/o memory
		Exponential	Linear	One-Step	
1 cluster	BIC	12882.49	13408.32	14004.85	16092.17
	$\ln(BF)$	-	262.92	561.18	1604.84
2 clusters	BIC	38178.28	39332.40	40810.29	44540.81
	$\ln(BF)$	-	577.06	1316.01	3181.26
8 clusters	BIC	118548.10	120438.30	122899.70	128520.70
	$\ln(BF)$	-	945.12	2175.85	4986.32

Table 4.4: (Sms data) Per each sub-network (1 cluster, 2 clusters, 8 clusters) the BIC of the best model (where the memory parameter is optimized) under each of the three memory decays (exponential, linear and one-step) and for the model w/o memory. In all the sub-networks, The lowest BIC is the one of the exponential model, and the two log-Bayes-Factor are calculated based on the following model comparisons: $BF(\mathcal{M}_{\text{Exponential}}, \mathcal{M}_{\text{Linear}})$, $BF(\mathcal{M}_{\text{Exponential}}, \mathcal{M}_{\text{One-Step}})$ and $BF(\mathcal{M}_{\text{Exponential}}, \mathcal{M}_{\text{w/o memory}})$.

friends on Facebook, 0 otherwise.

We specify the same model for the three sub-networks; thus, the logarithm

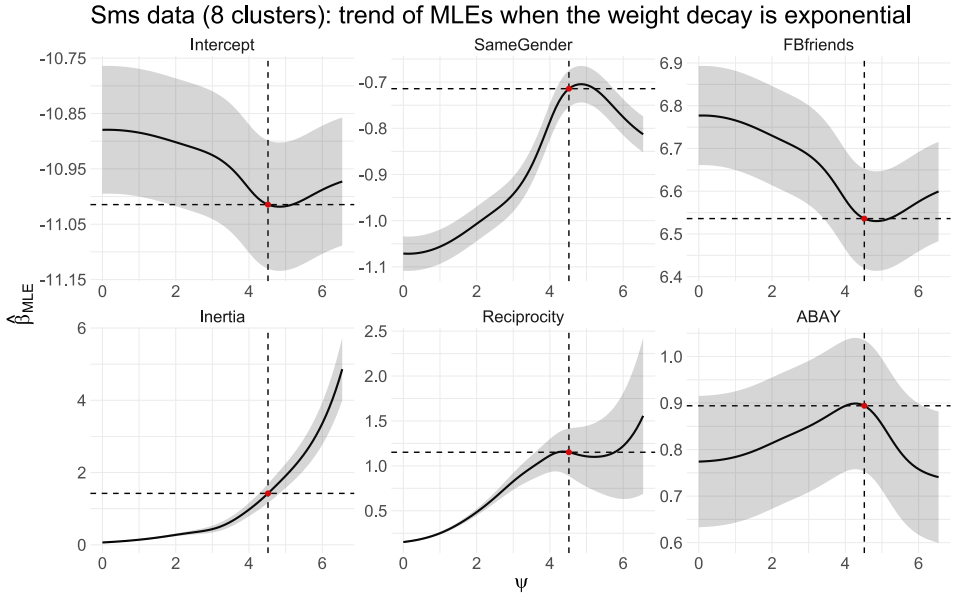


Figure 4.15: Sms data (8 clusters): trend of the maximum likelihood estimates (MLEs) for the exponential decay over ψ (logarithm of the memory parameter). The dashed black lines in each plot mark the estimate for the log-memory-parameter $\hat{\psi}_{MLE}$ (vertical lines) and the estimates of the effects β (horizontal lines) at the corresponding $\hat{\psi}_{MLE}$. The shaded regions are the confidence intervals at 0.95 for the effects β estimated at any value of ψ . For the Transitivity Closure (TClosure) estimates are plotted for an interval of ψ to make the trend much more readable.

of the rate (λ) for a text message sent by actor i to actor j at time t is modeled as

$$\begin{aligned} \ln \lambda(i, j, t) = & \beta_{\text{Intercept}} + \beta_{\text{SameGender}} \text{SameGender}(i, j) + \\ & + \beta_{\text{FBfriends}} \text{FBfriends}(i, j) + \beta_{\text{Inertia}} \text{weighted-Inertia}(i, j, t, \psi) + \\ & + \beta_{\text{Reciprocity}} \text{weighted-Reciprocity}(i, j, t, \psi) + \beta_{\text{ABAY}} \text{ABAY}(i, j, t) \end{aligned} \quad (4.16)$$

Also in this application weighted-Inertia and weighted-Reciprocity are assumed to follow a weight decay governed by ψ and the three parametrizations were examined, in the same fashion as with the Indian data.

In Figure 4.14, we see that for all three networks the negative profile log-likelihood for the exponential model is lowest, suggesting the best fit for an exponential decay. This is also confirmed by comparing the three optima, thus by the BIC's and the Bayes Factors shown in Table 4.4. For the exponential model, the trend of the estimates β over ψ for the model with eight clusters are shown

		Exponential decay						
		$\hat{\psi}$	Intercept	SameGender	FBfriends	Inertia	Reciprocity	ABAY
1 cluster	$\hat{\beta}$	4.438	-8.706	-1.357	5.745	0.662	0.984	-0.665
	$se(\hat{\beta})$	-	0.214	0.056	0.215	0.175	0.175	0.139
2 clusters	$\hat{\beta}$	4.491	-9.214	-1.392	5.691	1.228	0.852	-0.407
	$se(\hat{\beta})$	-	0.109	0.036	0.110	0.134	0.134	0.110
8 clusters	$\hat{\beta}$	4.519	-11.015	-0.714	6.536	1.423	1.152	0.894
	$se(\hat{\beta})$	-	0.059	0.020	0.059	0.135	0.135	0.072

Table 4.5: (Sms data) Maximum likelihood estimates for the exponential decay in each of the three sub-networks (1 cluster, 2 clusters, 8 clusters). The estimate of the logarithm of the memory parameter ($\exp(\hat{\psi})$) ranges approximately between *84hrs* and *92hrs* in the three networks. Estimates of effects β are overall significant.

in Figure 4.15, where the dot marks the maximum likelihood for each effect at the $\hat{\psi}_{MLE}$. The trends of the other two networks (1 cluster and 2 clusters) are shown in Appendix B.2. The optimal half-life ranges between approximately 84 hours and 92 hours, which implies that text messages become half as important to predict future observations after a little bit more than 3.5 days.

The maximum likelihood estimates for the exponential model regarding the three networks are shown in Table 4.5 using normalized decay functions, which should be taken into account when interpreting the endogenous effects. For example, for the network based on 8 clusters, inertia was estimated to be equal to 1.423, which implies that the rate of the last observed dyad is multiplied with $\exp(1.423 \times (\ln(2) / \exp(4.519))) \approx 1.011$, which implies an increase of about 1.1%, and if an event was observed, say, $\exp(4.519) \approx 92$ hours ago, this would have resulted in an increase of about 0.5% of the rate. Furthermore, both the variables SameGender and FBfriends show clear effects on the event rate. A negative effect for SameGender suggests that the text messages are more likely to be exchanged between students of a different gender. Indeed, the parameter $\hat{\beta}_{SameGender} = -0.714$ suggests that the hazard (sms) rate for a dyad where both actors have the same gender will be $(\exp(-0.714) - 1) \approx -51\%$ lower than the rate in which the two actors have different gender (holding all the other statistics constant). The effect for the variable FBfriends shows that the sequence of sms is strongly represented by students that are also friends on Facebook, since such variable results to have a large positive effect on the sms rate.

To get more insights about the predictive performance of the best fitting expo-

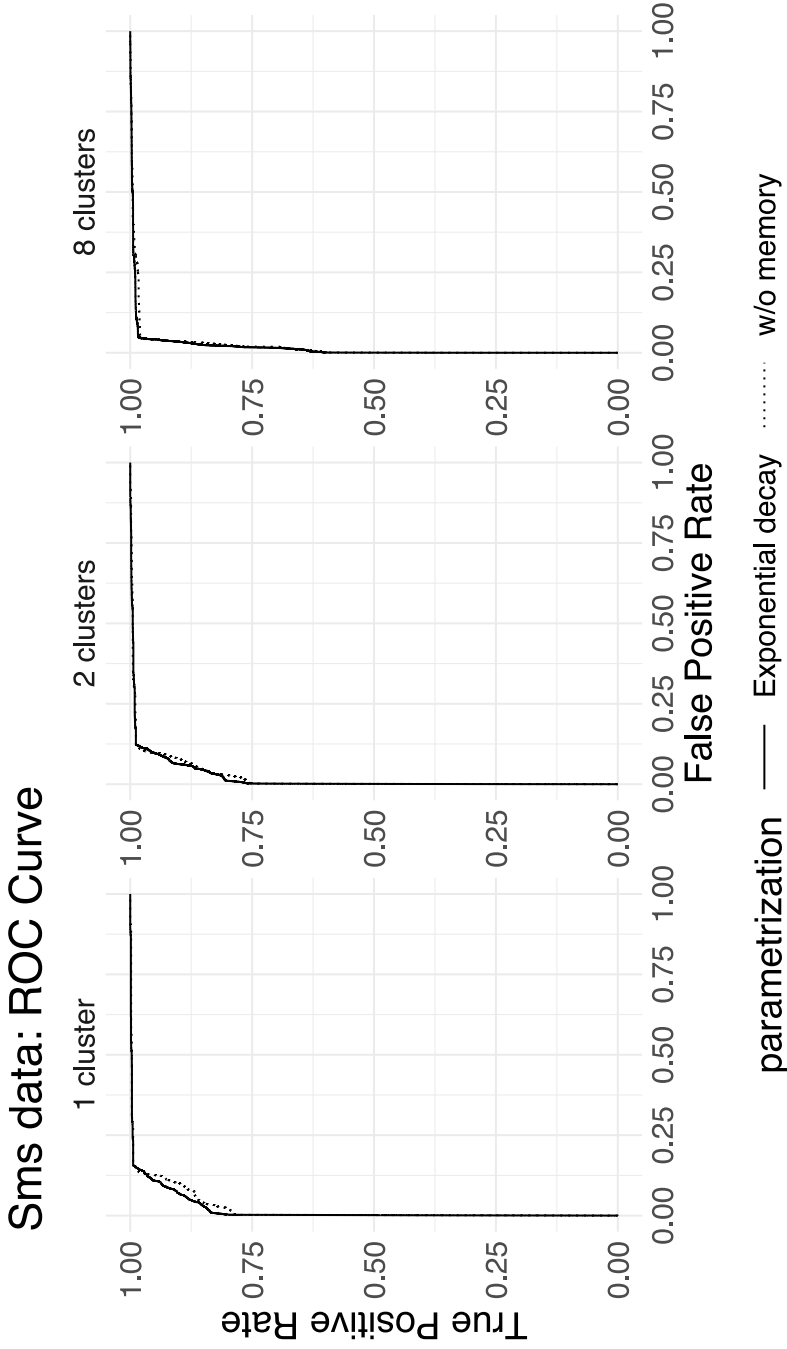


Figure 4.16: Sms data ROC curve of model with exponential memory decay and model without memory for the three sub-networks (1 cluster, 2 clusters and 8 clusters).

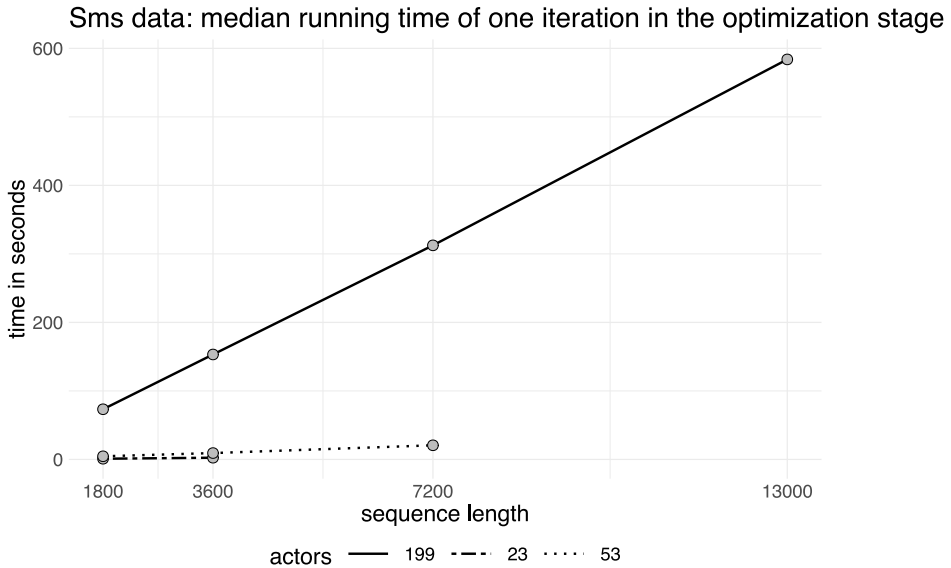


Figure 4.17: (Sms data): Median running time of one iteration in the optimization stage. The model that is estimated in the optimization stage is the same one introduced in the data example in Section 4.5.2. The time is reported in seconds (on the y-axis), the sequence length is the number of events considered (on the x-axis). The first and the second sub-sequence (networks with one and two clusters) do not show running times for larger lengths because they reach their maximum length (see Table 4.3).

ponential decay model in comparison to a model which ignores memory decay, we checked the ROC curves. These are displayed in Figure 4.16. Again we see that there is an improvement in predictive performance of the exponential decay model over the model without memory decay. The improvement is relatively small for the data based on 8 clusters.

The final objective of this study was to provide insights about the computational time of the methodology by considering the needed time for one iteration in the estimation stage. We ran one iteration for the optimization of the parameter β by fixing the log-memory parameter ψ to the maximum likelihood estimate from the sub-sequence with 8 clusters. We considered the three sub-sequences and for each sub-sequence, we run the estimation of the effects β over an increasing sequence length of 1800, 3600, 7200 and, 13000 events. Per each sequence length, we run the estimation 100 times. In Figure 4.17 the median (over 100 repetitions) running time for one iteration in the optimization stage is shown across both sub-networks (one, two and eight clusters) and sequence lengths. The model that is estimated in the optimization stage is the

same one introduced in the data example in Section 4.5.2. The time is reported in seconds on the y-axis. The sequence length on the x-axis is the number of events for the specific sub-network. The first and the second sub-sequence (networks with one and two clusters) do not show running times for lengths larger than their maximum length (see Table 4.3). We observe that the median running time of one iteration follows a linear increasing trend with a slope that becomes larger for the networks with a higher number of actors. This shows that the computation is feasible for small networks of 23 actors to fairly large networks of 199 actors.

4.6 Discussion

In the literature on relational event networks, weight decay functions have been used to capture the decreasing importance of past events to compute endogenous network statistics as a function of the transpired time. To achieve this, a parametric function can be chosen to model the decay of the weight of past events together with a chosen memory parameter that describes the speed of the decay or memory length. In previous studies both the decay function and the memory parameter governing have often been prespecified in a fairly ad hoc manner. As an alternative, the method presented in this chapter allows one to find the best fitting decay function and memory parameter using the observed sequence of relational events by inspecting several parametric decay functions.

The simulation studies and empirical applications in this chapter showed that a misspecification of the shape of the memory decay and/or a misspecification of the memory parameter lead to biased estimates of the effects of (endogenous) statistics, and consequently this may result in incorrect conclusions about the temporal interaction behavior in the network. This was shown by visualizing the trends of the estimated effects as a function of the memory parameter. For this reason, it is not recommended to use memory functions or memory parameters that are arbitrarily specified. Instead we recommend to optimize the decay using the observed data as our studies revealed that such biases are generally avoided in that case. Hence, we recommend network researchers who are interested to learn how the past affects the future in relational event networks to estimate the memory decay in the endogenous statistics using the proposed methodology.

Chapter 4

Of course, this chapter only considered three possible parametric functions to capture memory decay, all using one memory parameter. Many more single-parameter functions could be considered.

Moreover, the methodology could be extended to functions with two or more unknown parameters (e.g., smoothed-one step decay, negative power decay, hyperbolic-like decay which also allows for a long-term memory plateau). We leave this for future research. Decays depending on a multiple parameters of course add complexity both to the optimization stage and to the interpretation of the parameters. For this reason, the use of univariate memory models may be preferred as a first step to study memory decay in empirical relational event networks.

Another important direction for future research is to improve the estimation of the memory parameter that allows a quantification of its uncertainty, and how this transcends to the network parameters. Both classical likelihood methods as well as full Bayesian approaches could be considered for this purpose. Furthermore, even though our simulation revealed that the model is fairly robust against violations of the piece-wise constant hazard assumption, the potential impact of such misspecifications would be useful to explore in more depth in future research.

Finally we note that the code for the processing of the original data along with the application of the methodology presented in this chapter are available in the OSF repository with identifier DOI: 10.17605/OSF.IO/FD9QX (also reachable at <https://doi.org/10.17605/OSF.IO/FD9QX>).

The software developed to run the method discussed in this work will be available in the R package `bremory` Arena (2022a).

WEIGHTING THE PAST: AN EXTENDED RELATIONAL EVENT MODEL FOR NEGATIVE AND POSITIVE EVENTS

5

Abstract

In a time-ordered sequence of relational events, the sentiment of each interaction describes a qualitative characteristic of the relational event. The additional information about the sentiment of an event allows the researcher to learn better and interpret the dynamics governing the actors' decision-making process. For instance, at school, if "child A annoys child B," the act of A annoying B is labeled as a hostile action (negative sentiment). In a future event, if "child A hugs child B," the act of A hugging B will be labeled as a positive action (positive sentiment). In this example, both relational events have the same sender A and receiver B, but the sentiment of each interaction is different. For social network researchers, accounting for the two sentiments in the modeling framework is essential for a thorough understanding of the dynamics between actors in social networks and for theory development. In this chapter, we introduce a novel modeling framework where the probability of the next event sentiment is modeled separately from the next dyadic event and sentiment-driven parametric memory decays are embedded. We provide tools for estimating and testing model parameters and for learning how much longer (or shorter) negative past events influence the future relative to positive past events. Finally, we discuss the application of the new modeling framework on a real event sequence of attacks and trades between players of an online strategy game.

This chapter will be submitted to a scientific journal in a similar form as Arena, G., Mulder, J., & Leenders, R. Th. A. J. Weighting the past: An extended relational event model for negative and positive events.

5.1 Introduction

Understanding, explaining, and predicting social interaction behavior of actors in a temporal network has been an important focus for social network research. Time-stamped relational observations between actors in a network, also called relational events, are an increasingly popular source of data for this purpose (Butts, 2008; R. T. A. J. Leenders et al., 2016; Perry & Wolfe, 2013). These data sequences are chronologically ordered observations stating who is interacting with whom in the network and when. These data sources capture valuable information to study how social interaction dynamics unfolds over time and how the past affects the future. Empirical examples of such data include the relational events between countries using digital news reports (Brandes et al., 2009), email messages between colleagues in organizations (Mulder & Leenders, 2019), digital and in-person interactions between university students (Sapiezynski et al., 2019), social interactions between players in an online strategy game (Hajibaghery et al., 2018), face-to-face interactions between freshmen students (Meijerink-Bosman, Back, et al., 2022), radio communication between astronauts and the mission control center during the Apollo 13 mission (Karimova et al., 2022), to name a few.

Relational event data can naturally be analyzed using relational event models (Butts, 2008). These types of models summarize the past relational event history between actors in so-called endogenous statistics, such as inertia between actor i and j (quantifying the volume of past events from actor i to j), reciprocity between actor i and j (quantifying the volume of past events from j to i), or transitivity closure (quantifying the number of 2-paths between actors in the past), and exogenous statistics (such as actors' attributes). Because past events that recently occurred are likely to have a larger impact on future interactions than events that have long passed, it is common to weigh the importance of past events when computing the endogenous statistics according to the time that has transpired since the events were observed (Brandenberger, 2018b; Brandes et al., 2009). A relational event model can then be used to estimate the relative importance of these statistics to explain the observed relational event sequence and to better understand social interaction dynamics in a temporal network.

Dyadic events characterizing endogenous statistics can be assumed to follow an exponential weight decay. Studies in literature usually set the half-life pa-

parameter of the weight decay to an a priori value and the choice of such value is a result of some theory related to the field of study or has a mere exploratory purpose (Brandenberger, 2018b; Brandes et al., 2009).

Arena et al. (2023) showed that predefining half-lives might be valuable for gathering initial insights on whether and how much the effect of statistics on the event rate changes as the influence of past events becomes more (less) persistent by defining a larger (smaller) half-life. However, they show that an incorrect specification of the memory parameter might result in biased effects of endogenous statistics on the event rate. The solution they propose is to optimize the effects of the statistics along with the memory parameter for a given parametric weight decay. This way, the estimate of the memory parameter will describe the best weight decay under the observed data and allow for testing on the memory parameter itself.

In addition to information about the actors who are involved in the observed relational events and the time the events were observed, relational event data also often contains a type or sentiment of each event, be it the verb describing the interaction itself or some adjective or noun connoting the action. For example, the relational events between countries analyzed by Brandes et al., 2009 contained both negative (e.g., a boycott) and positive events (e.g., a trade). Similarly, the social interactions between players in an online strategy game analyzed by Hajibagheri et al., 2018 allowed for each player to attack or donate resources to other players. The sentiment of past events can play an important role in social interaction behavior because a boycott of a country by another country or a player attacking another player's resource fields is likely to yield very different subsequent interaction dynamics than if these countries had started a new trade or the player had donated resources to the other. The type of an event doesn't solely describe the sentiment of an interaction but it also gives more room to both theory building and new modeling frameworks that account for it. A central question for social network research is then how relational events of different sentiments affect the social interaction behavior between actors in the future? The literature has shown that hostile past events usually have a negative impact on the hazard rate than positive past events (Brandes et al., 2009) but it is generally unclear how long exactly negative and positive past events affect future social interaction. This chapter presents a relational event model that can be used to address this research question.

The contribution of the current chapter is two-fold. First, we extend the re-

lational event model by redefining the endogenous statistics such that past events are weighted according to the time that has transpired and its sentiment. The weight of past events decays using an exponential function with an unknown half-life parameter that is different for negative and positive past events to study how long past events are “remembered” depending on their sentiment. Consequently endogenous statistics are split into multiple statistics for different sentiments. For instance, if the event types are dichotomized in “negative” and “positive”, the transitivity closure might be described by four different patterns: (i) where all events are positive, (ii) where all events are negative, (iii) where the first event is positive and the second is negative, (iv) where the first event is negative and the second is positive. Second, we extend the relational event model with a Probit regression model for the sentiment of the observed relational events conditionally on the actors that are involved in the event and the time of the event. Thus, the joint model can be used to explain social interaction dynamics between actors by appropriately weighting the past negative and positive events and for predicting when the next event is likely to be observed, which actors are likely to be involved, and whether the sentiment of the next event is positive or negative. Furthermore, we can understand whether the weight decay of past events differs based on the sentiment (type) of the event and, how differently the next dyad and the next event type are influenced by a set of network dynamics and/or exogenous characteristics of the network.

The proposed model builds on previously proposed models. Butts (2008) proposed a model for modeling relational event with sentiments by modeling the rate parameter for each specific dyad and each sentiment. The current model captures the sentiment conditionally on the observed dyad, following Brandes et al. (2009). An advantage of considering a conditional approach is that the size of the risk set does not increase which is useful as large risk sets and large samples (which are common in practice) can easily result in computational and memory problems. Furthermore, it was noted by Lerner et al. (2013) that this approach may result in more natural social interaction dynamics. An important difference with the current model and Brandes et al. (2009)'s approach is that we use different memory decay parameters for different sentiments (to be able to study whether the length of the impact of negative past events differs from that of positive events) and we propose a methodology for estimating and testing different decay parameters given the observed data. Furthermore, optimizing the memory decay from the observed data was considered by Arena et al.

(2023) with the exception that here we consider events of different sentiments having different memory parameters to estimate and test.

The chapter is organized as follows: in Section 2, we introduce the SentiREM and discuss about modeling the next event sentiment separately from the next dyad. We continue in the same section with assuming a Probit regression for modeling dichotomized positive and negative events, discussing the optimization of the model parameters and presenting different possible tests on model parameters in a Bayesian fashion. In Section 3, we present simulation studies and show the behavior of the estimated parameters over different proportions for the two event types in the network and across different lengths of the event sequence. In Section 4, we estimate the SentiREM on an observed sequence of attacks and trades among players of a real-time strategy online game. We discuss the testing on model parameters, the bias of effects after misspecification of memory parameters and, finally, the interpretation of the effects of the two models. Finally, we discuss the key features of the methodology presented in this chapter, a possible limitation due to the assumption of dichotomized sentiments in the network of events and we also propose some idea about possible future works around the topic of memory decay in relational event data with sentiment.

5.2 SentiREM: A model for relational events with sentiments

Consider a time-ordered sequence of M relational events, any event e' that may happen is described by the triple $(s_{e'}, r_{e'}, c_{e'})$, respectively sender, receiver, and sentiment (type) of the event which we assume here to be either positive ($c_{e'} = p$) or negative ($c_{e'} = n$). Thus, the likelihood function for the observed sequence of events E_{t_M} is written as,

$$\mathcal{L}(\beta, \psi; E_{t_M}, \mathbf{X}) = \prod_{m=1}^M \left[\lambda(s_{e_m}, r_{e_m}, c_{e_m}, X_{e_m}, E_{t_{m-1}}, \beta, \psi) \prod_{e' \in \mathcal{R}} \exp \left\{ -\lambda(s_{e'}, r_{e'}, c_{e'}, X_{e'}, E_{t_{m-1}}, \beta, \psi) (t_m - t_{m-1}) \right\} \right] \quad (5.1)$$

where, at any time point t_m , the log-rate of every possible event e' in the risk set (\mathcal{R}) is a linear function of a vector of endogenous and exogenous statistics,

\mathbf{u}_{e', t_m} and parameters β that describe the effect of such statistics on the event rate,

$$\ln \lambda(s_{e'}, r_{e'}, c_{e'}, X_{e'}, E_{t_{m-1}}, \beta, \psi) = \mathbf{u}_{e', t_m}(\psi, E_{t_{m-1}}, \mathbf{X})\beta \quad (5.2)$$

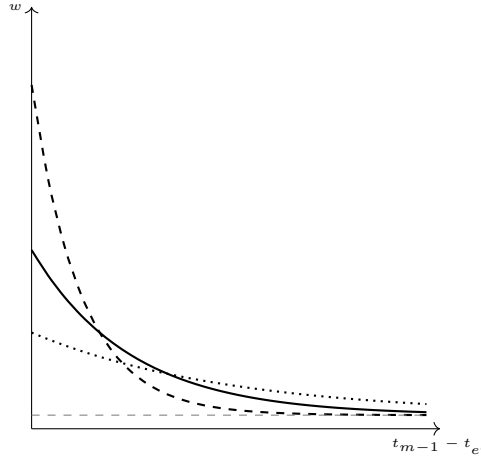


Figure 5.1: Example of exponential decay at different values of the half-life parameter: the dashed line has half-life $\exp(\psi) = 0.5$, the solid line has half-life $\exp(\psi) = 1$, the dotted line has half-life $\exp(\psi) = 2$. As the half-life value increases, the trend decreases slowly and the maximum weight lowers (because of the normalizing constant $\ln(2)/\exp(\psi)$ that makes the area under the decay sum 1).

The exogenous statistics in \mathbf{u}_{e', t_m} can be time-varying or constant and are based on information provided by \mathbf{X} . Whereas, the endogenous statistics quantify network dynamics over time and are based on the history of previous events, $E_{t_{m-1}}$, whose weights change based on the time elapsed since their occurrence. The weight of a past dyadic event e' transpired before t_{m-1} is assumed to follow an exponential decay function

$$w(t_{m-1} - t_{e'}, \psi) = \frac{\ln(2)}{\exp(\psi)} \exp \left\{ -(t_{m-1} - t_{e'}) \frac{\ln(2)}{\exp(\psi)} \right\} \quad (5.3)$$

where $t_{m-1} - t_{e'}$ is the time elapsed at t_{m-1} since the occurrence of the event e' at $t_{e'}$ and the steepness of the decay is governed by the log-memory parameter ψ , with $\exp(\psi)$ being the value of the half-life which describes at which value of the transpired time the weight of an event is halved. An example on what the exponential decay looks like is shown in Figure 5.1, where the shape of the decay is represented at three (increasing) values of the half-life, that are $\exp(\psi) = (0.5, 1, 2)$. As the half-life value increases, we observe that the trend

decreases its steepness becoming slower and the maximum weight adjusts to lower values. Such adjustment is due to the normalizing constant $\ln(2)/\exp(\psi)$ that makes the area under the decay sum 1. We assumed that the shape of the decay is exponential for all the sentiments modeled in the network but the memory parameter is different across sentiments, that is $\psi_p \neq \psi_n$ in a network with only two discrete sentiments. We therefore refer to the vector of log-memory parameters as $\psi = (\psi_p, \psi_n)$, with $\psi \in \mathbb{R}^2$. The anti-logarithm of such vector is the vector of memory parameters and we consider their reparametrization on the logarithmic scale so as to carry the optimization of the likelihood without constraints on any parameter.

In (5.1), the presence of event types increases the size of the risk set by a factor equal to the number of observed types. In order to explain why this happens, we consider the definition of a relational event e' as a triple $(s_{e'}, r_{e'}, c_{e'})$. Whenever there is only one event type (i.e. $c_{e'}$ is not measured or it is the same for all the events in the network), then the risk set \mathcal{R} consists of all the sender/receiver combinations and its size is generally calculated as $N(N - 1)$, where N is the number of actors that can be the sender or the receiver of an interaction. In this case, we model the hazard rate of the directed dyad $(s_{e'}, r_{e'})$ at time t , that is $\lambda(s_{e'}, r_{e'}, t)$. Whereas, if two or more event types (sentiments) are observed, then the risk set \mathcal{R} consists of all the sender/receiver/type combinations and its size is $N(N - 1)C$, where C is the number of observed types. This boils down to defining an hazard rate function for the sentiment-based dyad $(s_{e'}, r_{e'}, c_{e'})$ at time t , namely $\lambda(s_{e'}, r_{e'}, c_{e'}, t)$. A significant disadvantage of such model is given by the increasing size of the risk set which causes a higher computational burden for both the calculation of the statistics, their storage in memory, and the estimation of the model parameters.

The likelihood in (5.1) can be read as the product of the conditional probabilities of observing each event e_m at its time t_m of occurrence, given the sequence of already observed events $E_{t_{m-1}}$,

$$\prod_{m=1}^M p(s_{e_m}, r_{e_m}, c_{e_m}, t_{e_m} | E_{t_{m-1}}; \beta, \psi, \mathbf{X}) \quad (5.4)$$

The conditional probability of a specific event at t_m can be re-written as

$$\begin{aligned} & p(s_{e_m}, r_{e_m}, c_{e_m}, t_{e_m} | E_{t_{m-1}}; \boldsymbol{\beta}, \boldsymbol{\psi}, \mathbf{X}) = \\ & = p(c_{e_m} | s_{e_m}, r_{e_m}, t_{e_m}, E_{t_{m-1}}; \boldsymbol{\beta}^{(c)}, \boldsymbol{\psi}^{(c)}, \mathbf{X}) p(s_{e_m}, r_{e_m}, t_{e_m} | E_{t_{m-1}}; \boldsymbol{\beta}^{(sr)}, \boldsymbol{\psi}^{(sr)}, \mathbf{X}) \end{aligned} \quad (5.5)$$

in which the next event sentiment is modeled after conditioning to the next observed dyad. As a result of the conditional probabilities in (5.5), the set of parameters $\boldsymbol{\beta}$ as well as the memory parameters $\boldsymbol{\psi}$ divide into two different sets each for the specific probability and we distinguish them with the superscripts: (c) for the parameters describing the probability of the next event sentiment, (sr) for those parameters describing the probability of the next dyad.

With the formula (5.5), the product of conditional probabilities in (5.4) becomes

$$\begin{aligned} & \prod_{m=1}^M p(s_{e_m}, r_{e_m}, c_{e_m}, t_{e_m} | E_{t_{m-1}}; \boldsymbol{\beta}, \boldsymbol{\psi}, \mathbf{X}) = \\ & = \prod_{m=1}^M \underbrace{p(c_{e_m} | s_{e_m}, r_{e_m}, t_{e_m}, E_{t_{m-1}}; \boldsymbol{\beta}^{(c)}, \boldsymbol{\psi}^{(c)}, \mathbf{X})}_{\text{Sentiment}} \underbrace{p(s_{e_m}, r_{e_m}, t_{e_m} | E_{t_{m-1}}; \boldsymbol{\beta}^{(sr)}, \boldsymbol{\psi}^{(sr)}, \mathbf{X})}_{\text{REM}} \end{aligned} \quad (5.6)$$

where

- $p(c_{e_m} | s_{e_m}, r_{e_m}, t_{e_m}, E_{t_{m-1}}; \boldsymbol{\beta}^{(c)}, \boldsymbol{\psi}^{(c)}, \mathbf{X})$ is the probability of the sentiment of events. Following the initial assumption of the sentiment measured on a categorical scale with two categories, positive and negative, we can model the first element of the product (sentiment model) in (5.6) as a Probit model. In the next Section, we discuss the Probit model as sentiment model in more detail. However, the sentiment model can be specified as any model for the response variable $c \in C$ (with C being the set of possible sentiments), that can be measured on any scale, ranging from a categorical to a continuous scale, for instance: the ordered Logit model in the case of a sentiment measured in an ordinal categorical scale with more than two event types, the Normal model in the case of a continuous scale, the Poisson model in the case of counts and so forth (Agresti, 2015).
- $p(s_{e_m}, r_{e_m}, t_{e_m} | E_{t_{m-1}}; \boldsymbol{\beta}^{(sr)}, \boldsymbol{\psi}^{(sr)}, \mathbf{X})$ can be modeled as a REM (Butts, 2008) where the log-event rate for the event e' is modeled as in (5.2) but this time

only defined on dyad $(s_{e'}, r_{e'})$,

$$\ln \lambda(s_{e'}, r_{e'}, X_{e'}, E_{t_{m-1}}, \beta^{(sr)}, \psi^{(sr)}) = \mathbf{u}_{e', t_m}(\psi^{(sr)}, E_{t_{m-1}}, \mathbf{X})\beta^{(sr)} \quad (5.7)$$

The dimension of the risk set for the REM becomes $N(N - 1)$, meaning that the set considers only the dyadic events defined by any possible combination of sender and receiver, excluding the sentiment. However, the endogenous statistics in (5.7) can still be based on the sentiment of past events.

Therefore, we can write the likelihood function of the SentiREM (Sentiment Relational Event Model) as

$$\mathcal{L}_{\text{SentiREM}}(\beta, \psi; E_{t_M}, \mathbf{X}) = \mathcal{L}_{\text{Sentiment}}(\beta^{(c)}, \psi^{(c)}; E_{t_M}, \mathbf{X}) \cdot \mathcal{L}_{\text{REM}}(\beta^{(sr)}, \psi^{(sr)}; E_{t_M}, \mathbf{X}) \quad (5.8)$$

where $\mathcal{L}_{\text{Sentiment}}(\beta^{(c)}, \psi^{(c)}; E_{t_M}, \mathbf{X})$ is the likelihood of the sentiment model, in this case of a Probit model and, $\mathcal{L}_{\text{REM}}(\beta^{(sr)}, \psi^{(sr)}; E_{t_M}, \mathbf{X})$ refers to the likelihood of the REM model. Our interest focuses on finding the set of model parameters (β, ψ) that optimizes the likelihood of the SentiREM in (5.8).

The sentiment model: modeling the next event sentiment via Probit regression

Given that the possible sentiments are two (p, n) and that the probability that we want to model is the one of observing $c_{e_m} = n$ (negative sentiment), we can write

$$p(c_{e_m} = n | s_{e_m}, r_{e_m}, t_{e_m}, E_{t_{m-1}}; \beta^{(c)}, \psi^{(c)}, \mathbf{X}) = \Phi(\eta_m(\beta^{(c)}, \psi^{(c)}, E_{t_{m-1}}, \mathbf{X})) \quad (5.9)$$

where, $\Phi(\cdot)$ is the cumulative distribution function of the Normal distribution and $\eta_m(\cdot)$ describes the linear predictor at time t_m for the observed event e_m .

Then, the likelihood of the sentiment model $\mathcal{L}_{\text{Sentiment}}(\beta^{(c)}, \psi^{(c)}; E_{t_M}, \mathbf{X})$ modeled via Probit regression becomes

$$\mathcal{L}_{\text{Sentiment}}(\beta^{(c)}, \psi^{(c)}; E_{t_M}, \mathbf{X}) = \prod_{m=1}^M \left[\Phi(\eta_m(\beta^{(c)}, \psi^{(c)}, E_{t_{m-1}}, \mathbf{X}))^{\delta_m} \left(1 - \Phi(\eta_m(\beta^{(c)}, \psi^{(c)}, E_{t_{m-1}}, \mathbf{X})) \right)^{1-\delta_m} \right] \quad (5.10)$$

with

$$\delta_m = \begin{cases} 1 & \text{if } c_{e_m} = n \\ 0 & \text{if } c_{e_m} = p \end{cases} \quad (5.11)$$

The linear predictor at time t_m is a function of endogenous and exogenous statistics and can be written as

$$\eta_m(\boldsymbol{\beta}^{(c)}, \boldsymbol{\psi}^{(c)}, E_{t_{m-1}}, \mathbf{X}) = \mathbf{z}_m(\boldsymbol{\psi}^{(c)}, E_{t_{m-1}}, \mathbf{X})\boldsymbol{\beta}^{(c)}$$

with \mathbf{z}_m being the row-vector of the statistics for the dyad (s_{e_m}, r_{e_m}) observed at t_m , which are either based on exogenous information (\mathbf{X}) or on the history of previous events, $E_{t_{m-1}}$, and can depend on the memory parameters $(\psi_p^{(c)}, \psi_n^{(c)})$. Parameters $\boldsymbol{\beta}^{(c)}$ describe the log-linear effect of statistics on the probability that the next event type is negative (n): positive (negative) values of such parameters indicate a higher probability of the next event to be negative (positive). Hence, the Probit model can be described by two data structures: (i) a matrix of statistics, \mathbf{Z} , with as many rows as the number of events (M) and as many columns as the number of statistics in the linear predictor, (ii) a vector c^{obs} of length M with the time-ordered sequence of observed event types, which is the response variable of the Probit model.

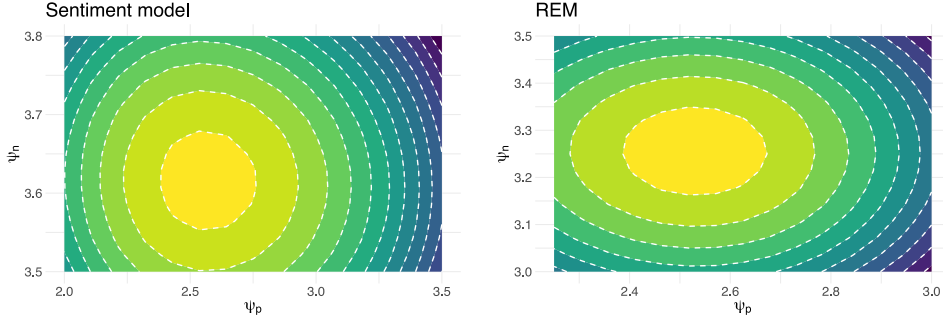
5.2.1 The estimation of the model parameters

We are now interested in finding the set of parameters $(\boldsymbol{\beta}, \boldsymbol{\psi})$ that maximizes the likelihood function of the SentiREM presented in (5.8), which is the same set that minimizes the negative log-likelihood. Given the separability on a parameter-level of the two models, we can carry out two separate optimizations to find the best set of parameters: one optimization of the sentiment model as to the parameters $(\boldsymbol{\beta}^{(c)}, \boldsymbol{\psi}^{(c)})$ and one optimization of the REM as to the parameters $(\boldsymbol{\beta}^{(sr)}, \boldsymbol{\psi}^{(sr)})$. Indeed, considering the separation of the two likelihoods, the minimization of the negative log-likelihood can be written as

$$\arg \min_{\boldsymbol{\beta}, \boldsymbol{\psi}} \{-\ln \mathcal{L}_{\text{SentiREM}}(\boldsymbol{\beta}, \boldsymbol{\psi}; E_{t_M}, \mathbf{X})\} = \left\{ \arg \min_{\boldsymbol{\beta}^{(c)}, \boldsymbol{\psi}^{(c)}} \{-\ln \mathcal{L}_{\text{Sentiment}}(\boldsymbol{\beta}^{(c)}, \boldsymbol{\psi}^{(c)}; E_{t_M}, \mathbf{X})\}, \right. \\ \left. \arg \min_{\boldsymbol{\beta}^{(sr)}, \boldsymbol{\psi}^{(sr)}} \{-\ln \mathcal{L}_{\text{REM}}(\boldsymbol{\beta}^{(sr)}, \boldsymbol{\psi}^{(sr)}; E_{t_M}, \mathbf{X})\} \right\} \quad (5.12)$$

The maximum likelihood estimates of each set of parameters can be found

with two different approaches: via the optimization of the bivariate negative profile log-likelihood or via the trust region optimization algorithm. Below we describe each of these approaches in detail.



(a) Sentiment model negative profile log-likelihood (b) REM negative profile loglikelihood

Figure 5.2: Example of negative profile log-likelihood for each model in the SentiREM

The bivariate negative profile log-likelihood

We define the bivariate negative profile log-likelihood for both sentiment and REM model as a two-dimensional function given the presence of two memory parameters,

$$-\ln \mathcal{L}_{\text{profile Sentiment}} \left(\psi_p^{(c)}, \psi_n^{(c)} \right) = \min_{\beta^{(c)}} \left\{ -\ln \mathcal{L}_{\text{Sentiment}} \left(\beta^{(c)}, \psi_p^{(c)}, \psi_n^{(c)}; E_{t_M}, \mathbf{X} \right) \right\}$$

with $(\psi_p^{(c)}, \psi_n^{(c)}) \in \mathbb{R}^2$

$$-\ln \mathcal{L}_{\text{profile REM}} \left(\psi_p^{(sr)}, \psi_n^{(sr)} \right) = \min_{\beta^{(sr)}} \left\{ -\ln \mathcal{L}_{\text{REM}} \left(\beta^{(sr)}, \psi_p^{(sr)}, \psi_n^{(sr)}; E_{t_M}, \mathbf{X} \right) \right\}$$

with $(\psi_p^{(sr)}, \psi_n^{(sr)}) \in \mathbb{R}^2$

(5.13)

The value of each negative profile log-likelihood is obtained as the minimum value of the negative log-likelihood where the memory parameters are fixed and the optimization is carried over the vector of effects of the statistics. This translates to two different model optimizations: if we fix the memory parameters in

the sentiment likelihood modeled via Probit and carry the optimization over the $\beta^{(c)}$ we optimize the negative log-likelihood of a Probit regression (which can be done using existing software like the `stats::glm()` function in R). Whereas, if we fix the memory parameters in the REM likelihood and carry the optimization over the $\beta^{(sr)}$ we are optimizing the negative log-likelihood of a REM and this can be also carried out by using existing software like the R packages `relevent` (Butts, 2023), and, `survival` (Therneau, 2022). Therefore, if we define a two-dimensional grid of values in \mathbb{R}^2 and for each pair of values we compute the two optimizations in (5.13) we will be able to represent a mesh surface plot of the negative profile log-likelihood for each model and an example of it is shown in Figure (5.2). Plotting the profile log-likelihood is still feasible on two dimensions and it helps us to understand the possible presence of local minima. If there exists a minimum for each negative profile log-likelihood, that value will correspond to the global minimum for both the optimized log-memory parameters ψ and the optimized vector of effects β . The sets of parameters of the sentiment and the REM model will then be a solution for the optimization of the complete negative log-likelihood $-\ln \mathcal{L}_{\text{SentiREM}}(\beta, \psi; E_{t_M}, \mathbf{X})$.

The trust region optimization

Optimization via trust region algorithm is widely used in non-constrained problems. The general idea of the trust region algorithm is to optimize an objective function starting by the definition of a region of trust per each parameter which will expand or shrink at each iteration until the convergence. The quantities of interest in a trust region optimization are the values that the objective function, its gradient, and its hessian assume at specific values of the model parameters. Such quantities are updated at each iteration and regulate the relative expansion (or contraction) of the region, resulting in the change of the value of the parameters for the next iteration. The minimum and maximum radius of the region can be defined by the researcher as well as other parameters like the maximum number of iterations. We omit here the explanation of the optimization and we refer to Fletcher (1987) and Nocedal and Wright (2006) for a more detailed explanation of the algorithm used in the R package `trust` (Geyer, 2015).

The trust region optimization requires the objective function to be differentiable at least twice with respect to all the model parameters. The negative log-likelihood of both the sentiment model and the REM is twice differentiable

with respect to the vector of effects of statistics $\beta^{(c)}$ and $\beta^{(sr)}$. However, this is not necessarily possible for the memory parameters because the memory decay can be specified by the researcher and can assume several parametric functions some of which are not twice differentiable (e.g., the one-step memory model proposed in Arena et al. (2023)). In order to avoid complications with the derivatives on the memory parameters, we maintain the assumption of exponential weight decay for both event types, which is a twice differentiable function. In the case where the chosen function for the weight decay is not twice differentiable, the preferred approach is to optimize the negative profile log-likelihood. The calculation of gradient and hessian can become complex depending on the set of statistics included in the linear predictor of sentiment and REM model.

5.2.2 Testing of model parameters

In this section, we present various hypothesis tests that can be executed under the proposed statistical framework; please note that many more hypotheses can be formulated in addition. In order to be able to test whether events with different sentiments differ in how long they impact future social interaction, we first propose a test on the memory parameters of the two sentiments. Subsequently, in order to test whether events of different sentiments have a different substantive impact on social interaction dynamics, we propose testing the effects of the endogenous statistics corresponding to different sentiments.

First, we test of whether events of different sentiments differ in how long they affect the interaction rate between actors:

$$\begin{cases} H_0 : \psi_p^{(sr)} = \psi_n^{(sr)} \\ H_1 : \psi_p^{(sr)} \neq \psi_n^{(sr)} \end{cases} \quad (5.14)$$

These hypotheses can be tested using the Bayes factor, which quantifies the relative evidence of one model (hypothesis) compared to another. Therefore, following the hypothesis in (5.14), we can define two models: (i) model \mathcal{M}_0 where, in the estimation stage, we constrain the two memory parameters to be equal (i.e. only one memory parameter is assumed in the model, and it is the same for any sentiment-related weight decay) and (ii) model \mathcal{M}_1 where we estimate the model parameters without any constraint on the memory parameters. The general formula of the Bayesian information criterion for any model

\mathcal{M} (BIC; Schwarz (1978)) is given by

$$BIC(\mathcal{M}) = k \log(M) - 2 \ln \mathcal{L}, \quad (5.15)$$

where k is the number of parameters in the model of interest, M is the number of events in the event sequence and, $-\ln \mathcal{L}$ is the optimized negative log-likelihood of the model of interest. Since the hypotheses in (5.14) refer only to the memory parameters in the model for the dyad (REM), the negative log-likelihood used in the computation of the BIC will be $-\ln \mathcal{L}_{\text{REM}}(\hat{\beta}^{(sr)}, \hat{\psi}^{(sr)}; E_{t_M}, \mathbf{X})$. The Bayes factor between the two models is given by

$$BF(\mathcal{M}_0, \mathcal{M}_1) = \exp\{BIC(\mathcal{M}_1)/2 - BIC(\mathcal{M}_0)/2\}, \quad (5.16)$$

which quantifies the relative evidence in the data in favor of \mathcal{M}_0 against \mathcal{M}_1 . Hypothesis H_1 in (5.14) might be further divided into two one-sided hypotheses, a left one-sided hypothesis $\psi_p^{(sr)} < \psi_n^{(sr)}$ and a right one-sided hypothesis $\psi_p^{(sr)} > \psi_n^{(sr)}$. Further, Bayes Factors can also be calculated to quantify the relative evidence of the three hypotheses against each other. Alternatively, the test in (5.14) could be executed using a classical Wald test, but that requires either a one-sided (with prespecified direction) or a two-sided test, and this test would not be statistically consistent (as there would be a strictly positive probability (typically .05) to incorrectly reject a correct null hypothesis). Finally, note that similar hypotheses could be formulated for the memory parameters under the sentiment model, i.e., $\psi_p^{(c)}$ and $\psi_n^{(c)}$.

We now consider the hypothesis tests on the strength of the effect of past events with different sentiments via the effects of the corresponding endogenous statistics. Suppose we want to test the strengths of positive and negative inertia; we then consider the following hypothesis test:

$$\begin{cases} H_0 : \beta_{\text{Inertia}_p} = \beta_{\text{Inertia}_n} \\ H_1 : \beta_{\text{Inertia}_p} < \beta_{\text{Inertia}_n} \\ H_2 : \beta_{\text{Inertia}_p} > \beta_{\text{Inertia}_n} \end{cases} \quad (5.17)$$

In (5.17), a Bayes Factor can be calculated for testing the relative evidence of the hypotheses, one against another one. We may instead want to test whether the effects of endogenous statistics are the same regardless the event type they refer to, and for this reason consider the hypothesis H_1 and H_2 as one single

hypothesis that is $\beta_{\text{Inertia}_p} \neq \beta_{\text{Inertia}_n}$ and calculate the Bayes Factor of such model against H_0 .

The Bayes Factor can also be used when testing hypotheses on more complex network dynamics, such as with second-order endogenous statistics. Such statistics are based on the time-ordered sequence of two dyads, with the third dyad closing the triangular pattern. If two or more sentiments are observed, they can generate different endogenous statistics of the same triadic pattern where the sequence of dyadic events is described by a different sequence of sentiments. For instance, in the case of two sentiments, positive and negative, the transitivity closure (tc) can assume four definitions: (i) when the two dyadic events are positive, (ii) when both the dyadic events are negative, (iii) when the first event is positive and the second is negative, (iv) when the first event is negative and the second is positive. A hypothesis test on the transitivity can be then written as,

$$\begin{cases} H_0 : \beta_{\text{tc}_{pp}} = \beta_{\text{tc}_{nn}} = \beta_{\text{tc}_{np}} = \beta_{\text{tc}_{pn}} \\ H_1 : \text{at least one } \beta_{\text{tc}_i} \neq \beta_{\text{tc}_j} \quad i, j \in \{pp, nn, np, pn\} \quad \text{with } i \neq j \end{cases} \quad (5.18)$$

Based on what the researcher aims to examine, both hypotheses H_0 and H_1 can be further divided into more specific hypotheses to test (e.g., one-sided hypotheses as discussed for Inertia).

Other hypotheses can be defined on higher order network dynamics or on non-nested models. In the former case, the network dynamics are described by three or more dyadic events and the testing can follow the examples in 5.17 and in 5.18. In the latter case, we want to compare two models whose linear predictors are different and neither of the two models can be obtained by imposing parametric restrictions on the other. Finally, the Bayes Factor remains a valid tool to test these hypotheses and provide a reliable answer to them.

5.3 Numerical performance of the SentiREM

In this section, we perform a simulation study to analyze the behavior of the maximum likelihood estimates of the SentiREM in simulated networks, varying the sequence length and the distribution of the sentiments across the events. We generate 600 event sequences of 2000 events and involving 10 actors. We

specify the same set of exogenous and endogenous statistics in the linear predictors of both models but with different set of effects β and memory parameters ψ . The specification of the linear predictor for both models can be written as,

$$\begin{aligned}
& \beta_0 + \beta_{\text{Dyadic1}} \text{Dyadic1}(s_{e_m}, r_{e_m}) + \beta_{\text{Dyadic2}} \text{Dyadic2}(s_{e_m}, r_{e_m}) + \\
& \quad \beta_{\text{Inertia}_p} \text{Inertia}_p(s_{e_m}, r_{e_m}, c_{e_m} = p, E_{t_{m-1}}, \psi_p) + \\
& \quad \beta_{\text{Reciprocity}_p} \text{Reciprocity}_p(s_{e_m}, r_{e_m}, c_{e_m} = p, E_{t_{m-1}}, \psi_p) + \\
& \quad \beta_{\text{ids}_p} \text{ids}_p(s_{e_m}, r_{e_m}, c_{e_m} = p, E_{t_{m-1}}, \psi_p) + \quad (5.19) \\
& \quad \beta_{\text{Inertia}_n} \text{Inertia}_n(s_{e_m}, r_{e_m}, c_{e_m} = n, E_{t_{m-1}}, \psi_n) + \\
& \quad \beta_{\text{Reciprocity}_n} \text{Reciprocity}_n(s_{e_m}, r_{e_m}, c_{e_m} = n, E_{t_{m-1}}, \psi_n) + \\
& \quad \beta_{\text{ids}_n} \text{ids}_n(s_{e_m}, r_{e_m}, c_{e_m} = n, E_{t_{m-1}}, \psi_n)
\end{aligned}$$

where β_0 is the intercept, the statistics Dyadic1 and Dyadic2 are two dyad-level exogenous statistics that are constant over time. The endogenous statistics inertia, reciprocity, and in-degree of the sender (ids) are computed per sentiment and weighted by an exponential memory decay that allows for a different memory parameter per event sentiment. The model parameters (β, ψ) of the SentiREM will be described by the two sets ($\beta^{(sr)}, \psi^{(sr)}$) for the REM and ($\beta^{(c)}, \psi^{(c)}$) for the sentiment model. The vector of effects and memory parameters used in the linear predictor of the REM is (β, ψ)^(sr) = ($\beta_0, \beta_{\text{Dyadic1}}, \beta_{\text{Dyadic2}}, \beta_{\text{Inertia}_p}, \beta_{\text{Reciprocity}_p}, \beta_{\text{ids}_p}, \beta_{\text{Inertia}_n}, \beta_{\text{Reciprocity}_n}, \beta_{\text{ids}_n}, \psi_p, \psi_n$)^(sr) = (-7, 0.1, -0.2, 7, 16, 4, 10, -17, -4, ln(12), ln(30)). The vector of effects for the sentiment model (Probit) is (β, ψ)^(c) = ($\beta_0, \beta_{\text{Dyadic1}}, \beta_{\text{Dyadic2}}, \beta_{\text{Inertia}_p}, \beta_{\text{Reciprocity}_p}, \beta_{\text{ids}_p}, \beta_{\text{Inertia}_n}, \beta_{\text{Reciprocity}_n}, \beta_{\text{ids}_n}, \psi_p, \psi_n$)^(c) = ($\beta_0^{(c)}, -0.3, 1.5, -5, -18, -3, 5, 15, 7, \ln(6), \ln(20)$), where we set the intercept of the sentiment model to different values $\beta_0^{(c)}$ = (-3, -2.5, -2, -1.5, -1, -0.5). We generate 100 relational event sequences for each value, resulting in networks with different proportions of the two event types: the proportions of positive (p) range between the 20% and 80%.

For each simulated network, we estimate a SentiREM on the sub-networks that are defined on the first 250, 500, 1000 events and on the whole network of 2000 events. The proportions of the two event types vary across the sub-networks. The optimization of the memory parameters is carried over their natural logarithm, thus we refer to their transformed true values ($\psi_p^{(sr)}, \psi_n^{(sr)}$) = (ln(12), ln(30)) \approx (2.48, 3.40) for the relational event model and ($\psi_p^{(c)}, \psi_n^{(c)}$) =

A modeling framework for sentiment-driven relational events

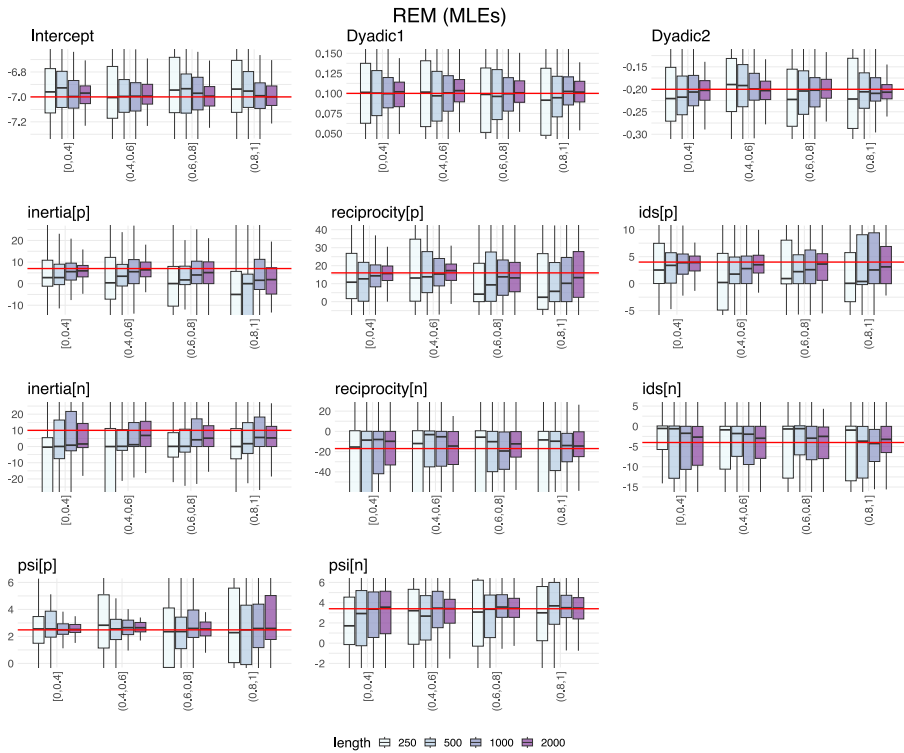


Figure 5.3: (Simulation study: modeling the next dyadic event) The distribution of the maximum likelihood estimates of the REM across different proportions of negative events (on the x-axis). True values of each model parameter are marked by the horizontal bold line.

$(\ln(6), \ln(20)) \approx (1.79, 2.99)$ for the sentiment model. For each model, we carry out the estimation of the whole set of model parameters via trust region optimization. Figure 5.3 and 5.4 show the trend of the estimates for each model parameter at different lengths of the event sequence (250, 500, 1000 and, 2000 events) and over different proportions of each the event types. For clarity of exposition, we divide the proportion of the negative sentiment along four classes on the x-axes.

The results coming from the REM show a general improvement of the estimation of the model parameters as the length of the event sequence increases and the proportion of events becomes less extreme. When modeling the next event sentiment with the Probit regression we observe the same behavior of the maximum likelihood estimates as in the REM estimates and we also notice a higher volatility of the estimates when the sequence becomes shorter. The

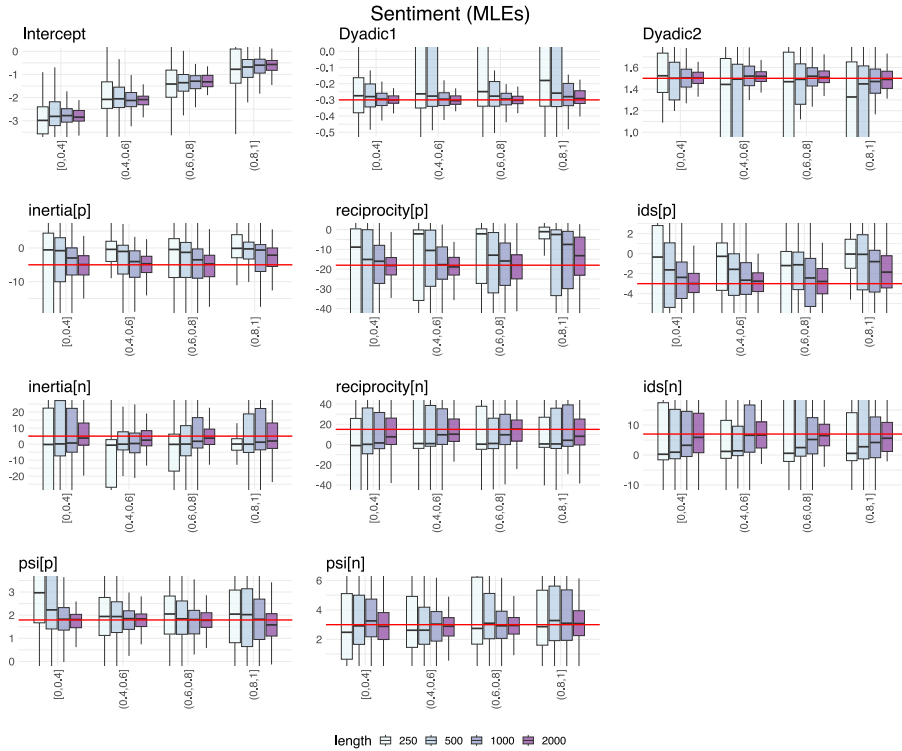


Figure 5.4: (Simulation study: modeling the next sentiment) The distribution of the maximum likelihood estimates of the sentiment model (Probit) across different proportions of negative events (on the x-axis). True values of each model parameter are marked by the horizontal bold line.

trend of the intercept in the sentiment model is expected since we estimated networks with six different intercepts ranging from -3 to -0.5 by a step of 0.5 and, also in this case we observe the increasing variability of the estimates as the network of events shortens. Overall, we can conclude that the estimation procedure works as intended.

5.4 Case study: modeling memory decay of trades and attacks between players in an online strategy game

We applied the SentiREM to a sequence of interactions observed in an online real-time strategy game (Hajibagheri et al., 2018). In terms of predictive per-

formance, the proposed model outperforms other existing approaches that set the memory parameters to fixed predefined constants. In this section, we present the empirical data and formulate the research questions, we discuss model specification and results of the SentiREM and, finally, we compare the predictive performance of the SentiREM to several fixed memory models and to a model without memory decay.

5.4.1 Data and research questions

In the strategy game under study, each player is the leader of her own village and focuses on developing and expanding her territory by building constructions or upgrading existing ones, developing resource fields, conquering side villages, and recruiting military troops with an offensive or defensive aim. Furthermore, each player can trade resources with other players via the marketplace, can attack other player's side villages (aiming to loot their resources or take over their lands) and can join or abandon alliances with other players.

The available network data consist of a time-ordered sequence of attacks and trades that are collected over the period of a month. In addition to the sequence of events, we also know when alliances formed by players. We focus on the relational event sequence of 7535 attacks (17%) or trades (83%), occurring among 269 actors of four alliances. The time variable is available as a timestamp ("yyyy-dd-mm hh:mm:ss") in the original data and we re-scaled the waiting time between events to hours. Therefore, in the estimation stage the natural logarithm of the memory parameters corresponds to the natural logarithm of hours. We are interested in (i) testing whether the memory decay of past attack events differs from the one of past trade events, (ii) testing whether past attacks have a stronger impact on social interaction dynamics than past trades, and (iii) understanding how accurate the model is in predicting the time, the dyad, and the sentiment of events.



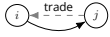

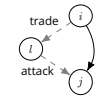
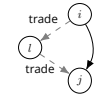
Statistic	Formula	Description
SameCommunity (i, j, t_m)	$\mathbb{I}(\text{community}(i, t_m) = \text{community}(j, t_m))$	SameCommunity is a time-varying statistic that assumes value 1 if sender and receiver of the interaction belong to the same community at the time of the interaction, 0 otherwise. The value assumed is the same for both directed dyads ($s_e = i, r_e = j$) and ($s_e = j, r_e = i$).
Inertia _{trade} $(i, j, t_m, \psi_{\text{trade}})$ 	$\sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(i, j, \text{trade}) w(t_{m-1} - t_e, \psi_{\text{trade}})$	Inertia for the trades quantifies the weighted volume of already occurred events in which the sender i initiated a trade with the receiver j .
Inertia _{attack} $(i, j, t_m, \psi_{\text{attack}})$ 	$\sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(i, j, \text{attack}) w(t_{m-1} - t_e, \psi_{\text{attack}})$	Inertia for the attacks quantifies the weighted volume of past events in which the sender i attacked receiver j 's side village.
Reciprocity _{trade} $(i, j, t_m, \psi_{\text{trade}})$ 	$\sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(j, i, \text{trade}) w(t_{m-1} - t_e, \psi_{\text{trade}})$	Reciprocity for the trades quantifies the weighted volume of past trades that i received from j .
Reciprocity _{attack} $(i, j, t_m, \psi_{\text{attack}})$ 	$\sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(j, i, \text{attack}) w(t_{m-1} - t_e, \psi_{\text{attack}})$	Reciprocity for the attacks quantifies the weighted volume of past events in which i was attacked by j .
EoF $(i, j, t_m, \psi_{\text{attack}})$ 	$\sum_{l \in \mathcal{S} \setminus \{i, j\}} \sum_{e \in E_{t_{m-1}}} \sum_{\substack{e^* \in E_{t_{m-1}}: \\ t_{e^*} \in [t_e - (t_{m-1} - t_e), t_e]}} \mathbb{I}_{e^*}(i, l, \text{trade}) \mathbb{I}_e(l, j, \text{attack}) w(t_{m-1} - t_e, \psi_{\text{attack}})$	Enemy-of-Friend (EoF) calculates the weighted volume of past trades generated from i to l and that happened before any attack initiated by l towards j . The weight is the one of the past attack (l, j) . In this triadic structure i and l are potential allies and j is a possible adversary (at least for l).
FoF $(i, j, t_m, \psi_{\text{trade}})$ 	$\sum_{l \in \mathcal{S} \setminus \{i, j\}} \sum_{e \in E_{t_{m-1}}} \sum_{\substack{e^* \in E_{t_{m-1}}: \\ t_{e^*} \in [t_e - (t_{m-1} - t_e), t_e]}} \mathbb{I}_{e^*}(i, l, \text{trade}) \mathbb{I}_e(l, j, \text{trade}) w(t_{m-1} - t_e, \psi_{\text{trade}})$	Friend-of-Friend (FoF) quantifies the number of past trades generated from i to l and that happened before any trade from l to j . The sum is weighted with the weight of the trade (l, j) . In this triadic structure i, l and j are all potential allies.

Table 5.1: Graph, formula, and description of each statistic specified in both linear predictors of the SentiREM. In the graphs, the dashed gray arrows represent past events (trades or attacks), the bold black arrows represent the event which the statistic is computed for. The indicator variables are written in a compact form to be read as, for instance, $\mathbb{I}_e(i, j, \text{trade}) = \mathbb{I}_e(s_e = i, r_e = j, c_e = \text{trade})$. Each event contributing to any of the statistics is weighted via the exponential decay function $w_{\text{trade}}(t_{m-1} - t_e, \psi_{\text{trade}})$ for the trades and $w_{\text{attack}}(t_{m-1} - t_e, \psi_{\text{attack}})$ for the attacks.

5.4.2 Model specification and results

We assume that the shape of the memory decay in the SentiREM follows a parametric exponential decay and per event type can be written as,

$$w_{\text{trade}}(t_{m-1} - t_e, \psi_{\text{trade}}) = \frac{\ln(2)}{\exp(\psi_{\text{trade}})} \exp\left\{- (t_{m-1} - t_e) \frac{\ln(2)}{\exp(\psi_{\text{trade}})}\right\}$$

$$w_{\text{attack}}(t_{m-1} - t_e, \psi_{\text{attack}}) = \frac{\ln(2)}{\exp(\psi_{\text{attack}})} \exp\left\{- (t_{m-1} - t_e) \frac{\ln(2)}{\exp(\psi_{\text{attack}})}\right\} \quad (5.20)$$

where $t_{m-1} - t_e$ is the time elapsed at the time point t_{m-1} since the time of occurrence t_e of the event e , the parameters ψ_{trade} and ψ_{attack} are the natural logarithm of the memory parameters for each event type; therefore, $\exp(\psi_{\text{trade}})$ and $\exp(\psi_{\text{attack}})$ are the half-life parameters for the trades and the attacks and explain after how many hours a past trade and a past attack will halve their weight in the computation of the endogenous statistics. We assume that the risk set of the REM is characterized only by the observed 1469 dyads, reducing the computational burden for the update of the endogenous statistics at different values of the memory parameter. We define a set of statistics that might explain the dynamics of trades and attacks: intercept, SameCommunity (1 is both actors at the time of the interaction belong to the same community and 0 otherwise), Inertia and Reciprocity for both sentiments "attack" and "trade", Enemy-of-Friend and Friend-of-Friend (describing triadic patterns in the fashion of the transitivity closure where the first two events events in the triad assume specific event types (Brandes et al., 2009)). For both Enemy-of-Friend and Friend-of-Friend we take the temporal order of the two events in the triad into account as defined for transitivity closure in Arena, Mulder, and Leenders (2022).

The graph of the dynamic pattern, the formula, and a further description of the statistics are reported in Table 5.1. In light of the chosen set of variables, in

Parameter (β, ψ)	REM (dyadic event rate)			Sentiment model (probability of attack)		
	Estimate (Std. Error)	Pr(> z)	Pr(=0)	Estimate (Std. Error)	Pr(> z)	Pr(= 0)
Intercept	-13.715 (0.023)	< 2.2e-16	< 2.2e-16	-0.229 (0.080)	0.004	0.591
SameCommunity	0.607 (0.028)	< 2.2e-16	< 2.2e-16	-1.378 (0.094)	< 2e-16	< 2.2e-16
Inertia (trade)	2.030 (0.025)	< 2.2e-16	< 2.2e-16	-10.170 (1.252)	4.44e-16	4.08e-13
Reciprocity (trade)	1.440 (0.037)	< 2.2e-16	< 2.2e-16	-16.360 (2.649)	6.55e-10	4.53e-07
Inertia (attack)	7.184 (0.133)	< 2.2e-16	< 2.2e-16	24.176 (3.689)	5.65e-11	4.1e-08
Reciprocity (attack)	18.941 (3.759)	4.68e-07	2.66e-04	1.445 (1.380)	0.295	0.980
Enemy-of-Friend	36.816 (1.246)	< 2.2e-16	< 2.2e-16	4.492 (3.858)	0.244	0.978
Friend-of-Friend	2.438 (0.077)	< 2.2e-16	< 2.2e-16	-124.377 (79.569)	0.118	0.962
$\hat{\psi}_{\text{trade}}$	1.316 (0.023)	< 2.2e-16	< 2.2e-16	1.859 (0.125)	< 2e-16	< 2.2e-16
$\hat{\psi}_{\text{attack}}$	3.779 (0.045)	< 2.2e-16	< 2.2e-16	1.724 (0.184)	< 2e-16	< 2.2e-16

Table 5.2: (Case study) Maximum likelihood estimates and standard errors (between brackets) for the parameters of both the relational event model (REM) and the sentiment model (Probit model). The columns named "Pr(>|z|)" and "Pr(=0)" report, respectively, p-value and posterior probability of each parameter being zero.

both models the linear predictor for a generic event e can be written as follows,

$$\begin{aligned}
 & \beta_0 + \beta_{\text{SameCommunity}} \text{SameCommunity}(s_e, r_e) + \\
 & \beta_{\text{Inertia}_{\text{trade}}} \text{Inertia}_{\text{trade}}(s_e, r_e, c_e = \text{trade}, E_{t_{m-1}}, \psi_{\text{trade}}) + \\
 & \beta_{\text{Reciprocity}_{\text{trade}}} \text{Reciprocity}_{\text{trade}}(s_e, r_e, c_e = \text{trade}, E_{t_{m-1}}, \psi_{\text{trade}}) + \\
 & \beta_{\text{Inertia}_{\text{attack}}} \text{Inertia}_{\text{attack}}(s_e, r_e, c_e = \text{attack}, E_{t_{m-1}}, \psi_{\text{attack}}) + \quad (5.21) \\
 & \beta_{\text{Reciprocity}_{\text{attack}}} \text{Reciprocity}_{\text{attack}}(s_e, r_e, c_e = \text{attack}, E_{t_{m-1}}, \psi_{\text{attack}}) + \\
 & \beta_{\text{EoF}} \text{EoF}(s_e, r_e, E_{t_{m-1}}, \psi_{\text{attack}}) + \\
 & \beta_{\text{FoF}} \text{FoF}(s_e, r_e, E_{t_{m-1}}, \psi_{\text{trade}})
 \end{aligned}$$

where we define a set of parameters (β, ψ) for both the sentiment model and the REM, with ψ being the vector of the natural logarithm of the memory pa-

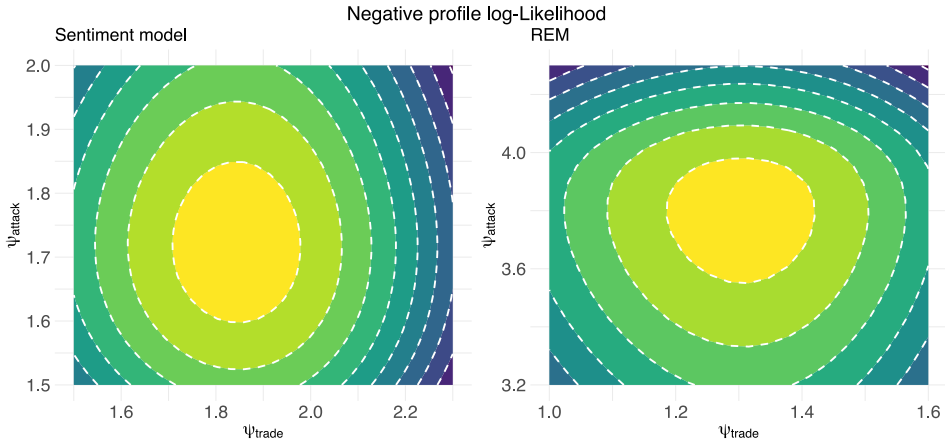


Figure 5.5: (Case study) Negative profile log-Likelihood for REM and sentiment model (Probit) over a grid of memory parameters.

rameters. We model the probability of the next event type to be an attack in the sentiment model and the hazard rate for the next dyad in the relational event model.

5

For both models, we optimize the negative log-Likelihoods and the contour plots of the two negative profile log-Likelihoods are shown in Figure 5.5.

The maximum likelihood estimates with their standard errors for the SentireM under the assumption that memory parameters are different across event types are reported in Table 5.2. In order to test whether past attack events have a different weight decay than past trade events we compare the model from Table 5.2 with a model where memory parameters are constrained to be the same across the event types. We describe the two hypotheses as

$$\begin{cases} H_0 : \psi_{\text{trade}} = \psi_{\text{attack}} \\ H_1 : \psi_{\text{trade}} \neq \psi_{\text{attack}} \end{cases} \quad (5.22)$$

and, to simplify the notation, we refer to the model in H_0 as M_0 and to the model in H_1 as M_1 . The two negative profile log-Likelihoods for M_0 are shown in Figure 5.6. In Table 5.3 we show the BIC of the REM and the sentiment model under each of the two conditions on the memory parameters are reported along with the Bayes Factor on its logarithmic scale (Kass & Raftery, 1995). From Figure 5.6 we can confirm that for both REM and the sentiment model there exists a model that is the best under H_0 (i.e., both functions show to have a minimum).

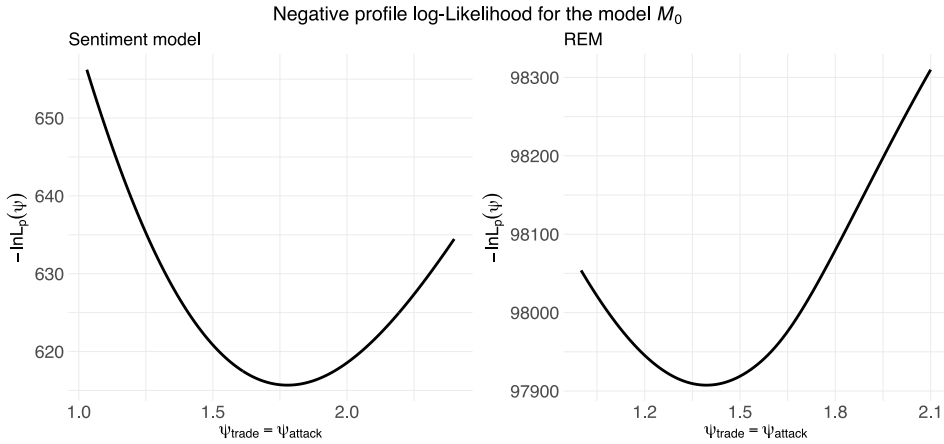


Figure 5.6: (Case study) Negative profile log-Likelihood for the model M_0 , where for both REM and sentiment model (Probit) the memory parameters are assumed to be the same across event types, that is $\psi_{trade} = \psi_{attack}$.

	M_0	M_1	$\log_{10} BF_{M_1 \text{ vs. } M_0}$
	BIC	BIC	
REM (dyadic event rate)	195904.30	195498.60	88.1
Sentiment model (probability of attack)	1320.65	1320.13	0.11

Table 5.3: (Case study) BIC and Bayes Factor (on the logarithmic scale of base 10) between the two models, M_0 and M_1 .

The Bayes Factor in Table 5.3 communicates two different results. Indeed, given that the Bayes Factor quantifies the evidence that is in favor of M_1 relative to M_0 , we find such evidence to be much stronger in the relational event model rather than in the sentiment model. In the REM, we find that the weight of attacks generally lasts much longer ($\exp(\psi_{attack}) \approx 44\text{hrs}$, almost 2 days) than the weight of trades ($\exp(\psi_{trade}) \approx 4\text{hrs}$), resulting in a much slower weight decay for the attacks than for the trades. In the sentiment model under M_1 , both memory parameters are already somewhat close to each other (between 5 and 6 hours) and the evidence in favor of the model M_1 against M_0 is negligible.

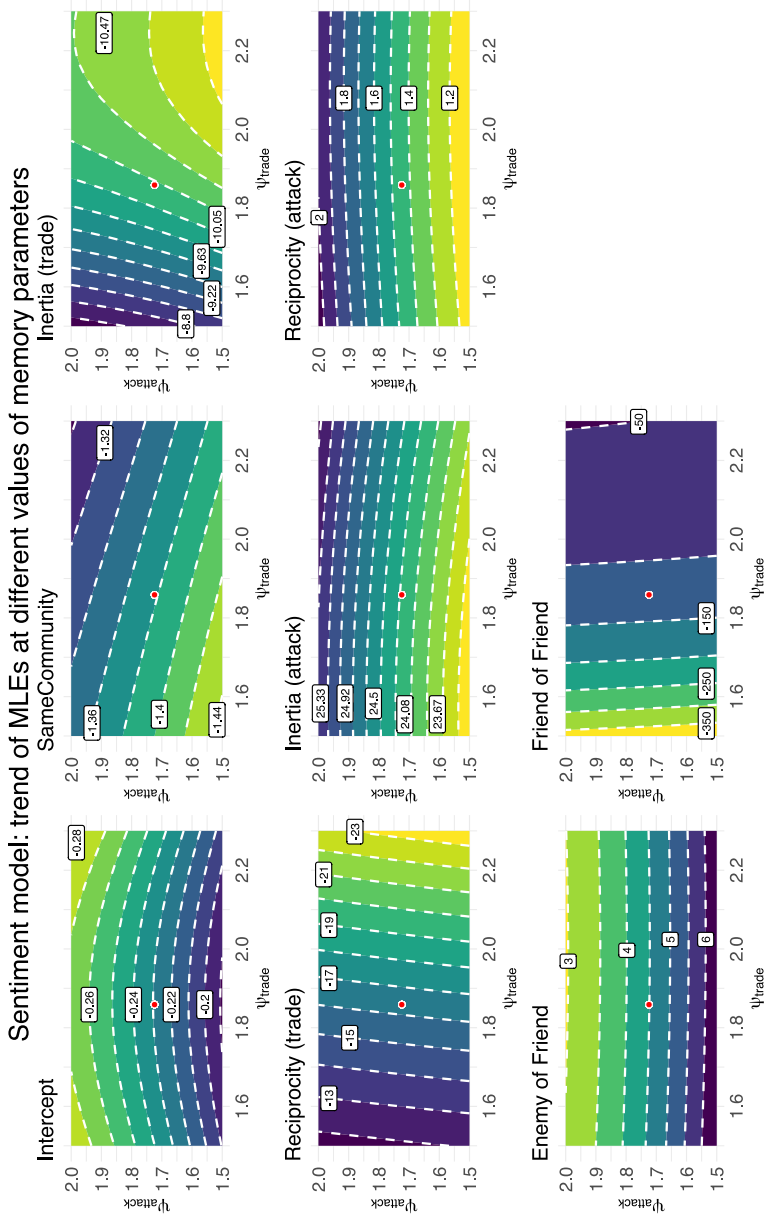


Figure 5.7: (Case study) Trend of the maximum likelihood estimates of sentiment model parameters at different values of memory parameters ($\psi_{\text{trade}}, \psi_{\text{attack}}$).

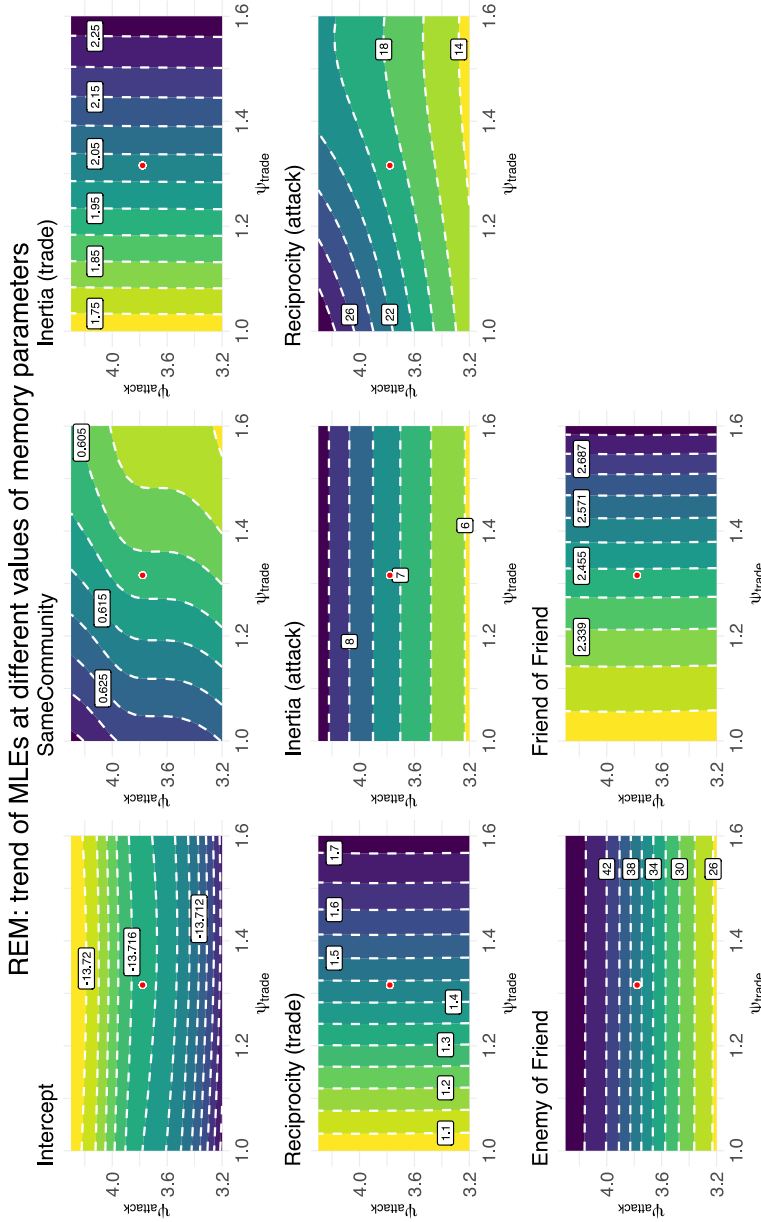


Figure 5.8: (Case study) Trend of the maximum likelihood estimates of relational event model parameters at different values of memory parameters ($\psi_{trade}, \psi_{attack}$).

Bias of estimates $\hat{\beta}$ after misspecification of memory parameters

The optimization of the model parameters in a SentiREM is carried out considering both effects β and log-memory parameters ψ in the optimization algorithm. Several studies in the literature have estimated relational event models that have a memory decay by fixing the memory parameters to a predefined value and considered only the effects β as parameters of interest (Brandenberger, 2018b; Brandes et al., 2009). Even though such an approach can be theory-driven and the results can be insightful on an exploratory level, it might lead to study models in which, given that the memory parameters are fixed to a priori values and not optimized, the resulting estimates $\hat{\beta}$ might be biased and do not describe the real values of the effects (Arena et al., 2023).

Next, we explore the bias of the estimated effects $\hat{\beta}$ in the SentiREM for the empirical data under study. In Figures 5.8 and 5.7, we show the trend of the maximum likelihood estimates of the effects of the statistics for both the REM and the sentiment model across a grid of values of the memory parameters $(\psi_{\text{trade}}, \psi_{\text{attack}})$.

The contour plots show that at different fixed values of $(\psi_{\text{trade}}, \psi_{\text{attack}})$ the resulting estimated effects $\hat{\beta}$ change value as well. This is the case for almost all of the statistics specified in the linear predictor and it happens for both the REM and the sentiment model. For instance, event weights in the endogenous statistic Enemy-of-Friend are based on the memory decay of the attack (l, j) (see Table 5.1), therefore the estimate of the effect β_{EoF} is more likely to change in response to changes of ψ_{attack} than to changes of ψ_{trade} . However, this doesn't happen, for instance, for the Inertia (trade) statistic in the sentiment model or Reciprocity (attack) in the relational event model; the estimated effects of the two statistics show a trend that depends on changes in both memory parameters.

The effects of statistics: interpretation of $\hat{\beta}$ in the SentiREM

In a SentiREM the two sets of effects $(\hat{\beta}^{(sr)}, \hat{\beta}^{(c)})$ assume a different interpretation given the different nature of dyad and sentiment model. In this section we aim to provide an interpretation of endogenous and exogenous statistics for both models defining a SentiREM.

The interpretation of the effects for the exogenous statistics follows the usual interpretation in a REM or in a Probit regression. Therefore, SameCommunity

has a positive effect of 0.607 in the REM model, indicating that if sender and receiver belong to the same alliance, the rate of occurrence of their interaction is the 83% higher than the case in which they belong to different alliances and holding all the other statistics constant ($\exp(.604) = 1.83$). Hence, being in the same alliance (SameCommunity) in the REM promotes the interaction between players. Alternatively, in the Probit model we are modeling the probability of the next event sentiment being an attack (rather than being a cooperative event). In this model, the SameCommunity effect is negative effect: players are more likely to attack players of other alliances and not their allies; belonging to the same alliance decreases the probability of a future attack.

In both models, the interpretation of the effects of the endogenous statistics ($\hat{\beta}$) might not be as straightforward as it is for the exogenous statistics due to the presence of the memory decay functions that are normalized (i.e., the weights for all the possible values of the transpired time add to 1). For example, consider Inertia (trade). In the REM, the estimate for Inertia (trade) is 2.030: the higher the volume of past trades from i to j the higher the future rate by which i targets some form of interaction (either a trade or an attack) to j . If the trade (i, j) is the last occurred event, we multiply the maximum weight for the trades (i.e., $\ln(2)/\exp(\hat{\psi}_{\text{trade}}) = \ln(2)/\exp(1.316) \approx 0.186$) with the estimate for Inertia (trade) ($\exp(2.030 \times 0.186) \approx 1.459$): an increase of about 45.9% of the event rate. If the last trade (i, j) occurred around 4 hours earlier, the event contributes to the future interaction rate with a factor of $\exp(2.030 \times 0.186 \times 0.5) \approx 1.210$, increasing the event rate for the dyad (i, j) with 21%. For the sentiment model, the estimate for statistic Inertia (trade) is -10.170 and the maximum weight for a past trade event is $\ln(2)/\exp(\hat{\psi}_{\text{trade}}) = \ln(2)/\exp(1.859) \approx 0.108$. The effect of the statistic is negative: the more trades i sent to j in the recent past the less likely is that i will attack j in the future. Note that the estimated memory decay is very fast, trades have their weight halved in about 6-7 hours. Thus, if the last trade from i to j is the last occurred event, it will decrease the probability of i attacking j by a factor of $(-10.170 \times 0.108) \approx -1.098$ (on the Z-score scale, of a standard normal distribution). Alternatively, if the last trade (i, j) occurred about 6-7 hours ago, the probability of i attacking j in the next interaction decreases by a factor $(-10.170 \times 0.108 \times 0.5) \approx -0.549$ (on the Z-score scale, of a standard normal distribution). In other words, the more recent trades (i, j) are, the lower the probability of i attacking j .

	BIC					
	MLE	w/o memory	fixed memory			
			1	2	3	4
REM (dyadic event rate)	195498.6	202274.8	197096.7	199552.9	198182.3	200623.8
Sentiment model (probability of attack)	1320.13	1587.79	1349.09	1452.40	1424.07	1563.57

Table 5.4: (Case study) Table with BIC's of the SentiREMs in the comparison: "MLE" is the SentiREM in which memory parameters were estimated by optimization, "w/o memory" is the SentiREM in which no weight decay affects the computation of the endogenous statistics, "fixed memory" (1,2,3 and 4) are the four models in which the memory parameters are fixed to predefined values.

Next, we define three hypotheses of interest:

$$\begin{cases} H_0 : \beta_{\text{trade}} = \beta_{\text{attack}} \\ H_1 : \beta_{\text{trade}} > \beta_{\text{attack}} \\ H_2 : \beta_{\text{trade}} < \beta_{\text{attack}} \end{cases} \quad (5.23)$$

where we write β_{trade} and β_{attack} as a more general notation without specifying statistic and model. The hypotheses can refer to the two Inertia effects or the two Reciprocity effects. Alternatively, we can investigate whether each pair of effects for each dynamic is equal or different across sentiments. Computationally, we use the method of the adjusted fractional Bayes factor with Gaussian approximation (Mulder et al., 2021). Consider the effects of Inertia for trades and attacks: in the sentiment model H_2 shows strong evidence against H_0 ($\log_{10} BF \approx 1.29$) and H_1 (BF is ∞). In the REM, H_2 also shows a very strong evidence against the other two hypotheses (both Bayes Factors are ∞). For Reciprocity, in the sentiment model H_2 has a strong evidence against H_0 ($\log_{10} BF \approx 6.24$) and H_1 ($\log_{10} BF \approx 9.07$). The REM model shows similar results: the evidence supports H_2 against H_0 ($\log_{10} BF \approx 3.07$) and H_1 ($\log_{10} BF \approx 5.79$). In sum, the results suggest that persistence and reciprocity of past attacks generally have a significantly larger effect on the hazard rate and on the probability of attack than past trades.

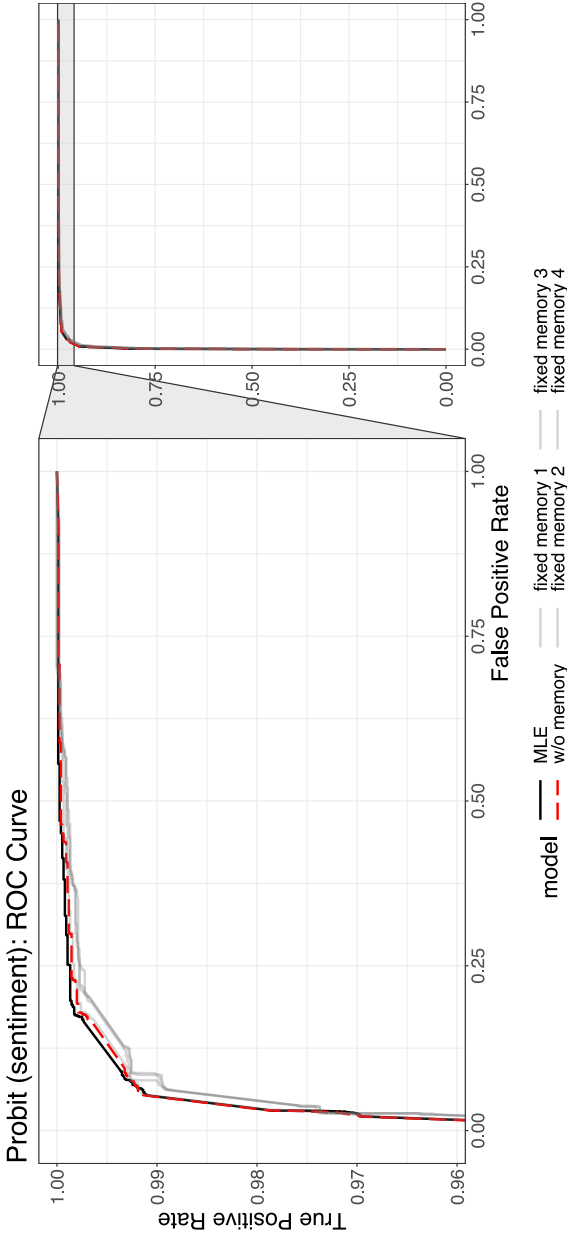


Figure 5.9: (Case study) ROC curves for the Probit model (sentiment): the bold black line is the Probit model in which the memory parameters ($\psi_{\text{trade}}, \psi_{\text{attack}}$) were optimized, the dashed red line is the model without memory (w/o memory) and the solid gray lines are the four fixed memory models, in which the memory parameters were fixed to four combinations of values.

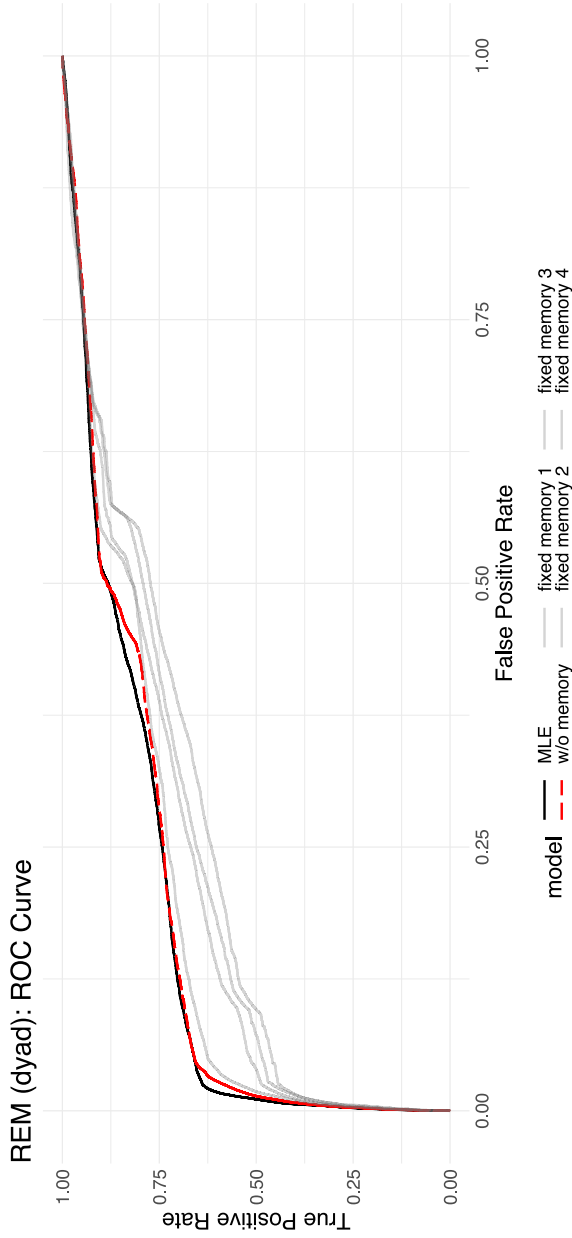


Figure 5.10: (Case study) ROC curves for the REM (dyad): the bold black line is the REM in which the memory parameters ($\psi_{\text{trade}}, \psi_{\text{attack}}$) were optimized, the dashed red line is the model without memory (w/o memory) and the solid gray lines are the four fixed memory models, in which the memory parameters were fixed to four combinations of values.

5.4.3 A comparison of SentiREMs with fixed memory decays or without memory decay

In the literature, the most common approach when embedding event weight decay in a relational event model consists of predefining the decay function by setting its memory parameter to a specific value (Brandenberger, 2018b; Brandes et al., 2009). The result of this approach is that the estimates show the effects of the statistics conditional on the specific value of the memory parameter set by the researcher. Unfortunately, there is no substantive literature to guide researchers in the specification of these values, so the modeling results are based on the value of the memory decay parameters that the researcher is interested in specifically or that we guessed by the researcher (and, as a result, will often be quite arbitrary). In Figures 5.8 and 5.7 we showed that fixing the memory parameters a priori may lead to estimates that are not the most likely for the observed data. Indeed, there will be models where the fixed memory parameters will be very close to their maximum likelihood estimates, and other models where the fixed memory parameters will be far from their best estimates and, in turn, the estimated effects for the statistics will likely not correctly describe the network dynamics in the observed data.

To illustrate what this means for our data, we compare the SentiREM with optimized memory parameters against a set of SentiREMs where the memory parameters for both trades and attacks are fixed to predefined values and against a SentiREM in which no memory decay is assumed. We will refer to the model with optimized memory parameters as "MLE", and to the model without memory as "w/o memory." We define four models by varying the set of memory parameters ($\psi_{\text{trade}}, \psi_{\text{attack}}$) to 12 hrs and 336 hrs (14 days) as follows: $\{(\ln(12), \ln(12)), (\ln(336), \ln(12)), (\ln(12), \ln(336)), (\ln(336), \ln(336))\}$ (in logarithmic scale). We refer to these four models as, respectively, "fixed memory 1," "fixed memory 2," "fixed memory 3," and "fixed memory 4." In Table 5.4 we show the BIC's and show the predictive performance of the models for both the REM and the sentiment model in Figures 5.9 and 5.10. The predictive performance of the SentiREMs under comparison are described by the ROC curve for each model. We use solid gray lines for the four fixed memory models, black solid lines for the "MLE" model, and a red dashed line for the model without memory. We can observe how the BIC of the "MLE" model is the lowest in both REM and sentiment model and also among the ROC curve it performs overall better than all other models.

The information provided by the BIC in Table 5.4, by the ROC curves in Figure 5.9 and 5.10 and by the trend of maximum likelihood estimates in Figure 5.7 and 5.8 confirm that: (i) fitting and predictive performance increase when the memory parameters in a SentiREM are estimated from the data, (ii) every time memory parameters are fixed to a priori values the model loses predictive power, the estimates may suffer from bias with resulting loss of fit, (iii) a model without memory tends to perform better than models with memory decay fixed a priori (this happens so long as the fixed value of the memory parameters is not the same as their maximum likelihood estimates).

5.5 Discussion

In this chapter, we introduced the SentiREM: a model for temporal networks of relational events with sentiment where the next dyadic event is modeled along with the next sentiment. We specified sentiment-based parametric memory decays that are embedded in the likelihood of the model and are estimated along with the effects of the statistics of interest. The proposed model consists of two separable models: the model for the next dyadic event (which is a traditional relational event model) and the model of the next sentiment (which can be modeled via different regression models). In our specific implementation, we assumed the sentiment of the events to be measured on discrete values that can be either positive or negative and modeled the probability of the next sentiment being negative via Probit regression.

We proposed two optimization methods for the likelihood of the SentiREM and discussed meaningful tests on parameters such as testing on memory parameters for different sentiments (so as to understand whether negative events have a differently lasting effect than positive events) and testing on differences between effects of the same network dynamic but calculated on different sentiments (e.g., positive inertia and negative inertia). We provided numerical simulations and an application of the SentiREM to a real case study in which we showed the key features of the model, interpreted and tested model parameters. We also showed that the proposed model results in better performance and fit to empirical data than existing approaches that either fix the memory parameters or do not assume any memory decay. The SentiREM can provide researchers with substantive insights about complex temporal interaction dynamics in real-life social networks, about how long negative and positive events

Chapter 5

affect social interactions, or how the past affects future negative and positive events.

Future work can contribute with sentiment models that are different from the Probit. For instance, if the sentiment is measured on three or more categorical levels, one might use an ordered Logit model. If the event type is measured in counts, one might prefer a Poisson regression where a zero-inflated model extension could be used to easily handle scenarios in which the lowest event type dominates the network of events. If the sentiment lies on a continuous scale, one could either dichotomize this scale and use the outlined Probit model or define another regression model that suits the scale of measurement of the sentiment.

Other future directions might focus on other shapes of memory decay. For instance, we could define a smoothed-one-step memory decay that maintains a constant weight until some elapsed time after which the weight drops exponentially according to some half-life parameter; the decay function might then be described by two memory parameters. Finally, we could also assume that the memory decay differs across senders. This is a realistic assumption in the context of directed interactions and we believe that it can be easily explored.

CHAPTER

DISCUSSION

6

The work presented in this dissertation supports the theory around the presence of memory retention in networks of relational events. The concept of memory retention is measured by means of a memory decay function that describes the trend of the weight of past events based on the time that is elapsed since their occurrence. The value of these weights is used in the computation of the endogenous statistics. The results indicate that the more recent a past event is, the higher its weight is expected to be. The longer-passed the event the lower its weight becomes. Therefore, the memory decay function is a decreasing function and its shape may differ across networks of relational events. In this dissertation, we introduced methods for estimating and testing memory decay functions from an observed sequence of relational events.

In order to approximate the memory decay function without any parametric assumption, in Chapter 3 we first defined a discrete K-step-wise function and then proposed a Bayesian model averaging approach. The approach calculates the weighted average of the step-wise memory decay estimator across a set of step-wise models each one having a different number and size of the steps. In this way, the researcher can understand whether the importance of past events follows some form of decay or not. We introduced two weighting systems, a first one defined on the posterior probabilities of the step-wise models and a second one based on the predictive performance of the step-wise models. This Bayesian semi-parametric approach provided insights around the average most likely shape of memory decay underlying a relational event sequence. The results indicated that the memory decay can be different than a predefined step-wise function (Perry & Wolfe, 2013; Quintane et al., 2013). Indeed, we found that memory decay followed a smooth decreasing function and this was a first insight on the continuous decrease of the effect of past events as the time transpired since their occurrence increases. However, had the real shape of the decay followed any K-step-wise function, the Bayesian model averaging approach would have still been able to estimate a decay close to the step-wise one by resulting in a large posterior probability for a step-wise model. Thereby the methodology would also still be suitable for learning step-wise decay functions if present. In the analysis of a real case study, the researcher can explore the memory shapes of the different network dynamics by first estimating the shape of the decay using simple memory models and then running the methodology on a more complex set of memory models that improve the estimation.

The shape of the memory decay can be also described by smooth parametric functions, such as an exponential decay, a linear decay or other more or less complex parametric functions. For this reason, in Chapter 4 we introduced a relational event model with memory parameters embedded in the likelihood which were estimated along with the effects of the statistics in the hazard rate. We provided a method to estimate the memory parameter given a specific parametric decay and then we proposed a Bayesian approach to compare models with different parametric decays and to find which decay function best fits the observed sequence of relational events. The parametric method showed that a misspecification of the memory parameter brings to incorrect estimates of the other effects in the model, in turn, leading to wrong interpretations about the influence of the statistics in the network.

The memory decay function can also follow a different shape according to the sentiment of the relational event. Indeed, the weight of a negative interaction (e.g., rebuke, intimidation) may decrease slower than the weight of any other positive relational event (e.g., praise, trade). In Chapter 5, we therefore explored this aspect by introducing the SentiREM, that is a new modeling framework where the next dyadic event is modeled separately from the next sentiment. The SentiREM offers several advantages. First, the effect of sentiment-driven network dynamics is estimated separately for two characteristics of the network: in a relational event model where it describes the influence of the dynamic on the rate of any dyadic interaction and in a Probit model where it describes the influence of the past on the probability of observing a negative sentiment. Second, the inclusion of memory decay functions in the likelihood function allows the model to account for the changing influence of past events as time goes by. Third, new network dynamics can be defined by different combinations of the sentiment of the dyadic events. Furthermore, as the parametric model introduced in Chapter 4, also the SentiREM showed the potential bias of the estimates of the effects due to a misspecification of the memory parameters.

The novel methods introduced in this dissertation improved the analyses of relational event sequences on different aspects. From a statistical point of view, they resulted in improved model fitting and improved predictive performance in comparison to other models where the memory decay was either fixed to a prespecified decay function or it was not assumed at all. From a substantive point of view, they resulted in a more detailed understanding of how past events

affects future social interactions in relational event networks . Indeed, given the event weight of the specific past events characterizing the network statistic and the estimated effect of the statistic, a researcher can quantify the effect that a single occurrence of a behavioral pattern has on the event rate or on the probability of the next event sentiment.

However, the three methods also suffer from a few limitations which establish the ground for future research. In the semi-parametric approach, the computation and the estimation of the step-wise models can be expensive with larger networks or when more network dynamics are included in the linear predictor, the number of step-wise models can be insufficient to obtain a good estimate of the memory decay, the weighting systems introduced for the averaging are only two and this can result in a limited comparison between results. Future research can develop new methods and algorithms for: optimizing the computation of step-wise-defined statistics in presence of a larger set of network dynamics as well a larger network of events, estimating the number of step-wise models that is sufficient for the averaging, defining techniques based on bootstrap aggregation of the step-wise models which can improve the model performances, formulating new weighting systems based on the predictive performances of the step-wise models.

Likewise, the two parametric methods presented in this dissertation show several aspects that can be improved in future research. First, three parametric decay functions were introduced having only one memory parameter. Second, the computational burden for the calculation of the endogenous statistics as well as the estimation of the model parameters can dramatically increase with the number of actors and events. Third, the parametric memory decay is assumed to be the same across all the actors. Fourth, the sentiment model is specified only for a discrete and dichotomized sentiment.

We suggest different paths that can help to overcome such limitations. New shapes of parametric decays can be formulated and the computation of endogenous statistics with weight decay functions as well as the estimation algorithms can be improved by means of innovative algorithms that run in parallel on the GPU rather than on the CPU of a computer. The memory decay can be defined on an actor-level, more precisely, on a sender-level. Then the researcher can define methods of hierarchical modeling where the memory parameter is assumed to differ across actors in the network. Finally, the model for the event sentiment can be expanded to other statistical models than the Probit.

Different models can be defined according to the measurement scale of the sentiment, for instance, if the sentiment is measured in counts the researcher can model the sentiment via a Poisson model, if it is measured on an ordinal scale, the researcher can model the sentiment via ordered Logit (or ordered Probit). The sentiment may be also measured on a continuous scale, thus the researcher may either simplify the model by dichotomizing the scale into positive and negative sentiments and use a Probit model or define a continuous model that suits the scale of the sentiment.

Semi-parametric and parametric methods show, each one in their own way, that: (i) there exists some form of memory retention in any observed relational event sequence, (ii) it can differ according to the sentiment of the relational event. The introduction of such methods for relational event models allows researchers to reveal the speed and shape of memory decay about past relational events, improve model fit, predictive performance and avoid potential bias. The results drive the research to more theory building about time-sensitive social interaction dynamics. The evidence on the memory parameters that is collected from similar case studies can form a solid base for the formulation of a unified theory around the concept of memory retention. Different theories may be derived across and within fields of study where relational event sequences are the object of research.

A

A.1 Endogenous statistics

In Table **A.1** the indicator variable for any event e where $(s_e = i, r_e = j)$ follows the short notation $\mathbb{I}_e(i, j)$ and the same applies to any other dyad. Given each statistic, the formula in the first row shows the interval definition of the statistic as regards dyad (i, j) in the k -th interval; whereas, the formula in the second row shows the continuous definition where $\beta(\gamma, \theta)$ is the trend function that follows one of the decays discussed in Section **3.4** or another more complex evolution. Note how in the continuous formulas the event history at t_m , that is $E_{t_{m-1}}$, doesn't depend on any interval.



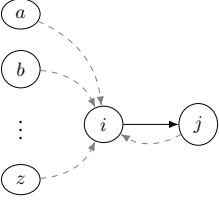
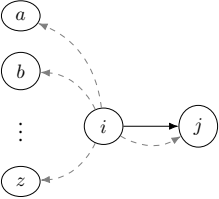
	Endogenous Statistic	Formula
First Order	Inertia 	$\text{inertia}_k(i, j, t_m) = \sum_{e \in E_{t_{m-1}, k}} \mathbb{I}_e(i, j)$ $\text{inertia}(i, j, t_m, \beta(\gamma, \theta)) = \sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(i, j) \beta(\gamma_e(t_m), \theta)$
	Reciprocity 	$\text{reciprocity}_k(i, j, t_m) = \sum_{e \in E_{t_{m-1}, k}} \mathbb{I}_e(j, i)$ $\text{reciprocity}(i, j, t_m, \beta(\gamma, \theta)) = \sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(j, i) \beta(\gamma_e(t_m), \theta)$
	Sender in-degree 	$\text{indegree}_k^{\text{snd}}(i, j, t_m) = \sum_{l \in S \setminus \{i\}} \sum_{e \in E_{t_{m-1}, k}} \mathbb{I}_e(l, i)$ $\text{indegree}^{\text{snd}}(i, j, t_m, \beta(\gamma, \theta)) = \sum_{l \in S \setminus \{i\}} \sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(l, i) \beta(\gamma_e(t_m), \theta)$
	Sender out-degree 	$\text{outdegree}_k^{\text{snd}}(i, j, t_m) = \sum_{l \in S \setminus \{i\}} \sum_{e \in E_{t_{m-1}, k}} \mathbb{I}_e(i, l)$ $\text{outdegree}^{\text{snd}}(i, j, t_m, \beta(\gamma, \theta)) = \sum_{l \in S \setminus \{i\}} \sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(i, l) \beta(\gamma_e(t_m), \theta)$

Table A.1: First and second order endogenous statistics: the formula in the first row is the interval definition of the statistic, whereas the formula in the second row represents the continuous definition of the statistic where the function $\beta(\gamma, \theta)$ describes the decay of the effect.

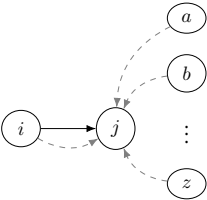
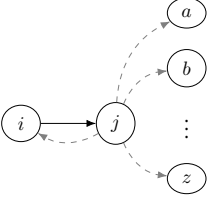
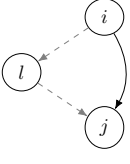
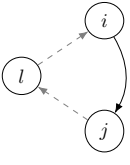
	Endogenous Statistic	Formula
First Order	Receiver in-degree 	$\text{indegree}_k^{\text{rec}}(i, j, t_m) = \sum_{l \in \mathcal{S} \setminus \{j\}} \sum_{e \in E_{t_{m-1}, k}} \mathbb{I}_e(l, j)$ $\text{indegree}^{\text{rec}}(i, j, t_m, \beta(\gamma, \theta)) = \sum_{l \in \mathcal{S} \setminus \{j\}} \sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(l, j) \beta(\gamma_e(t_m), \theta)$
	Receiver out-degree 	$\text{outdegree}_k^{\text{rec}}(i, j, t_m) = \sum_{l \in \mathcal{S} \setminus \{j\}} \sum_{e \in E_{t_{m-1}, k}} \mathbb{I}_e(j, l)$ $\text{outdegree}^{\text{rec}}(i, j, t_m, \beta(\gamma, \theta)) = \sum_{l \in \mathcal{S} \setminus \{j\}} \sum_{e \in E_{t_{m-1}}} \mathbb{I}_e(j, l) \beta(\gamma_e(t_m), \theta)$
Second Order	Transitivity closure 	$\text{transitivity closure}_k(i, j, t_m) = \sum_{l \in \mathcal{S} \setminus \{i, j\}} \sum_{e \in E_{t_{m-1}, k}} \sum_{\substack{e^* \in E_{t_{m-1}} \\ t_{e^*} \in [t_e - \gamma_e(t_m), t_e]}} \mathbb{I}_e(l, j) \mathbb{I}_{e^*}(i, l)$ $\text{transitivity closure}(i, j, t_m, \beta(\gamma, \theta)) = \sum_{l \in \mathcal{S} \setminus \{i, j\}} \sum_{e \in E_{t_{m-1}}} \sum_{\substack{e^* \in E_{t_{m-1}} \\ t_{e^*} \in [t_e - \gamma_e(t_m), t_e]}} \mathbb{I}_e(l, j) \mathbb{I}_{e^*}(i, l) \beta(\gamma_e(t_m), \theta)$
	Cyclic closure 	$\text{cyclic closure}_k(i, j, t_m) = \sum_{l \in \mathcal{S} \setminus \{i, j\}} \sum_{e \in E_{t_{m-1}, k}} \sum_{\substack{e^* \in E_{t_{m-1}} \\ t_{e^*} \in [t_e - \gamma_e(t_m), t_e]}} \mathbb{I}_e(l, i) \mathbb{I}_{e^*}(j, l)$ $\text{cyclic closure}(i, j, t_m, \beta(\gamma, \theta)) = \sum_{l \in \mathcal{S} \setminus \{i, j\}} \sum_{e \in E_{t_{m-1}}} \sum_{\substack{e^* \in E_{t_{m-1}} \\ t_{e^*} \in [t_e - \gamma_e(t_m), t_e]}} \mathbb{I}_e(l, i) \mathbb{I}_{e^*}(j, l) \beta(\gamma_e(t_m), \theta)$

Table A.1: First and second order endogenous statistics: the formula in the first row is the interval definition of the statistic, whereas the formula in the second row represents the continuous definition of the statistic where the function $\beta(\gamma, \theta)$ describes the decay of the effect. (continued)

A

A.2 From step-wise to continuous effects

Consider an increasing sequence of $K + 1$ time widths $\gamma = (\gamma_0, \gamma_1, \dots, \gamma_K)$, such that $\gamma_k - \gamma_{k-1} = \Delta$ for $k = 1, \dots, K$ (i.e., evenly spaced intervals). A graphical representation of intervals at t_m is presented below in Figure [A.1](#).

In the context of endogenous statistics that are defined on intervals (see Section [B.3.1](#)), one could already apply the formulas in Appendix [A.1](#) and then esti-

Appendix A

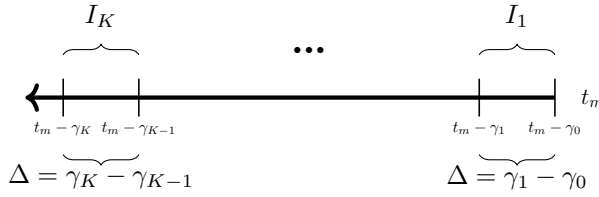


Figure A.1: K evenly spaced intervals, with time widths $\gamma = (\gamma_0, \gamma_1, \dots, \gamma_K)$ such that $\gamma_k - \gamma_{k-1} = \Delta$ for $k = 1, \dots, K$.

mate the step-wise trend for each network statistic of interest. In general, when intervals are evenly spaced, we could write $\gamma_k = k \cdot \frac{\gamma_K}{K}$ for $k = 0, \dots, K$, where γ_K is the largest observable width (it can be the length of the study itself). If the number of intervals (K) increases, their size (Δ), in turn, shrinks. Indeed, considering the size of an interval that is calculated as the difference between two adjacent widths, $\Delta = (\gamma_k - \gamma_{k-1})$.

$$\lim_{K \rightarrow \infty} (\gamma_k - \gamma_{k-1}) = \lim_{K \rightarrow \infty} \left[k \cdot \frac{\gamma_{max}}{K} - (k-1) \cdot \frac{\gamma_{max}}{K} \right] = \lim_{K \rightarrow \infty} \frac{\gamma_{max}}{K} = 0$$

This result holds for $k = 1, \dots, K$. Therefore, an extreme scenario consists in a large number of intervals whose sizes are so small that at t_m each of them contains only one or no relational event. As a consequence of this, one would estimate a step-wise trend where each step is defined approximately on a value of the transpired time and it represents the relative effect based on those events that assumed that specific value throughout the event histories (E_t , with $t = t_1, \dots, t_M$). Indeed, any event since its occurrence assumes a value reflecting its recency that is updated at every time point onward and, thus, it increases over time (from t_1 to t_M if considering the time points where events were observed). Every value of transpired time calculated at each time point can be observed at least once in the network and when it is observed multiple times this happens at different time points. For instance, two different events could both occur 33 minutes earlier than the present time point but with the condition that the present time point they refer to is different for both of them (because events are assumed not to occur at the same time point). Finally, the estimation of the effects over such a large number of intervals is impractical and it serves only to convey insights about the possibility of continuously changing effects in contrast to step-wise decays.

A.3 Interval generator (the algorithm)

Algorithm 1: Generating S intervals with K steps (having either increasing or decreasing size).

set (*inputs and memory allocation for the output*):

$K \leftarrow$ number of steps for each sequence

$S \leftarrow$ number of sequences to generate

$s = 1 \leftarrow$ starting with generating the first sequence

$min_size \leftarrow$ minimum size intervals

$\gamma_K \leftarrow$ maximum time width that each sequence can reach

$decreasing \leftarrow$ logical TRUE/FALSE whether to generate either decreasing size or increasing size intervals

$\mathcal{W} \leftarrow$ empty matrix of dimensions $[S \times K]$ where to store the generated sequences of widths $\gamma = (\gamma_1, \dots, \gamma_K)$ (excluding γ_0 that is equal to zero by default)

while $s \leq S$ **do**

generate $\xi \sim Dir(K, \alpha)$ with $\alpha = \mathbf{1}_K$;

sort ascending ξ ;

while $\min \{\xi\} < min_size$

generate $\xi \sim Dir(K, \alpha)$ with $\alpha = \mathbf{1}_K$;

sort ascending ξ ;

if $decreasing = TRUE$ **then sort descending** ξ ;

$\gamma \leftarrow$ **cumulative sum** of ξ ;

update $\gamma = \gamma * \gamma_K$;

save γ in the s -th row of \mathcal{W} ;

update $s = s + 1$;

return \mathcal{W} ;

A.4 Maximum likelihood estimates for the models specified in the model comparison

In Table [A.2](#) the maximum likelihood estimates for the models specified in Section [3.6.4](#). In the calculation of the BIC, the penalization accounts for the number of parameters in each model (# parameters). In the step-wise models (StepEqual, StepIncr and bestWAIC) the effect of each statistic in each interval is reported (e.g., in model StepEqual, inertia has two effects $\beta_{\text{inertia}_1} = 0.05$ and $\beta_{\text{inertia}_2} = 0.03$, and so the other statistics).

	BMA	w/o memory	Exp 7	Exp 30	Exp 90	StepEqual	StepIncr	bestWAIC
Inertia	7.091* (0.07)	23.979* (0.29)	5.157* (0.05)	6.48* (0.07)	7.15* (0.08)	0.05* (0.001) 0.03* (0.002)	0.18* (0.008) 0.05* (0.004) 0.03* (0.002) 0.03* (0.002)	0.55* (0.02) 0.08* (0.006) 0.04* (0.004) 0.03* (0.001)
Reciprocity	2.993* (0.11)	9.414* (0.48)	2.443* (0.08)	2.87* (0.1)	2.91* (0.1)	0.02* (0.002) 0.01* (0.002)	0.05* (0.013) 0.01* (0.006) 0.02* (0.003) 0.01* (0.002)	0.27* (0.03) 0.02 (0.009) 0.02* (0.005) 0.01* (0.001)
Transitivity Closure	0.264* (0.009)	0.083* (0.004)	0.872* (0.03)	0.298* (0.009)	0.106* (0.004)	0.001* (0.0002) 0.0006* (6.19E-05)	0.02* (0.007) 0.001* (0.001) 0.001* (0.0002) 0.0006* (6.25E-05)	0.289* (0.04) 0.002 (0.003) 0.002* (0.001) 0.001* (4.41E-05)
Intercept	-4.367* (0.17)	-3.997* (0.015)	-4.162* (0.015)	-4.33* (0.017)	-4.347* (0.017)	-4.353* (0.017)	-4.354* (0.017)	-4.349* (0.017)
# parameters	4	4	4	4	4	7	13	13
BIC	56436.84	60735.04	58262.89	56915.38	56799.1	56917.46	56696.91	56246.77

Table A.2: Maximum likelihood estimates for the models specified in Section [3.6.4](#).

B

B.1 Weights following a step-wise function

In Perry and Wolfe (2013) endogenous statistics are calculated based on a set of intervals of the transpired time of past events. Consider a vector of $K + 1$ increasing widths $\gamma = (\gamma_0, \dots, \gamma_K)$ with $\gamma_0 < \dots < \gamma_K$ and a network dynamic like inertia. After we calculate inertia in the intervals at all the time points, the estimated effects define a step-wise function for the effect of the specific network dynamic at any time point in the event sequence. The step-wise effect function is described by $\beta_{\text{inertia}} = (\beta_{\text{inertia}_1}, \dots, \beta_{\text{inertia}_K})$

$$\beta_{\text{inertia}_1} \sum_{\substack{e' \in E_{t_m-1}: \\ (t_m - t_{e'}) \in (\gamma_0, \gamma_1]}} \mathbb{I}(s(e') = i, r(e') = j) + \dots + \beta_{\text{inertia}_K} \sum_{\substack{e' \in E_{t_m-1}: \\ (t_m - t_{e'}) \in (\gamma_{K-1}, \gamma_K]}} \mathbb{I}(s(e') = i, r(e') = j) \quad (\text{B.1})$$

where per each interval in $k = 1, \dots, K$ the effect β_{inertia_k} multiplies by the value of the inertia computed in the k -th interval at time t_m . The (B.1) can be rewritten in a way similar to (4.2) where the weight decay this time follows a step-wise function defined on the vector of increasing widths γ . Indeed, by considering the same vector of widths and associating a vector of K weights $w = (w_1, \dots, w_K)$ to each interval the step-wise weight decay becomes

$$w(\gamma) = \begin{cases} w_1 & \text{if } \gamma \in (\gamma_0, \gamma_1] \\ \vdots & \\ w_K & \text{if } \gamma \in (\gamma_{K-1}, \gamma_K] \\ 0 & \text{otherwise} \end{cases} \quad (\text{B.2})$$

Hence, the statistic can be written as a weighted sum as in (4.2) but in this case following a step-wise decay for the weights,

$$\beta_{\text{inertia}} \text{inertia}(i, j, t_m) = \beta_{\text{inertia}} \sum_{e' \in E_{t_m-1}} \mathbb{I}(s(e') = i, r(e') = j) w(t_m - t_{e'}) \quad (\text{B.3})$$

where $w(t_m - t_{e'})$ follows the step-wise function in (B.2). Considering that weights are the same within each interval, the sum in (B.3) can be re-arranged as follows

$$\begin{aligned}
\beta_{\text{inertia}} \text{inertia}(i, j, t_m) &= \beta_{\text{inertia}} \left[\sum_{\substack{e' \in E_{t_{m-1}}: \\ (t_m - t_{e'}) \in (\gamma_0, \gamma_1]}} \mathbb{I}(s(e')=i, r(e')=j) w_1 + \dots \right. \\
&\quad \left. \dots + \sum_{\substack{e' \in E_{t_{m-1}}: \\ (t_m - t_{e'}) \in (\gamma_{K-1}, \gamma_K]}} \mathbb{I}(s(e')=i, r(e')=j) w_K \right] = \quad (\text{B.4}) \\
&= \beta_{\text{inertia}} w_1 \sum_{\substack{e' \in E_{t_{m-1}}: \\ (t_m - t_{e'}) \in (\gamma_0, \gamma_1]}} \mathbb{I}(s(e') = i, r(e') = j) + \dots \\
&\quad \dots + \beta_{\text{inertia}} w_K \sum_{\substack{e' \in E_{t_{m-1}}: \\ (t_m - t_{e'}) \in (\gamma_{K-1}, \gamma_K]}} \mathbb{I}(s(e') = i, r(e') = j)
\end{aligned}$$

The (B.4) is exactly the same formula in (B.1) with the only difference that here the step-wise function of weights is explicitly written. Therefore, the equivalence between the two vectors of effects can be written as follows

$$\begin{bmatrix} \beta_{\text{inertia}_1} \\ \vdots \\ \beta_{\text{inertia}_K} \end{bmatrix} = \begin{bmatrix} \beta_{\text{inertia}} w_1 \\ \vdots \\ \beta_{\text{inertia}} w_K \end{bmatrix} \quad (\text{B.5})$$

The same idea of a changing weight of events according to their time recency is also here but it is proposed from a different perspective. In the specific case of a step-wise function, the number of steps and their widths are the parameters describing the function.

B.2 Sms data (sub-networks with 1 cluster and 2 clusters): trend of MLEs when the weight decay is exponential

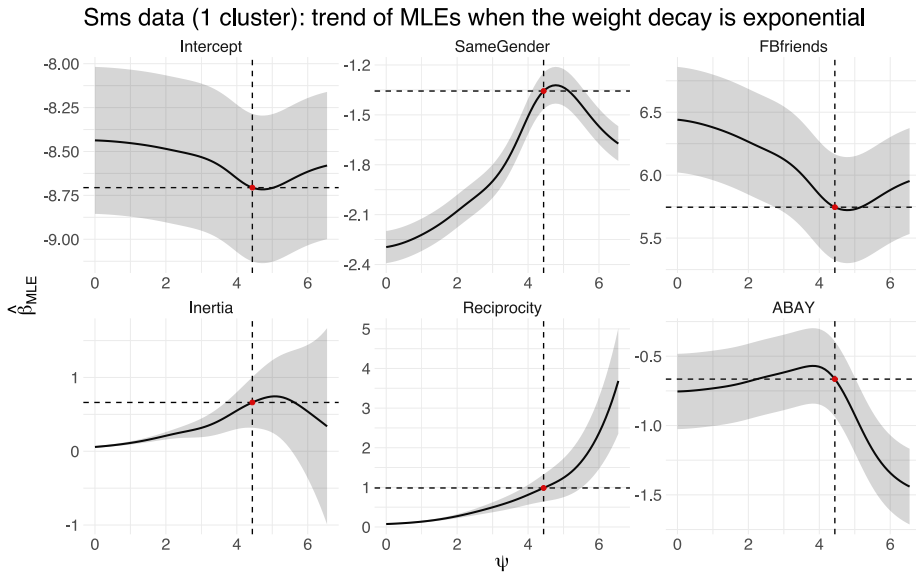


Figure B.1: Sms data (1 cluster): trend of the maximum likelihood estimates (MLEs) for the exponential decay over ψ (logarithm of the memory parameter). The dashed black lines in each plot mark the estimate for the log-memory-parameter $\hat{\psi}_{MLE}$ (vertical lines) and the estimates of the effects β (horizontal lines) at the corresponding $\hat{\psi}_{MLE}$. The shaded regions are the confidence intervals at 0.95 for the effects β estimated at any value of ψ .

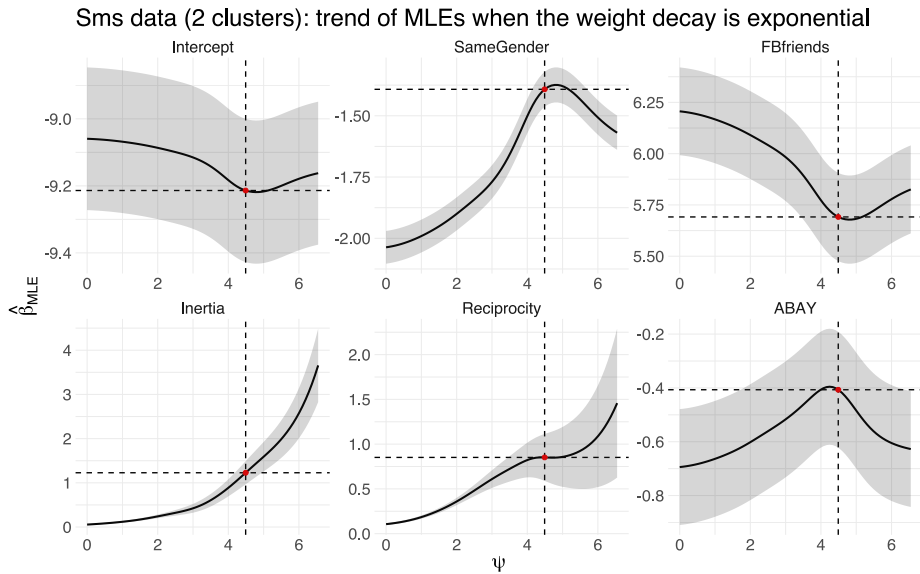


Figure B.2: Sms data (2 clusters): trend of the maximum likelihood estimates (MLEs) for the exponential decay over ψ (logarithm of the memory parameter). The dashed black lines in each plot mark the estimate for the log-memory-parameter $\hat{\psi}_{MLE}$ (vertical lines) and the estimates of the effects β (horizontal lines) at the corresponding $\hat{\psi}_{MLE}$. The shaded regions are the confidence intervals at 0.95 for the effects β estimated at any value of ψ .

SOFTWARE PACKAGES FOR THE ANALYSIS
OF RELATIONAL EVENT NETWORKS



The methods and analysis presented in this dissertation were carried out by means of software packages developed in R (R Core Team, 2022) with algorithms and functions written in R or C/C++. In this appendix, we present three software packages :

- `remify` (Arena, 2022b): a package for pre-processing raw relational event sequences
- `remstimate` (Arena, Lakdawala, et al., 2022): a package for the optimization of tie-oriented or actor-oriented models
- `bremory` (Arena, 2022a): a package for modeling the influence of past social interactions in relational event networks. This package contains the functions to perform the methods presented in this dissertation

C.1 `remify` : pre-processing raw relational event sequences

In the processing of relational event history (REH) data, the `remify` package (Arena, 2022b) aims to:

1. process REH data and arrange them in a new structure of class `reh` ;
2. transform relational event history data from other formats or other sources to a `reh` structure (or vice versa).

The two main functions `reh()` and `rehshape()` perform respectively the processing and the conversion of the REH data. We discuss the two functions according to the latest version `2.0.0` of the package.

C.1.1 A function for processing raw data

The function `reh()` prepares the raw data along with other inputs that characterize the relational event sequence (actors' names, event types' names, starting time point of the event sequence, set of interactions to be excluded from the risk set at specific time points, etc.). The internal routines will transform the structure of the input event sequence into a new one where actors and event types will have assigned unique identification numbers (IDs). Furthermore, the output object of the function will provide other information that will be required by `remstimate` , `bremory` and `remstats` (Meijerink-Bosman et al., 2023).

As example, the list `randomREH` available inside the package will be used (documentation available via `?randomREH`)

```
library(remify)      # loading library
data(randomREH)     # loading data
names(randomREH)    # objects inside the list 'randomREH'
## [1] "edgelist" "actors"  "types"  "origin" "omit_dyad"
```

Input arguments

We discuss each input argument that can be supplied to `remify::reh()` :

edgelist The edgelist must be a `data.frame` or a `matrix` and column names must be: `"time"` , `"actor1"` , `"actor2"` , `"type"` and `"weight"` . The compulsory columns are `time` , `actor1` and `actor2` whereas it is not mandatory to supply a column for `type` or `weight` . However, any column in the edgelist has to follow the naming as defined above and the order of the columns can vary.

```
head(randomREH$edgelist)
##           time actor1 actor2      type
## 1 2020-03-05 02:47:08 Kayla Kiffani competition
## 2 2020-03-05 02:50:18 Colton Justin      conflict
## 3 2020-03-05 03:30:26 Kelsey Maya      cooperation
## 4 2020-03-05 03:38:50 Alexander Colton competition
## 5 2020-03-05 03:56:16 Wyatt Kelsey      conflict
## 6 2020-03-05 04:06:45 Derek Breanna competition
```

actors The vector of actor names (if left unspecified, names will be taken from the input edgelist). Their data type can be either `numeric` or `character` . In the `randomREH` data, a vector of actor names is provided.

```
randomREH$actors
## [1] "Crystal" "Colton"  "Lexy"   "Kelsey"
↪ "Michaela"
## [6] "Zackary"  "Richard" "Maya"   "Wyatt"
↪ "Kiffani"
## [11] "Alexander" "Kayla"  "Derek"  "Justin"  "Andrey"
## [16] "Francesca" "Megan"  "Mckenna" "Charles" "Breanna"
```

types The vector of type names (if left unspecified, names will be taken from the input `edgelist`). The data type can be either `numeric` or `character` . In the `randomREH` data a vector of types is provided.

```
randomREH$types
## [1] "conflict" "competition" "cooperation"
```

directed A `logical` value indicating whether events are directed (`TRUE`) or not (`FALSE`). If dyadic events are undirected, the names of the actors of any observed event will be sorted by following their alphanumerical order (e.g. `[actor1,actor2] = ["Maya","Lexy"]` will become `[actor1,actor2] = ["Lexy","Maya"]`).

ordinal A `logical` value indicating whether we want to model only the sequence of events like in a Cox model (`TRUE`) or we want to model the event sequence along with the waiting times between events (`FALSE`).

origin If the initial time point of the event sequence, t_0 , is known it can be supplied to the argument `origin` and it must have the same class of the time variable specified in the column `time` of the edge-list. In the case the argument `origin` is left unspecified (`NULL`), it is set by default to one time unit earlier than t_1 (time of occurrence of the first relational event). For instance, when time is measured in seconds then $t_0 = t_1 - 1sec$, when measured in minutes then $t_0 = t_1 - 1min$, when measured in hours then $t_0 = t_1 - 1hr$, when measured in days then $t_0 = t_1 - 1day$ and so forth. In the `randomREH` data a origin t_0 is provided.

```
randomREH$origin
## [1] "2020-03-05 02:32:53 CET"
```

omit_dyad This argument is useful when certain dyads must be removed from the risk set in specific time windows (e.g. an actor drops out of the network, specific groups of actors cannot interact anymore starting from some time point). Therefore, the processing of such information makes the risk set represent better the real data. `omit_dyad` consists of a list of lists, and each list must have two objects named:

`dyad`, that is a `data.frame` where to specify the dyads to be removed from the risk set, and `time` which is a vector of two values defining the first and last time point of the time window in which such dyads couldn't occur. Consider the example on the `randomREH` data. For instance, we want to modify the risk set according to two changes that apply on different time intervals:

1. a first change that excludes events with `type = "conflict"` from the risk set since a specific time point until the end of the event sequence

```
# start and stop time point defining the time window
randomREH$omit_dyad[[1]]$time
## [1] "2020-05-07 22:42:38 CEST" "2020-05-23 23:46:41
↪ CEST"

# dyads to be removed from the risk set during the time
↪ window
randomREH$omit_dyad[[1]]$dyad
##  actor1 actor2  type
## 1     NA     NA conflict
```

2. a second change in which two actors `"Michaela"` and `"Zackary"` couldn't interact with anybody else after a specific time point and until the end of the observation period

```
# start and stop time point defining the time window
randomREH$omit_dyad[[2]]$time
## [1] "2020-05-20 01:30:09 CEST" "2020-05-23 23:46:41
↪ CEST"

# dyads to be removed from the risk set during the time
↪ window
randomREH$omit_dyad[[2]]$dyad
##  actor1 actor2 type
## 1 Michaela <NA>  NA
## 2 <NA> Michaela NA
## 3 Zackary  <NA>  NA
## 4 <NA> Zackary  NA
```

The object `dyad` will give instructions such that the function will remove from the risk set at the indicated time windows all

the events where: (1) `type = "conflict"` , (2) `"Michaela"` and `"Zackary"` are senders or receivers.

Every time one field among `actor1` , `actor2` , `type` is left undefined (`<NA>`), the omission from the risk set applies to all the possible values of that field. Indeed, in the first change where we needed to remove all the events where `conflict` was the type, we did it by leaving both columns `actor1` and `actor2` unspecified (`<NA>`). In the second change, the first row of the `dyad` object is `Michaela <NA> NA` and it describes all the dyadic events in which the sender is `"Michaela"` , the receiver is any possible receiver and the event type is any of the possible types (`"conflict"` and `"cooperation"`).

model Whether the modeling framework of interest is tie or actor oriented, the argument `model` can assume either value `"tie"` or `"actor"`

Structure of the output object

We run the processing function on the data `randomREH` ,

```
edgelist_reh <- reh(edgelist = randomREH$edgelist,
  actors = randomREH$actors,
  types = randomREH$types,
  directed = TRUE,           # events are directed
  ordinal = FALSE,         # REM with waiting times
  origin = randomREH$origin,
  omit_dyad = randomREH$omit_dyad,
  model = "tie")           # tie-oriented modeling
```

The output saved in `edgelist_reh` is an S3 object of class `reh` and contains the following objects:

M The number of observed relational events (also referred to as the length of the event sequence)

```
edgelist_reh$M
## [1] 9915
```

N The total number of actors that could interact in the network

```
edgelist_reh$N
```

```
## [1] 20
```

- C** The number of event types (also referred to as the sentiment of the event) that could be observed in the network.

```
edgelist_reh$C
```

```
## [1] 3
```

- D** The number of possible dyads that can be observed in the network and accounts for the event types:

- if the network is directed, then $D = N(N-1)C$
- if the network is undirected, then $D = (N(N-1)/2)C$

D also represents the largest observable size of the risk set.

```
edgelist_reh$D
```

```
## [1] 1140 # that is 20*19*3
```

- IntereventTime** A `numeric` vector of waiting times between two subsequent events, that is

$$\begin{bmatrix} t_1 - t_0 \\ t_2 - t_1 \\ \dots \\ t_M - t_{M-1} \end{bmatrix}$$

```
head(edgelist_reh$intereventTime)
```

```
##           [,1]
## [1,]  854.6961
## [2,]  189.9698
## [3,] 2408.3461
## [4,]  504.0680
## [5,] 1046.0560
## [6,]  628.0785
```

- edgelist** A `data.matrix` that consists of the converted input edgelist, with columns named `"time"`, `"dyad"`, `"weight"`. From the input edgelist, the columns `"actor1"`, `"actor2"` and

"type" are converted into a numeric ID value ranging in $\{0, \dots, D - 1\}$ and it is the value of the column "dyad". This choice is made because internal routines in other packages like `remstimate` and `bremory` are written in C++.

```
head(edgelist reh$edgelist)
##           time dyad weight
## [1,] 1583372829 181     1
## [2,] 1583373019 463     1
## [3,] 1583375427 962     1
## [4,] 1583375931   3     1
## [5,] 1583376977 732     1
## [6,] 1583377605 116     1
```

If "weight" is not supplied as input, then its corresponding column in the output edgelist will be a vector of 1's.

omit_dyad

If the argument `omit_dyad` is supplied to the processing function, then the output object will contain the processed list under the same name. In the case of tie-oriented modeling, the list contains two objects:

- a `matrix` named "riskset" that assumes value 1 if the dyad is at risk, 0 otherwise.

All the possible risk set modifications occurring in the event sequence are described by row, thus the number of rows is potentially variable across different specifications of the input `omit_dyad`. The columns identify all the possible the dyads in the sequence (D columns).

- a vector named "time" that for each time point indicates which modification of the risk set (row index of the `matrix` "riskset") is observed.

```
edgelist_reh$omit_dyad$riskset[,1:10] # printing out the risk set
# modifications of only the first 10 columns (dyads).
# Two modifications of the risk set are observed (by row)
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]   1   1   1   1   1   1   1   1   1   1
## [2,]   1   1   1   1   1   1   1   1   1   1
```

The processed object "omit_dyad" will be required by other packages like `remstimate` and `bremory` to handle time-

varying risk sets. Given that such packages have algorithms written in C++, the values of the vector `"time"` start at 0 (indicating the first row of the object `"riskset"`) and they assume value `-1` when no risk set alteration is observed at a specific time point.

```
edgelist_reh$omit_dyad$time[1:10] # printing out the
# first 10 time points. We can see that in none of
# the 10 time points any modification takes place (-1)
## [1] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

In the case of actor-oriented modeling with dynamic risk set, the output list `"omit_dyad"` consists of three objects: a vector named `"time"` (same vector as explained above) and two risk set matrices named `"senderRiskset"` and `"riskset"`, describing, respectively, the time-varying risk set for the sender activity model and the time-varying risk set for the receiver choice model. Both risk set matrices follow the same structure as the `"riskset"` object in the tie-oriented modeling. Whenever the argument `omit_dyad` is not supplied to the function `remify::reh()`, then its processed output in the `reh` object will consist of an empty list.

Attributes

The attributes of the output `reh` object are

```
names(attributes(edgelist_reh))
## [1] "names"      "class"      "with_type"  "weighted"  "directed"
## [6] "ordinal"    "model"      "riskset"    "dictionary" "time"
```

names The vector of the names of the output objects which are discussed in the previous section.

```
attr(edgelist_reh, "names")
```

```
## [1] "M"           "N"           "C"           "D"
## [5] "intereventTime" "edgelist"    "omit_dyad"
```

class The class name of the output object, that is `"reh"`

```
attr(edgelist_reh, "class")
## [1] "reh"
```

with_type A `logical` value indicating whether more than one event type is observed in the network (`TRUE`) or not (`FALSE`).

```
attr(edgelist_reh, "with_type")
## [1] TRUE
```

weighted A `logical` value indicating whether relational events have weights (`TRUE`) or not (`FALSE`).

```
attr(edgelist_reh, "weighted")
## [1] FALSE
```

directed A `logical` value indicating whether we know (`TRUE`) for each event whom originated the action (sender) and whom was the target (receiver) of the interaction, or we don't know (`FALSE`) the source and the target of an event but only the actors that were involved in it.

```
attr(edgelist_reh, "directed")
## [1] TRUE
```

ordinal A `logical` value indicating whether in the model we want to consider the waiting times between events (`FALSE`) or consider only the time order of the relational events (`TRUE`).

```
attr(edgelist_reh, "ordinal")
## [1] FALSE
```

model A `character` value that describes whether the output object is suitable for the actor-oriented model (`"actor"`) or for the tie-oriented model (`"tie"`).

```
attr(edgelist_reh, "model")
## [1] "tie"
```

riskset A `character` value that describes the type of risk set that resulted from the processing of the data. If `omit_dyad` is provided as input, this means that the risk set is going to change for certain time windows and then the argument value is `"dynamic"` . Otherwise, when `omit_dyad = NULL` , the risk set is assumed to be the largest observable one (of dimension D) and it maintains the same structure throughout the network the argument is `"static"` .

```
attr(edgelist_reh, "riskset")
## [1] "dynamic"
```

dictionary A `list` of two `data.frame` named `"actors"` and `"types"` :

- the `data.frame` `"actors"` has two columns: the first column with actor names (`"actorName"`) sorted according to their alphanumerical order, the second column with their corresponding ID's (`"actorID"`), ranging in $\{0, \dots, N-1\}$
- the `data.frame` `"types"` has two columns: the first column with type names (`"typeName"`) sorted according to their alphanumerical order, the second column with their corresponding ID's (`"typeID"`), ranging in $\{0, \dots, C-1\}$

```
attr(edgelist_reh, "dictionary")
## $actors
##   actorName actorID
## 1 Alexander     0
## 2   Andrey     1
## 3   Breanna     2
## 4   Charles     3
## 5   Colton     4
## 6   Crystal     5
## 7   Derek      6
```

```
## 8 Francesca 7
## 9 Justin 8
## 10 Kayla 9
## 11 Kelsey 10
## 12 Kiffani 11
## 13 Lewy 12
## 14 Maya 13
## 15 Mckenna 14
## 16 Megan 15
## 17 Michaela 16
## 18 Richard 17
## 19 Wyatt 18
## 20 Zackary 19
##
## $types
##      typeName typeID
## 1 competition 0
## 2 conflict 1
## 3 cooperation 2
```

time A list of three objects that are named "class", "value" and "origin" :

- "class" is a character value that returns the class of the column "time" provided in the input `edgelist`
- "value" is a `data.frame` of size $[M \times 2]$ where the first column is the variable "time" supplied in the input `edgelist` and the second column is the object "intereventTime" inside the output `reh` object.
- "origin" is the input argument already discussed in the previous section about input arguments of the function `reh()` .

```
str(attr(edgelist_reh, "time")) # printing out only the str()
# of the attribute since the data.frame `value` is large
## List of 3
## $ class : chr [1:2] "POSIXct" "POSIXt"
## $ value : 'data.frame': 9915 obs. of 2 variables:
## ..$ time : POSIXct[1:9915], format: "2020-03-05
↪ 02:47:08" "2020-03-05 02:50:18" ...
```

```
## ..$ intereventTime: num [1:9915] 855 190 2408 504 1046 ...
## $ origin: POSIXct[1:1], format: "2020-03-05 02:32:53"
```

Methods

The methods available for an `reh` object are:

summary Prints out a brief summary of the relational network data.

```
summary(edgelist_reh)
## Relational Event Network
## (processed for tie-oriented modeling):
## > events = 9915
## > actors = 20
## > (event) types = 3
## > riskset = dynamic
## > directed = TRUE
## > ordinal = FALSE
## > weighted = FALSE
## > time length ~ 80 days
## > interevent time
##     >> minimum ~ 0.0011 seconds
##     >> maximum ~ 5811.4011 seconds
```

dim Returns some useful dimensions characterizing the network, such as: number of events, number of actors, number of event types, number of dyads.

```
dim(edgelist_reh)
## events actors types dyads
## 9915     20     3  1140
```

getDynamicRiskset If the input argument `omit_dyad` is supplied to the processing function, then the method returns the processed dynamic risk set matrix (or the two dynamic risk sets if actor-oriented modeling), the same that are explained in the previous section about the output.

```
getDynamicRiskset(edgelist_reh)$riskset[,1:9]
# printing out the risk set modifications of
# only the first 9 columns (dyads).
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]  1   1   1   1   1   1   1   1   1
## [2,]  1   1   1   1   1   1   1   1   1
```

actorName After supplying one or more actorID's the method returns the corresponding (input) names.

```
actorName(reh = edgelist_reh, actorID = c(0,12,19))
## [1] "Alexander" "Lexy"      "Zackary"
```

typeName After supplying one or more typeID's to `typeName()`, the method returns the corresponding (input) names.

```
typeName(reh = edgelist_reh, typeID = c(0,2))
## [1] "competition" "cooperation"
```

actorID Returns the corresponding ID's given a set of actorName's.

```
actorID(reh = edgelist_reh, actorName =
→ c("Michaela", "Alexander", "Lexy"))
## [1] 16  0 12
```

typeID Returns the corresponding ID's given a set of typeName's.

```
typeID(reh = edgelist_reh, typeName = "cooperation")
## [1] 2
```

C.1.2 A function for transforming processed event histories into different formats

The function `rehshape()` aims to transform the structure of a relational event history object from a format to another one, e.g., from a `reh` structure to a

new structure suitable for other packages that also estimate tie or actor oriented models (and vice versa). At the current package version, the function `reshape()` can convert data used as input for the function `reLevent::rem()` (Butts, 2023) to an `reh` object and vice versa.

C.2 `remstimate` : optimization tools for relational event history data

The `remstimate` package (Arena, Lakdawala, et al., 2022) aims to provide optimization tools for the likelihood of both tie-oriented (REM, Butts (2008)) and actor-oriented (DyNAM, Stadtfeld and Block (2017)) modeling frameworks.

We discuss the input required by the main function `remstimate()` and the output of a `remstimate` object along with its attributes and methods. The content presented in this appendix refers to the latest version of the package which is the 2.0.0 .

Input arguments

The input arguments in `remstimate()` are:

- reh** A `reh` object, namely an output object from the processing function `remify::reh()` .
- stats** A `remstats` object (Meijerink-Bosman et al., 2023): when the input `model="tie"` , then `stats` is an array of statistics with dimensions $[M \times D \times P]$: where M is the number of events, D is the number of possible dyads (the largest risk set), P is the number of statistics; if the input `model="actor"` , then `stats` is a list of two arrays named `"rate"` and `"choice"` with dimensions $[M \times N \times P]$, where N is the number of actors (treated as senders in the array `"rate"` , as receivers in the array `"choice"` .
- method** The optimization method to use, which can be: maximum likelihood estimation (`"MLE"`), gradient descent optimization algorithm with optimized adaptive movement estimation (`"GDADAMAX"`), Bayesian Sampling Importance Resampling (`"BSIR"`), Hamiltonian Monte Carlo (`"HMC"`).
- ncores** The number of threads for the parallelization (default is 1, namely, no parallelization).
- prior** A prior distribution over the model parameters when using Bayesian methods (`"BSIR"` , or `"HMC"`).
- nsim** If `method = "HMC"` , then `nsim` is the number of simulations (iterations) in each chain. If `method = "BSIR"` , then `nsim` is the number of samples from the proposal distribution.

- nchains** The number of chains to generate (used for the method `"HMC"`).
- burnin** The number of initial iterations to be added as burn-in (used for the method `"HMC"`).
- thin** The number of steps to skip in the posterior draws of the HMC (used for the method `"HMC"`).
- init** A vector of initial values if `model = "tie"` , or a list of two vectors, named `"rate"` and `'choice'` if `model = "actor"` . This argument is used for the methods `"GDADAMAX"` and `"HMC"` .
- epochs** The number of iterations used in the methods `"GDADAMAX"` .
- epsilon** The inter-iteration difference of the loss function used in the method `"GDADAMAX"` and it is used as stop-rule within the algorithm.
- seed** Seed for the reproducibility of results based on the Bayesian optimization methods (`"BSIR"` and `"HMC"`).
- silent** A `logical` value, if `silent = FALSE` the progress of optimization status will be printed out.

Structure of the output object

The output of the function `remstimate()` is an S3 object of class `remstimate` and for any optimization method it contains a list of objects such as:

- coefficients** A `numeric` vector of estimates of the model parameters. Each element of the vector is named after the variable or interaction name.
- loglik** The value that the log-likelihood function assumes at the values of the estimates (`coefficients`). By default, The optimization methods minimize the negative log-likelihood ($-\ln \mathcal{L}$) but the `loglik` value return is its opposite. In the Bayesian optimization methods it is the value that the posterior log-likelihood assumes at the posterior modes.
- gradient** The value of the gradient vector calculated at the values of the estimates (`coefficients`).
- hessian** The values of the Hessian matrix calculated at the values of the estimates (`coefficients`).
- vcov** The estimated variance and covariance matrix of the parameters (in the case of Bayesian methods like `"BSIR"` and `"HMC"` it is estimated from the posterior distribution).
- se** The estimated standard errors of the estimates (in the case

	of Bayesian methods like <code>"BSIR"</code> and <code>"HMC"</code> it is estimated from the posterior distribution).
residual.deviance	The value of the residual deviance, that is $-2 \ln \mathcal{L}$ where $\ln \mathcal{L}$ is the value of the log-likelihood function calculated at the values of the estimates (namely, the output <code>loglik</code>).
null.deviance	The null deviance of the relational event model explains how much well the model fits and predict the data with only the intercept specified in the linear predictor.
model.deviance	The model deviance quantifies the distance in terms of deviance between the null model and the model of interest specified by the user and it is calculated as <code>null.deviance - residual.deviance</code> .
df.null	The degrees of freedom for the null model.
df.model	The degrees of freedom for the model of interest.
AIC	The Akaike's Information Criterion (AIC) measuring the goodness of fit of the model.
AICC	The small sample corrected AIC.
BIC	The Bayesian Information Criterion (BIC) measuring the goodness of fit of the model.
converged	A <code>logical</code> value indicating whether the optimization converged (<code>TRUE</code>) or not (<code>FALSE</code>).
iterations	The number of iterations of the optimization algorithm before convergence.

If the optimization method is Bayesian (`"BSIR"` or `"HMC"`), the output object contains the following additional information:

draws	The matrix of draws from the posterior distribution of the parameters.
post.mode	Vector of posterior modes estimated from the posterior distribution of the parameters.
post.mean	Vector of posterior means estimated from the posterior distribution of the parameters.
log_posterior	Vector of values of the logarithmic posterior density calculated at the values of the posterior draws.

Attributes

The attributes available for a `remestimate` object can be listed via the following function

```
names(attributes(remestimate_object))
```

When a `remestimate` object is created, the attributes available are:

- formula** The `formula` object specified as the linear predictor and provided by the input `stats` .
- model** A `character` value indicating the modeling framework used (whether it was tie-oriented, `"tie"` , or actor-oriented, `"actor"`).
- ordinal** A `logical` value indicating the type of likelihood function used whether it is for ordinal (`TRUE`) or continuous time modeling (`FALSE`).
- method** A `character` value indicating the optimization method used ("MLE", "GDADAMAX", "BSIR" or "HMC").
- approach** A `character` value indicating the approach to which the optimization method corresponds (for `"MLE"` and `"GDADAMAX"` the value of the attribute `"approach"` is `"Frequentist"` , for `"BSIR"` and `"HMC"` the value of the attribute `"approach"` is `"Bayesian"`).
- statistics** A `character vector` of names of the statistics specified in the model.

Depending on the optimization method used, further attributes are present. For instance, if `method = "GDADAMAX"` , then the output object will also contain information about the number of `"epochs"` and the value of the parameter `"epsilon"` . If the method is a Bayesian approach then the output will contain the attribute `"prior"` that returns the prior specified as input and the attribute `"nsim"` that is the number of simulations (draws). Furthermore, if `method = "HMC"` then the output will also contain information about the number of chains generated (attribute `"nchains"`), the parameters `"burnin"` and `"thin"` .

Methods

The methods available for a `remestimate` object are:

- print** Short summary containing a few lines such as the estimated parameters, names of statistics, modeling framework and the chosen optimization method
- summary** An extended summary with more detailed information about the model.
- aic** Returns the Akaike's Information Criterion (AIC).
- aicc** Returns the small sample corrected AIC.
- bic** Returns Bayesian Information Criterion (BIC).

Other methods that are under development and will be available in future versions of `remestimate` are:

- predict** This method generates a set of predictions under specific requirements that are provided as input arguments in the method function (for instance, how many time points a-head to predict).
- plot** The plot method will return several plots as to diagnostics on the estimated model.
- waic** This method returns the Watanabe-Akaike's Information Criterion that measure the predictive performance of a model and can be used for model comparison.

C.3 `bremory` : modeling the influence of past social interactions in relational event networks

The methods presented in this dissertation are available in the `bremory` package (Arena, 2022a). We summarily discuss the main functions of the package.

The functions `smm()` and `bma()` perform the Bayesian semi-parametric approach presented in Chapter 3, the function `pmm()` performs the parametric methods presented in Chapters 4 and 5.

`smm()` (Fitting step-wise memory models) A function that fits one or more K-step-wise models on the same specified linear predictor. The user can: (i) supply one or more predefined K-steps functions, (ii) make the function `smm()` generate a set of K-steps functions via the algorithm presented in Appendix A.3. The output is an S3 object of class `smm` and it either returns a simple summary of the estimates and standard errors about the only step-wise model estimated or, in the case of a set of two

or more K-step-wise models, it will have a more complex structure collecting all the necessary information from each fitted step-wise model which will be used for the Bayesian model averaging stage. The K-steps functions given as input or internally generated can differ in number of steps and maximum time length (γ_{\max}). The internal routines for the computation of endogenous statistics on time intervals are optimized and both computation and model estimation can be parallelized on multiple threads.

`bma()` (Performing **b**ayesian **m**odel **a**veraging) A function that performs the Bayesian Model Averaging on a set of K-step-wise models. The user must supply a `smm` object as input, containing the set of estimated step-wise models. There are two default weighting systems available for the averaging, which are `"BIC"` and `"WAIC"`. Furthermore, the user can set the number of draws from the posterior distribution and the size of the grid of γ 's where to approximate the posterior trend of the effects β .

`pmm()` (Fitting **p**arametric **m**emory **m**odels) A function that fits a parametric memory model. The parametric model can be either a relational event model with one memory parameter or a SentiREM. The available memory decay functions are `"exponential"`, `"linear"` and, `"one-step"`. The regression model available for the event sentiment is the Probit model. The estimation of the model parameters can be run either via the maximization of the profile log-likelihood or via the optimization of the log-likelihood based on trust region algorithms.

BIBLIOGRAPHY

- Aalbers, R. H., Dolfsma, W., & Leenders, R. T. (2016). Vertical and horizontal cross-ties: Benefits of cross-hierarchy and cross-unit ties for innovative projects. *Journal of Product Innovation Management*, 33(2), 141–153. <https://doi.org/https://doi.org/10.1111/jpim.12287>
- Agresti, A. (2015). *Foundations of linear and generalized linear models*. John Wiley.
- Amati, V., Lomi, A., & Mascia, D. (2019). Some days are better than others: Examining time-specific variation in the structuring of interorganizational relations. *Social Networks*, 57, 18–33. <https://doi.org/https://doi.org/10.1016/j.socnet.2018.10.001>
- Ancona, D. G., Goodman, P. S., Lawrence, B. S., & Tushman, M. L. (2001). Time: A new research lens. *Academy of Management Review*, 26(4), 645–663. <https://doi.org/10.5465/amr.2001.5393903>
- Arena, G. (2022a). *Bremory : Modeling memory retention in relational event data* [R package version 1.0.0]. <https://github.com/TilburgNetworkGroup/bremory>
- Arena, G. (2022b). *Remify: Processing and transforming reh to formats suitable for the remverse packages and more* [R package version 2.0.0]. <https://github.com/TilburgNetworkGroup/remify>
- Arena, G., Lakdawala, R., & Generoso Vieira, F. (2022). *Remstimate: Optimization tools for tie-oriented and actor-oriented relational event models* [R package version 2.0.0]. <https://github.com/TilburgNetworkGroup/remstimate>
- Arena, G., Mulder, J., & Leenders, R. (2022). A bayesian semi-parametric approach for modeling memory decay in dynamic social networks. *Sociological Methods & Research*. <https://doi.org/10.1177/00491241221113875>
- Arena, G., Mulder, J., & Leenders, R. T. A. (2023). How fast do we forget our past social interactions? understanding memory retention with parametric

Bibliography

- decays in relational event models. *Network Science*, 11(2), 267–294. <https://doi.org/10.1017/nws.2023.5>
- Bianchi, F., & Lomi, A. (2022). From ties to events in the analysis of interorganizational exchange relations. *Organizational Research Methods*. <https://doi.org/10.1177/10944281211058469>
- Boschee, E., Lautenschlager, J., O'Brien, S., Shellman, S., Starz, J., & Ward, M. (2015). *ICEWS Coded Event Data*. <https://doi.org/10.7910/DVN/28075>
- Brandenberger, L. (2018a). *Rem: Relational event models* [R package version 1.3.1]. <https://CRAN.R-project.org/package=rem>
- Brandenberger, L. (2018b). Trading favors—examining the temporal dynamics of reciprocity in congressional collaborations using relational event models. *Social Networks*, 54, 238–253. <https://doi.org/https://doi.org/10.1016/j.socnet.2018.02.001>
- Brandes, U., Lerner, J., & Snijders, T. A. (2009). Networks evolving step by step: statistical analysis of dyadic event data. *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2009*, 200–205. <https://doi.org/10.1109/ASONAM.2009.28>
- Brass, D. J., & Labianca, G. (1999). Social capital, social liabilities, and social resources management. In R. T. A. J. Leenders & S. M. Gabbay (Eds.), *Corporate social capital and liability* (pp. 323–338). Springer US. https://doi.org/10.1007/978-1-4615-5027-3_18
- Butts, C. T. (2008). A relational event framework for social action. *Sociological Methodology*, 38(1), 155–200. <https://doi.org/https://doi.org/10.1111/j.1467-9531.2008.00203.x>
- Butts, C. T. (2021). *Relevant: Relational event models* [R package version 1.1]. <https://CRAN.R-project.org/package=relevant>
- Butts, C. T. (2023). *Relevant: Relational event models* [R package version 1.2-1]. <https://CRAN.R-project.org/package=relevant>
- Clauset, A., Newman, M. E. J., & Moore, C. (2004). Finding community structure in very large networks. *Phys. Rev. E*, 70, 066111. <https://doi.org/10.1103/PhysRevE.70.066111>
- Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2), 187–220. Retrieved January 31, 2023, from <http://www.jstor.org/stable/2985181>
- Cronin, M. A., Weingart, L. R., & Todorova, G. (2011). Dynamics in groups: Are we there yet? *Academy of Management Annals*, 5(1), 571–612. <https://doi.org/10.1080/19416520.2011.590297>

- Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695. <https://igraph.org>
- Fletcher, R. (1987). *Practical methods of optimization*. John Wiley.
- Fritz, C., Thurner, P. W., & Kauermann, G. (2021). Separable and semiparametric network-based counting processes applied to the international combat aircraft trades. *Network Science*, 9(3), 291–311. <https://doi.org/10.1017/nws.2021.9>
- Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., & Rubin, D. (2013). *Bayesian data analysis (3rd ed.)* Chapman; Hall/CRC. <https://doi.org/10.1201/b16018>
- Geyer, C. J. (2015). *Trust* [R package version 0.1-8]. <https://CRAN.R-project.org/package=trust>
- Grünwald, P., & van Ommen, T. (2017). Inconsistency of bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, 12(4), 1069–1103. <https://doi.org/10.1214/17-BA1085>
- Hajibagheri, A., Sukthankar, G., Lakkaraju, K., Alvares, H., Wigand, R. T., & Agarwal, N. (2018). Using massively multiplayer online game data to analyze the dynamics of social interactions. In K. Lakkaraju, G. Sukthankar, & R. T. Wigand (Eds.), *Social interactions in virtual worlds: An interdisciplinary perspective* (pp. 375–416). Cambridge University Press. <https://doi.org/10.1017/9781316422823.015>
- Hanneke, S., Fu, W., & Xing, E. P. (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4(none), 585–605. <https://doi.org/10.1214/09-EJS548>
- Karimova, D., Mulder, J., & Leenders, R. T. A. J. (2022). Separating the wheat from the chaff: Bayesian regularization in dynamic social networks. <https://doi.org/10.48550/ARXIV.2203.12474>
- Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430), 773–795. <https://doi.org/10.1080/01621459.1995.10476572>
- Kitts, J. A., Lomi, A., Mascia, D., Pallotti, F., & Quintane, E. (2017). Investigating the temporal dynamics of interorganizational exchange: Patient transfers among italian hospitals. *American Journal of Sociology*, 123(3), 850–910. <https://doi.org/10.1086/693704>
- Kossinets, G., & Watts, D. J. (2006). Empirical analysis of an evolving social network. *Science*, 311(5757), 88–90. Retrieved February 2, 2023, from <http://www.jstor.org/stable/3843310>

Bibliography

- Kozlowski, S. W. J., Chao, G. T., Grand, J. A., Braun, M. T., & Kuljanin, G. (2016). Capturing the multilevel dynamics of emergence: Computational modeling, simulation, and virtual experimentation. *Organizational Psychology Review*, 6(1), 3–33. <https://doi.org/10.1177/2041386614547955>
- Kratzer, J., Leenders, R. T., & Van Engelen, J. M. (2006). Managing creative team performance in virtual environments: An empirical study in 44 rd teams. *Technovation*, 26(1), 42–49. <https://doi.org/https://doi.org/10.1016/j.technovation.2004.07.016>
- Krivitsky, P. N., & Handcock, M. S. (2022). *Tergm: Fit, simulate and diagnose models for network evolution based on exponential-family random graph models* [R package version 4.1.1]. The Statnet Project (<https://statnet.org>). <https://CRAN.R-project.org/package=tergm>
- Labianca, G., & Brass, D. J. (2006). Exploring the social ledger: Negative relationships and negative asymmetry in social networks in organizations. *Academy of Management Review*, 31(3), 596–614. <https://doi.org/10.5465/amr.2006.21318920>
- Lawless, J. F. (2002). *Statistical models and methods for lifetime data*. John Wiley & Sons. <https://doi.org/10.1002/9781118033005>
- Leenders, R. T. A. J., & Dolfsma, W. A. (2016). Social networks for innovation and new product development. *Journal of Product Innovation Management*, 33(2), 123–131. <https://doi.org/https://doi.org/10.1111/jpim.12292>
- Leenders, R. T. A. J., Contractor, N. S., & DeChurch, L. A. (2016). Once upon a time: Understanding team processes as relational event networks. *Organizational Psychology Review*, 6(1), 92–115. <https://doi.org/10.1177/2041386615578312>
- Leenders, R. T., van Engelen, J. M., & Kratzer, J. (2003). Virtuality, communication, and new product team creativity: A social network perspective [Special Issue on Research Issues in Knowledge Management and Virtual Collaboration in New Product Development]. *Journal of Engineering and Technology Management*, 20(1), 69–92. [https://doi.org/https://doi.org/10.1016/S0923-4748\(03\)00005-5](https://doi.org/https://doi.org/10.1016/S0923-4748(03)00005-5)
- Leifeld, P., & Cranmer, S. J. (2019). A theoretical and empirical comparison of the temporal exponential random graph model and the stochastic actor-oriented model. *Network Science*, 7(1), 20–51. <https://doi.org/10.1017/nws.2018.26>
- Leifeld, P., Cranmer, S. J., & Desmarais, B. A. (2018). Temporal exponential random graph models with btergm: Estimation and bootstrap confidence

- intervals. *Journal of Statistical Software*, 83(6), 1–36. <https://doi.org/10.18637/jss.v083.i06>
- Lerner, J., Bussman, M., Snijders, T. A. B., & Brandes, U. (2013). Modeling frequency and type of interaction in event networks. *CORVINUS JOURNAL OF SOCIOLOGY AND SOCIAL POLICY*, 4, 3–32. <https://doi.org/10.14267/cjssp.2013.01.01>
- Marcum, C. S., & Butts, C. T. (2015). Constructing and modifying sequence statistics for relevant using informr in r. *Journal of Statistical Software*, 64(5). <https://doi.org/10.18637/jss.v064.i05>
- Meijerink-Bosman, M., Arena, G., Karimova, D., Lakdawala, R., Shafiee Kamalabad, M., & Generoso Vieira, F. (2023). *Remstats: Computes statistics for relational event history data* [R package version 3.0.1]. <https://github.com/TilburgNetworkGroup/remstats>
- Meijerink-Bosman, M., Back, M., Geukes, K., Leenders, R., & Mulder, J. (2022). Discovering trends of social interaction behavior over time: An introduction to relational event modeling. *Behavior Research Methods*. <https://doi.org/10.3758/s13428-022-01821-8>
- Meijerink-Bosman, M., Leenders, R., & Mulder, J. (2022). Dynamic relational event modeling: Testing, exploring, and applying. *PLOS ONE*, 17(8), 1–23. <https://doi.org/10.1371/journal.pone.0272309>
- Mitchell, T. R., & James, L. R. (2001). Building better theory: Time and the specification of when things happen. *Academy of Management Review*, 26(4), 530–547. <https://doi.org/10.5465/amr.2001.5393889>
- Moerbeek, H. H. S., & Need, A. (2003). Enemies at work: Can they hinder your career? *Social Networks*, 25(1), 67–82. [https://doi.org/10.1016/S0378-8733\(02\)00037-0](https://doi.org/10.1016/S0378-8733(02)00037-0)
- Monge, P. R. (1990). Theoretical and analytical issues in studying organizational processes. *Organization Science*, 1(4), 406–430. <https://doi.org/10.2307/2634972>
- Mulder, J., et al. (2020). *Remverse*.
- Mulder, J., & Leenders, R. T. A. (2019). Modeling the evolution of interaction behavior in social networks : A dynamic relational event approach for real-time analysis. *Chaos, Solitons and Fractals: the interdisciplinary journal of Nonlinear Science, and Nonequilibrium and Complex Phenomena*, 119, 73–85. <https://doi.org/10.1016/j.chaos.2018.11.027>
- Mulder, J., Williams, D. R., Gu, X., Tomarken, A., Böing-Messing, F., Olsson-Collentine, A., Meijerink, M., Menke, J., van Aert, R., Fox, J.-P., Hoijtink, H., Rosseel, Y.,

Bibliography

- Wagenmakers, E.-J., & van Lissa, C. (2021). BFpack: Flexible bayes factor testing of scientific theories in R. *Journal of Statistical Software*, 100(18), 1–63. <https://doi.org/10.18637/jss.v100.i18>
- Nocedal, J., & Wright, S. (2006). *Numerical optimization*. John Wiley.
- Offer, S. (2021). Negative social ties: Prevalence and consequences. *Annual Review of Sociology*, 47(1), 177–196. <https://doi.org/10.1146/annurev-soc-090820-025827>
- Panzarasa, P., Opsahl, T., & Carley, K. M. (2009). Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology*, 60(5), 911–932. <https://doi.org/https://doi.org/10.1002/asi.21015>
- Patison, K. P., Quintane, E., Swain, D. L., Robins, G., & Pattison, P. (2015). Time is of the essence: An application of a relational event model for animal social networks. *Behavioral Ecology and Sociobiology*, 69(5), 841–855. <https://doi.org/10.1007/s00265-015-1883-3>
- Perry, P. O., & Wolfe, P. J. (2013). Point process modelling for directed interaction networks. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 75(5), 821–849. <https://doi.org/10.1111/rssb.12013>
- Pilny, A., Schechter, A., Poole, M. S., & Contractor, N. (2016). An illustration of the relational event model to analyze group interaction processes. *Group Dynamics: Theory, Research, and Practice*, 20(3), 181–195. <https://doi.org/10.1037/gdn0000042>
- Quintane, E., & Carnabuci, G. (2016). How do brokers broker? tertius gaudens, tertius iungens, and the temporality of structural holes. *Organization Science*, 27(6), 1343–1360. <https://doi.org/10.1287/orsc.2016.1091>
- Quintane, E., Pattison, P. E., Robins, G. L., & Mol, J. M. (2013). Short- and long-term stability in organizational networks: Temporal structures of project teams. *Social Networks*, 35(4), 528–540. <https://doi.org/10.1016/j.socnet.2013.07.001>
- R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. <https://www.R-project.org/>
- Raftery, A. E. (1995). Bayesian model selection in social research. *Sociological Methodology*, 25, 111–163. <https://doi.org/10.2307/271063>
- Rivera, M. T., Soderstrom, S. B., & Uzzi, B. (2010). Dynamics of dyads in social networks: Assortative, relational, and proximity mechanisms. *Annual Re-*

- view of Sociology*, 36(1), 91–115. <https://doi.org/10.1146/annurev.soc.34.040507.134743>
- Sapiezynski, P., Stopczynski, A., Lassen, D. D., & Lehmann, S. (2019). Interaction data from the Copenhagen Networks Study. *Scientific Data*, 6(1), 1–10. <https://doi.org/10.1038/s41597-019-0325-x>
- Schechter, A., Pilny, A., Leung, A., Poole, M. S., & Contractor, N. (2018a). Step by step: Capturing the dynamics of work team process through relational event sequences. *Journal of Organizational Behavior*, 39(9), 1163–1181. <https://doi.org/https://doi.org/10.1002/job.2247>
- Schechter, A., Pilny, A., Leung, A., Poole, M. S., & Contractor, N. (2018b). Step by step: Capturing the dynamics of work team process through relational event sequences. *Journal of Organizational Behavior*, 39(9), 1163–1181. <https://doi.org/10.1002/job.2247>
- Schoenfeld, D. (1982). Partial residuals for the proportional hazards regression model. *Biometrika*, 69(1), 239–241. <http://www.jstor.org/stable/2335876>
- Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2), 461–464. <https://doi.org/10.1214/aos/1176344136>
- Snijders, T. A. B. (2017a). Siena: Statistical modeling of longitudinal network data. In R. Alhajj & J. Rokne (Eds.), *Encyclopedia of social network analysis and mining* (pp. 1–9). Springer New York. https://doi.org/10.1007/978-1-4614-7163-9_312-1
- Snijders, T. A. B. (2017b). Stochastic actor-oriented models for network dynamics. *Annual Review of Statistics and Its Application*, 4(1), 343–363. <https://doi.org/10.1146/annurev-statistics-060116-054035>
- Stadtfield, C., & Block, P. (2017). Interactions, actors, and time: Dynamic network actor models for relational events. *Sociological Science*, 4(14), 318–352. <https://doi.org/10.15195/v4.a14>
- Stadtfield, C., & Geyer-Schulz, A. (2011). Analyzing event stream dynamics in two-mode networks: An exploratory analysis of private communication in a question and answer community. *Social Networks*, 33(4), 258–272. <https://doi.org/https://doi.org/10.1016/j.socnet.2011.07.004>
- Stadtfield, C., & Hollway, J. (2020). *Goldfish: Goldfish – statistical network models for dynamic network data* [R package version 1.4.8]. www.social-networks.ethz.ch/research/goldfish.html
- Therneau, T. M. (2022). *A package for survival analysis in r* [R package version 3.4-0]. <https://CRAN.R-project.org/package=survival>

Bibliography

- Tranmer, M., Marcum, C. S., Morton, F. B., Croft, D. P., & de Kort, S. R. (2015). Using the relational event model (rem) to investigate the temporal dynamics of animal social networks. *Animal Behaviour*, *101*, 99–105. <https://doi.org/https://doi.org/10.1016/j.anbehav.2014.12.005>
- Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, *27*(5), 1413–1432. <https://doi.org/10.1007/s11222-016-9696-4>
- Volinsky, C. T., Raftery, A. E., Madigan, D., & Hoeting, J. A. (1999). David Draper and E. I. George, and a rejoinder by the authors. *Statistical Science*, *14*(4), 382–417. <https://doi.org/10.1214/ss/1009212519>
- Watanabe, S. (2013). A widely applicable bayesian information criterion. *Journal of Machine Learning Research*, *14*(1), 867–897.
- Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2018). Using Stacking to Average Bayesian Predictive Distributions (with Discussion). *Bayesian Analysis*, *13*(3), 917–1007. <https://doi.org/10.1214/17-ba1091>

SUMMARY

A relational event network consists of a time-ordered sequence of social interactions (relational events) in which we know the time of occurrence for each event and the actors involved in the interaction. Such network data is increasingly available and can be observed in a multitude of scenarios, for instance: in-person interactions among citizens of a rural area, email communication among employees in a firm, text messages sent among university students, socio-political interactions occurring among countries, interactions among players in an online strategy game, to mention a few.

Past interactions with peers, friends, colleagues, and more can affect our decisions about whom to interact with in the imminent future and how. What happened in the past can influence our future interactions to a certain extent depending on how recently past events occurred, the sentiment of such events, the form of interaction (e.g., be it an in-person or a digital one) and possibly other factors. It comes naturally to attribute large weights to recently occurred events and to weigh less those events that happened long ago and of which our perception may have faded. However, there can be further differences in the speed and shape of the memory decay depending on the sentiment of the past event that we consider. For instance, If we had an argument with a friend and this happened recently, we may attribute to such an event a weight larger than the weight of a positive event, be it a compliment or any other positive social interaction with that friend. In other words, memory decay describes how fast we forget about past relational events, and its study is crucial to understand better the impact of actor-level or group-level network dynamics in a network of social interactions.

Summary

Researchers have devoted marginal focus to developing methods for the analysis of memory decay in relational event networks. Most studies assume an equal weight across all the past events, making recently occurred events weigh as much as long-passed ones on the actor's decision-making process. Other studies consider a short-run and a long-run definition of network dynamics to compare the effects between dynamics based on recent events and dynamics based on long-passed events. Alternatively, some studies calculate network dynamics in multiple time intervals to investigate the presence of a possible trend of the effects of the network dynamics from the most recent to the less recent events in a step-function fashion. Furthermore, some studies introduce the use of a parametric form of memory decay, where the weight of past events follows an exponential decay, and its speed is defined by a memory parameter that researchers usually fix a priori to a specific value. Even though researchers have been conducting a theory-based choice of the time intervals or the memory parameter, such an approach may be limiting, potentially hindering aspects of the data that could instead present insights on social phenomena and their dynamics, supporting new theory development. However, their contributions have been helpful and have stimulated the development of time-sensitive model extensions around the relational event network modeling framework.

In this dissertation, we introduce time-sensitive methods for modeling the influence of past social interactions in relational event networks. First, we present a real case study in which we focus on analysing a sequence of emails about innovation sent among employees in a company. We introduce norms of reciprocity and inertia defined in short-run and long-run forms in order to understand whether and how they explain the email rate of employees along with other variables available with the data (Chapter 2). Then, we introduce a semi-parametric method for estimating the shape of memory decay by applying a Bayesian model averaging approach to a "bag" of step-wise relational event models where network dynamics are calculated on multiple time intervals (Chapter 3). Next, we propose a method that assumes a parametric memory decay, provides a way to estimate such a memory parameter from the observed sequence of events and a Bayesian test to establish which decay function fits the data best (Chapter 4). Finally, we present the SentiREM for modeling the event rate of the next dyadic event separately from the probability of the next event type (sentiment). We focus on the scenario where we observe two event types in the network and model the next event sentiment via Probit regres-

sion. We provide a sentiment-based definition of the network dynamics, introduce the methodology for estimating the sentiment-based memory parameters and some Bayesian tests for both the memory parameters and the effects of sentiment-based network dynamics (Chapter 5).

The methodology presented in this dissertation is made available in the R package *bremory* (Appendix C).

ACKNOWLEDGEMENTS

First of all, I would like to thank my two supervisors, Joris and Roger, for giving me the opportunity to start my PhD. The memory of the first (online) interview is still vivid in my mind. I was enthusiastic about the transformative and enriching journey that I was going to embark on a few months later. I am immensely grateful to both of you for your continuous support and guidance throughout my PhD journey. Thanks for the insightful meetings. They have always been very professional, calm, and structured, but also with a good dose of laughs and funny chats. I enjoyed the intermezzo chats about music, music bands, Eurovision, and Dutch and Italian customs. In every meeting, your dedication and expertise were always inspiring and shaped my professional growth and research. You both gave me constructive feedback and showed belief in my skills. Your support, patience, and encouragement have also been a source of motivation during more challenging stages of this journey. Thank you both for helping me articulate my research effectively. Your mentoring and suggestions have made me grow as a researcher. I am incredibly fortunate to have had the chance to pursue my PhD trajectory under your mentorship.

I want to extend my sincere gratitude to each committee member for dedicating their time and valuable expertise to read and evaluate my thesis. Thomas Snijders, Jeroen Vermunt, Eric-Jan Wagenmakers and Laurence Brandenberger, it is truly an honor that you are part of my committee. Thank you for providing suggestions and constructive comments on my dissertation.

Next, I want to thank my colleagues in the research group I am part of. Di-ana, Marlyne, Mahdi, Rumana and Fábio, I am grateful for your support and for making this experience great professionally and personally. Thank you for your

Acknowledgements

feedback, constructive dialogues, laughs, and lovely memories we shared during these years. Diana and Marlyne, thank you for giving me a warm welcome to the research group when I started in October 2018. I can never forget the weekly lunches at JADS, the chats in Italian with Diana (they always made me feel at home), and the engaging conversations about the Dutch language and culture with Marlyne (which made me feel like in my second home, the Netherlands). Marlyne, thank you for your availability, professionalism, and expertise. Every time we have a meeting, we are always on the same wave, always understanding each other. I am grateful for working next to you, also at the latest stage of software development in which we are involved. Mahdi, thank you for your support and suggestions when I was still at the beginning of this experience. Rumana and Fábio, thank you for being trustful and excellent companions during these years. Thank you both for your efforts and dedication while working together to develop part of the software. Rumana, thank you for your understanding and empathy. I incredibly enjoyed the talks, laughs and experiences we shared. Thanks for being not just a colleague that gives me support and encouragement but also a fantastic friend with which I look forward to creating more memories. Fábio, thank you for the (long) talks at the office during lunch break and for the funny conversations that we often shared with Rumana too. I will never forget the funny and less funny moments spent with you, Rumana and Marlyne in Australia. It was an odyssey of fantastic and unfortunate events that I will remember. For instance, I won't ever forget that I could only receive my luggage back almost three weeks after we left for Australia. Still, I will never forget moments like when we went on the Skyrail and could admire the rainforest of Kuranda from above hanging in the air (that was a bit scary, I must say) when we visited the Aquarium and reacted to any animal that we had never seen until then. I truly cherish all the memories I have collected with you. All the moments I spent with you have made this journey memorable, helped, and kept me motivated.

I want to thank further Rumana and Marlyne, which are my paranymphs. Thank you both for your feedback and support during the preparation of my defense.

Moreover, I want to thank Erwin, Mihai, Edoardo, Angelica, and Giovanni. I feel lucky to have met you since my PhD began. That period was such a significant moment in my life. Everything was new to me, the country, the culture, the Dutch language, speaking a second language and reshaping my way of for-

mulating thoughts and sentences from Italian to English, meeting new people and making new friends. All these changes seemed manageable since I quickly got along and shared beautiful moments with you. Erwin, Mihai, and Edoardo, thank you for all the moments we shared during these years, thanks for the game nights, and for being not just colleagues but also friends (other than neighbors). I knew I could count on you whenever I needed something, and I never felt alone. Mihai, thanks for being a supportive office mate, a trustful friend and a good companion when the pandemic hit. I always remained delighted by our talks about technology, statistics and science behind a bowl of yoghurt at breakfast or at the office. Angelica and Giovanni, thank you for your warmth and kindness. I always felt welcome whenever I had the chance to share a dinner or lunch with you. If I want to feel like in Italy here in the Netherlands, you are the first two people that come to my mind, then Edoardo (:~P).

During the last year of my PhD, I was working on the last project, and like a bolt from the blue, I met my twinning spirit, Rick. Rick, since then, you have never left my side. Thank you for your patience and understanding during the most challenging moments of my PhD. I consider myself lucky to have found such a kind and genuine soul as you are. Thanks for your support and sincerity and for sharing your passion for board games with me. Thanks for all the love and affection you show me, the moments we have shared so far, and more to come.

Un ringraziamento speciale va ai miei genitori, Pina e Michele, i quali mi hanno sempre sostenuto in qualsiasi mia scelta da ben 29 anni. Mamma, i tuoi consigli e le tue parole mi hanno sempre dato forza e coraggio anche nei momenti più difficili e delicati. Il tuo affetto e la tua presenza costante non mi hanno mai fatto percepire la distanza che ci separa. Grazie di essere stata orecchio per le mie esperienze e di avermi portato alla riflessione quando ne avevo bisogno. Papà, grazie per avermi supportato in ogni mia scelta sul mio percorso universitario: dal trasferimento a Palermo, a quello a Padova per poi finire, ancora più lontano, a Tilburg. Le tue passioni (hobby) e la dedizione che mostri verso di esse sono un esempio che non dimenticherò mai. Grazie a entrambi per esserci stati ogni volta che ne avevo bisogno. Io sono grato di rendervi felici e orgogliosi, nonostante i sacrifici e la lontananza. Vi voglio bene.

Voglio ringraziare anche le mie due sorelle, Federica e Francesca, e i miei due cognati, Pietro e Massimiliano per il loro supporto durante questi anni. Federica e Francesca, grazie per il sostegno, per le lunghe conversazioni, per i consigli, e

Acknowledgements

per tenermi sempre aggiornato sui miei nipotini che mi mancano tanto. Grazie ai miei tre nipoti, Michele, Salvo e Alessandra ai quali dedico la mia tesi. L'amore che comunicate riesce sempre a raggiungermi sino a Tilburg, nonostante la distanza che ci separa. *Nicuzzi miei*, vi voglio un mondo di bene. Grazie alla mia famiglia intera, perché ogni volta che ritorno in Sicilia, casa si riempie ed è uno dei ricordi che mi rende sempre nostalgico.

Last but not least, voglio ringraziare Francesca (detta *La Francy*), Orazio e Jacopo (detto *Jacopew*). Ritornando al tema "*grandi distanze ci separano but always present*", questi tre personaggi non mi hanno mai mollato un secondo durante questi anni. Grazie per le interminabili conversazioni, per essere stati una spalla e un orecchio quando ne avevo bisogno.

Giuseppe Arena

Tilburg, 27th July 2023

