

Latent Variable Models for Events on Social Networks

Owen G. Ward

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2022

© 2022

Owen G. Ward

All Rights Reserved

Abstract

Latent Variable Models for Events on Social Networks

Owen G. Ward

Network data, particularly social network data, is widely collected in the context of interactions between users of online platforms, but it can also be observed directly, such as in the context of behaviours of animals in a group living environment. Such network data can reveal important insights into the latent structure present among the nodes of a network, such as the presence of a social hierarchy or of communities. This is generally done through the use of a latent variable model. Existing network models which are commonly used for such data often aggregate the dynamic events which occur, reducing complex dynamic events (such as the times of messages on a social network website) to a binary variable. Methods which can incorporate the continuous time component of these interactions therefore offer the potential to better describe the latent structure present.

Using observed interactions between mice, we take advantage of the observed interactions' timestamps, proposing a series of network point process models with latent ranks. We carefully design these models to incorporate important theories on animal behaviour that account for dynamic patterns observed in the interaction data, including the winner effect, bursting and pair-flip phenomena. Through iteratively constructing and evaluating these models we arrive at the final cohort Markov-Modulated Hawkes process (C-MMHP), which best characterizes all aforementioned patterns observed in interaction data. The generative nature of our model provides evidence

for hypothesised phenomena and allows for additional insights compared to existing aggregate methods, while the probabilistic nature allows us to estimate the uncertainty in our ranking. In particular, our model is able to provide insights into the distribution of power within the hierarchy which forms and the strength of the established hierarchy. We compare all models using simulated and real data. Using statistically developed diagnostic perspectives, we demonstrate that the C-MMHP model outperforms other methods, capturing relevant latent ranking structures that lead to meaningful predictions for real data.

While such network models can lead to important insights, there are inherent computational challenges for fitting network models, particularly as the number of nodes in the network grows. This is exacerbated when considering events between each pair of nodes. As such, new computational tools are required to fit network point process models to the large social networks commonly observed. We consider online variational inference for one such model. We derive a natural online variational inference procedure for this event data on networks. Using simulations, we show that this online learning procedure can accurately recover the true network structure. We demonstrate using real data that we can accurately predict future interactions by learning the network structure in this online fashion, obtaining comparable performance to more expensive batch methods.

Table of Contents

Chapter 1: Introduction	1
Chapter 2: Background	5
2.1 Network Data	5
2.2 Statistical Models for Network Data	6
2.3 Inference for Network Data	8
2.4 Network Data across Time	12
2.5 Models for Dynamic Networks	13
2.6 Point Processes and Models for Network Event Data	15
2.7 Social Dynamics on Networks	20
Chapter 3: Latent Ranking Models for Social Animal Interactions	21
3.1 Introduction	21
3.2 Background	23
3.2.1 Issues with conventional approaches	26
3.3 Latent ranking structured network point process models	27
3.3.1 Latent ranking structured models for network point processes	28
3.3.2 Model inference	38
3.4 Results	39

3.4.1	Comparison Models	39
3.4.2	Model implementation	41
3.4.3	Synthetic results	41
3.4.4	Real data results	42
3.5	Discussion	51
Chapter 4: Online Inference for Community Detection using Events on Networks		56
4.1	Introduction	56
4.2	Online Streaming Data and Latent Cluster Assignment	59
4.3	An Online Learning Framework for Event Streams	65
4.3.1	Online Learning Algorithms for Network Point Processes	66
4.3.2	Approximation via Variational Inference	69
4.4	Simulation Studies	72
4.5	Real Data Analysis	82
4.6	Conclusion	84
Chapter 5: Discussion and Future Directions		86
References		89
Appendix A: Additional Material for Chapter 4		97
A.1	Algorithms	97
A.2	Additional Simulations	97

List of Tables

4.1	Median RMSE of predicted event counts vs true event counts in held out test set across 50 simulations. Online/Non-online estimates.	84
4.2	Median computation time for Online/Full Model fitting (seconds) across 50 simulations.	84
A.1	The Data Structure for Storing Historic Events. The upper diagram shows the structure under the Poisson model, where the key is the pair of nodes and the value is the corresponding cumulative number of all past events. The bottom diagram shows the structure under the Hawkes model, where the key is still the nodes and the value is the corresponding time sequence between $t_{current} - R$ and $t_{current}$ stored in queue structure.	98

List of Figures

2.1	The output of a two-dimensional latent space model fit to the dolphin interaction data of Lusseau (2003).	9
3.1	An example of a <i>win/loss</i> matrix and the corresponding reordered matrix according to the I&SI method. The entries shaded in red in the matrix are the <i>inconsistencies</i> , where the lower-ranked individual wins more frequently than the higher-ranked individual.	25
3.2	(a) Contour plot for $\alpha^{i,j} := g_\eta(f_i, f_j)$ where $f_i, f_j \in [0, 1]$. (b) Matrix of K-S statistics after fitting the C-HP model to the real data (reordered by I&SI ranking). The rows and columns of this matrix correspond to senders and receivers of an agonistic behaviour, respectively. Color shading reflects the values of the K-S test statistics. Red lines are empirical cumulative distribution functions of <i>rescaled-inter-event</i> times and black lines are cumulative distribution functions of exponential random variable with rate 1.	32
3.3	(a) Matrix of baseline rates $\lambda_1^{i,j}$ (reordered by I& SI rankings). These degree-corrected baseline rates allow for a more flexible node level model, clearly seen in the top row (a mouse which is involved in starting a large number of fights) and the bottom row (a mouse which does not start any fights but is often fought). (b) Pearson residuals for the C-HP and C-DCHP models.	34
3.4	Simulation results from simulating 50 interaction datasets with common C-MMHP parameters. (a) Shows posterior inference of latent rank variable $f_i, i = 1, \dots, 10$ by C-HP, C-DCHP and C-MMHP. Each value is the posterior mean for f_i inferred from 50 independent simulations from a C-MMHP model with the same underlying parameters, with the true rank values overlaid in red. (b) Show the inferred intensity for one pair of individuals (top ranked to second highest ranked) in one simulation using three models. Here we fit each of the C-HP, C-DCHP and C-MMHP models to this data and plot the inferred intensity function for this pair. The events and the state they occurred in, along with the times the process changed state, are also shown in this plot. The red/blue shaded area underneath shows the magnitude of the error in the estimation of the intensity in terms of overestimation and underestimation.	42

3.5	The Spearman rank correlation between the inferred ranking from each of these models, along with existing methods, and the known true ranking.	43
3.6	Real data fitting results. (a) Comparison of rank inference using different model with I&SI rank for Cohort 5. (b) Comparison of rank inference using different model with I&SI rank for Cohort 3.	46
3.7	Prediction of events and rank. (a) shows the MAE of predicted error for all cohorts, using the median predicted count for each model for each cohort on each day. (b) Summarizes the Spearman rank correlation of predicted rank for all cohorts, where each cohort is predicted by the posterior mean of $\hat{\lambda}_i(t^{(d)})$	48
3.8	Further results on C-MMHP. (a) Glicko score ranking prediction of last three days using posterior draws, after fitting the first 18 days data in C-MMHP. True Glicko score ranking of all the time period is shown with the solid colored line, while the posterior prediction mean is in dashed line and one standard deviation is plotted in shaded color. The x-axis corresponds to the total number of interactions across the cohort. (b) Rank correlation between DSNL inferred latent rank and Glicko score ranking for each day in one cohort. Three colored bar indicated the performance of three inferred rankings conducted on the overall interactions, active and inactive respectively.	50
3.9	Boxplot of the posterior draws for the latent rank of animals in one cohort, fitting the C-MMHP model to both the first 14 days and last 14 days of observations separately. The mice are ordered by the rank from the first 14 days.	54
4.1	Community recovery in terms of Adjusted Rand Index (ARI) for events simulated from a point process block model. Aggregate methods which look at the overall count between nodes, or bin the data (PZ)(Pensky and Zhang, 2019) cannot recover the community structure. This is the case regardless of how we aggregate the data to form the PZ estimator (as given by the multiple box-plots). Modelling the exact event times through a point process can recover the true community structure well.	58
4.2	Overall community recovery for our online inference procedure as we increase the number of nodes in the network, with each box-plot corresponding to 50 simulations. For $n = 100$ community recovery is challenging, however as the number of nodes increases we can correctly recover the community structure in almost all simulations.	74
4.3	Demonstrating the ARI of the estimated clustering as we observe events in term for varying network size. Here we show the mean ARI with error bars corresponding to one standard deviation. We see that for all network sizes, we can identify community structure, with the larger networks doing so quickly and with less variability.	75

4.4	ARI as we consider network structure with an increasing number of communities. As we increase the number of communities in the network we remain able to recover the community structure, with increasing uncertainty as the number of communities increases.	76
4.5	Computed ELBO based on the full data, using our online estimates. We see that the ELBO from the online estimates quickly converges to the optimum found by the corresponding batch estimate.	77
4.6	The full ELBO plotted against the percent of total events used to obtain the corresponding parameter estimates. We see that the online procedure obtains good estimates of the full ELBO using significantly fewer events.	78
4.7	The normalized ELBO and the full ELBO for simulated data, suitably scaled. Empirically, these quantities converge similarly over time, indicating this normalized may be a suitable measure of convergence.	79
4.8	Normalized ELBO for a range of network sizes. This metric quickly converges for all simulations.	80
4.9	Average difference in estimated and true rate matrix across time across multiple simulations. We see that these converge quickly, particularly as events are observed for a longer period.	81
4.10	Smoothed regret estimates as a function of time for a fixed network structure observed for varying lengths of time.	82
4.11	Average cumulative loss for our online inference procedure, with each line denoting one run of the procedure. The red horizontal line is the best average cumulative loss in hindsight. We see that as T increases, our online procedure comes close to this best loss.	83
A.1	Community recovery under Poisson simulated data for varying window size.	99
A.2	Community recovery under the Hawkes model, in terms of ARI, as we increase the number of nodes. Here we simulate events from the same underlying network structure each time.	100
A.3	Community recovery for the Hawkes model as we vary the number of communities.	101
A.4	Online community recovery under the Hawkes model as the number of nodes increases, for a fixed observation period.	101

A.5 Parameter Recovery for block Hawkes model as the observation time increases for a fixed network size. 103

Acknowledgements

There are many people who I am fortunate to be able to thank here who have undoubtedly made the last several years a more enjoyable and rewarding experience. Firstly, I am deeply indebted to my advisor Tian Zheng. She has been an excellent colleague and mentor, instinctively knowing when I have needed guidance while also knowing when I needed to fend for myself. Tian's enthusiasm and passion is infectious, and I have learned much from her.

I would also like to thank the members of my committee, Zhiliang Ying, James Curley, Yuqi Gu and John Paisley for being generous with their time and for their insights and feedback. I thank Zhiliang and James for their support and guidance on the job market this year.

I have been lucky to work with and make many friends during my time at Columbia, including Andrew Davison, Alessandro Grande, Chengliang Tang, Ding Zhou, Guanhua Fang, Jing Wu, Jonathan Auerbach, Tim Jones, Phyllis Wan, and many others. A PhD is a daunting task and spending time around the many great people in the department has made that task much more enjoyable. I also want to thank Dood Kalicharan and Anthony Cruz and the other members of the department for everything they do for making our lives as students easier and making the department a great place to be. As supportive as humans can be, sometimes a friend is required who doesn't know anything about statistics. I've been lucky to spend time with some great dogs in New York, including Cobi and Kassi, who knew exactly how to take my mind off work.

My parents Finnian and Noreen, my brothers Robert and Fintan and my sister Maeve have always been supportive of everything I've wanted to do, and I thank them for their belief in me, along with always being able to distract me by talking about hurling.

Last, but certainly not least, I thank my wife, Sophie, without whom none of this would have been possible, nor would it make any sense. Thank you for always believing in me, being there for me, and for being you.

Do mo thuismitheoirí agus mo bhean, Sophie.

Chapter 1: Introduction

The study of network data is a prominent area of interest across many fields. Network data can occur in a wide range of domains, and has been used to describe data such as communication networks (Baumes et al., 2004), protein interactions (Wu, Vallenius, et al., 2009), and academic citation patterns (Lazer et al., 2009). Social network data, describing social interactions between actors in a network, is widely collected and available. Such data has been used for a wide variety of tasks, including understanding social relationships (Girvan and Newman, 2002), explaining the spread of diseases through a network (Morris and Kretzschmar, 1995), and examining corporate structures across companies (Friel et al., 2016).

In many applications, particularly for common examples of social networks, of crucial importance are the raw interaction events that take place over the unobserved or partially observed network. Aggregating such events into a simple adjacency matrix could inadvertently mask important and interesting structures and patterns. For example, a classical social network dataset may simply indicate the presence or absence of an “edge” between two nodes, when in reality these nodes interact repeatedly in time. A typical example of this consists of the Enron email corpus, where emails between 150 Enron employees were made public (Klimt and Yang, 2004). This dataset contains the exact times of these emails along with their recipient lists and the content of the emails. While this data has been widely used for fitting network models, it is commonly aggregated to a static network, removing the temporal component and instead reducing interactions to simply indicating whether there was at least one email exchanged between employees. Methods for network data can therefore be somewhat unrealistic, and may not take account of all the information available. This aggregation is common for many widely used network datasets. Methods for network data which can incorporate available temporal information may be better able to capture the true dynamics (social or otherwise) present in such networks.

Event data on networks can be used to understand social dynamics present between nodes in a network, with the goal of revealing information about how why and how nodes in a network interact. For example, is a node more likely to interact with other nodes which have similar characteristics (such as nodal covariates)? Can the nodes be partitioned into groups such that nodes in the same group behave similarly? Statistical models for networks which can identify such social characteristics are of particular interest.

While properties of these networks, such as community structure, are of interest, they are usually not directly observed. A natural way to infer such characteristics of a network is through the use of latent variable models. Such models posit that there are some underlying parameters which give rise to the observed interactions which occur on the network. It is these latent characteristics of the nodes which are of interest. For example, whether two nodes in a network interact could be determined by some unobserved information, such as whether they are members of the same community, with interactions between members of the same community much more likely than between communities. While we do not observe such community structure directly, the observed network data may be inherently generated by it. Latent variable models connect the raw observed interaction data with underlying properties of the network and the nodes in the network. Through such models we are able to infer characteristics of the underlying latent structure of the network which can lead to insights into properties of interest, such as understanding social dynamics between the nodes.

In this thesis we focus on latent variable models for event data on networks. In Chapter 2, we thoroughly review the important components of latent variable models for network data, first through the lens of static network data, represented only by an adjacency matrix. We review what social structures such models seek to capture, along with extensions of these network models to incorporate dynamic network data, consisting of repeated observations of an adjacency matrix. We then consider the concept of event data in general and define models for event data on networks, highlighting the existing work in this direction.

In Chapter 3 we consider the problem of learning social hierarchy from aggressive interactions

between mice. Group-based social dominance hierarchies are of essential interest in understanding social structure (DeDeo and Hobson, 2021). Recent animal behaviour research studies can record aggressive interactions observed over time. Models that can explore the underlying hierarchy from the observed temporal dynamics in behaviours are therefore crucial. Traditional ranking methods aggregate interactions across time into win/loss counts, equalizing dynamic interactions with the underlying hierarchy. Although these models have gleaned important behavioural insights from such data, they are limited in addressing many important questions that remain unresolved. In this Chapter, we take advantage of the observed interactions’ timestamps, proposing a series of network point process models with latent ranks. We carefully design these models to incorporate important theories on animal behaviour that account for dynamic patterns observed in the interaction data, including the winner effect, bursting and pair-flip phenomena. Through iteratively constructing and evaluating these models we arrive at the final cohort Markov-Modulated Hawkes process (C-MMHP), which best characterizes all aforementioned patterns observed in interaction data.

The generative nature of our model provides evidence for hypothesised phenomena and allows for additional insights compared to existing aggregate methods, while the probabilistic nature allows us to estimate the uncertainty in our ranking. In particular, our model is able to provide insights into the distribution of power within the hierarchy which forms and the strength of the established hierarchy. We compare all models using simulated and real data. Using statistically developed diagnostic perspectives, we demonstrate that the C-MMHP model outperforms other methods, capturing relevant latent ranking structures that lead to meaningful predictions for real data.

In Chapter 4 we address the problem of efficient inference for models incorporating events on networks. Model fitting in this setting can be challenging, as we must account for both the number of nodes in the network and also the number of events between these nodes. Computational complexity hampers the scalability of such approaches to large sparse networks. To circumvent this challenge, we propose a fast online variational inference algorithm for learning the community structure underlying dynamic event arrivals on a network, using continuous-time point process

latent network models. The proposed inference procedure is then compared, using both simulation studies and real data, to non-online variants. We demonstrate that online inference can obtain comparable performance, in terms of community recovery, to non-online variants, while realising computational gains. This procedure can work well for such data, and can capture the true latent dynamics present in an online fashion, without the need to repeatedly process all the data or to even store all the data. Our proposed

Finally, in Chapter 5 we summarise the questions addressed in this thesis and consider important future directions which can build on this work. We briefly discuss the questions of how to assess the fit of network models. We also consider the question of privacy in network data, a pertinent issue with the wide range of social network data that is currently collected.

Chapter 2: Background

Statistical models for network data have been widely studied, both theoretically and in practice. Here we review the components of these models, highlighting some of the main existing methods in the context of this thesis. Finally, we review the literature on social dynamics and their connection to network data, motivating the developments of later chapters.

2.1 Network Data

Networks can be used to encode relationships between nodes. For social networks, this is often social relationships, such as friendship between people or positive relationships between animals. To illustrate this, we consider a classic network, describing social interactions among dolphins (Lusseau et al., 2003; Lusseau, 2003). Here, a population of dolphins was observed over several years. A social network was then constructed, where an edge between two dolphins denotes ‘social companionship’, meaning these dolphins were observed together more often than expected. Another famous example of network data concerns the Enron email data, where the nodes consist of the employees of the company and each interaction via email is recorded.

In its simplest form, network data is used to describe the static relationship between a series of nodes. This is determined by the set of nodes (also called vertices or actors) in the network, \mathcal{V} , and the set of relationships between these nodes, which are given by a set of edges, \mathcal{E} . Then a network G is determined by these objects, $G = G(\mathcal{V}, \mathcal{E})$. For a network containing n nodes, we shall describe the relationship between these nodes using an $n \times n$ matrix A , where A_{ij} encodes the relationship from node i to node j .

In many types of network data, relationships between nodes can be described by a binary value, with $A_{ij} = 1$ indicating there is a “tie” from node i to node j and $A_{ij} = 0$ indicating there is no tie.

For example, in the dolphin dataset, $A_{ij} = 1$ would indicate social companionship between dolphin i and dolphin j . These relationships may be directed, meaning that we do not require A_{ij} and A_{ji} to take the same value. If the relationships between nodes are undirected then $A^T = A$.

2.2 Statistical Models for Network Data

Statistical models for the adjacency matrix representation of a network, A , have been widely studied. Such models posit that the relationships between nodes in a network are a result of some latent properties of either the network itself and/or the nodes in the network. Interest is then in the estimation of the latent variables in these models and what social structure they can describe. For example, in the classic Erdős Renyi model (Erdős and Rényi, 1960; Gilbert, 1959), it is assumed that each pair of nodes in the network has equal probability p of forming a tie, that is,

$$P(A_{ij} = 1) = p, \forall i \neq j.$$

This model, while mathematically simple, is not sufficient to capture the structure commonly seen in real network data.

A natural goal, in the context of network data, is to identify groups of nodes which show similar characteristics compared to the other nodes in the data. This is known as *community detection*. Community detection has found a broad range of applications, from identifying proteins with similar functions in Protein-Protein interaction networks (Jonsson et al., 2006), to malicious actors in a social network (Lu et al., 2015). Statistical tools beyond the Erdős Renyi model are needed to incorporate such structure in networks. The first such statistical model of this form is the Stochastic Block model (SBM), first proposed for social network data by Holland et al. (1983).

This model assumes that there are K latent communities, with each node belonging to one such community. The probability a node belongs to community k is π_k , with $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k)$. Then, the probability of an edge between any two nodes is determined by the community assignment of the corresponding nodes. In particular, the edge probabilities between any two nodes i and j are

determined by the latent community assignment $\mathbf{z}_i, \mathbf{z}_j \in \{1, \dots, K\}$ and a $K \times K$ block matrix B , where

$$p(A_{ij} = 1 | \mathbf{z}_i, \mathbf{z}_j) = B_{\mathbf{z}_i, \mathbf{z}_j}.$$

This model enforces a stochastic equivalence between nodes in the same community. The probability of an edge from a node in community k_1 to a node in k_2 is the same for any node pair from these communities. This requires exact community assignment for each node. In particular, we can write \mathbf{z}_i as an indicator vector

$$\mathbf{z}_i = \{0, \dots, 0, 1, 0, \dots, 0\},$$

where all but one entry is 0, with the indicator in the k th entry corresponding to the community assignment of node i . A natural extension is to relax this strict assignment, allowing more flexible forms of \mathbf{z}_i . For example, the mixed membership stochastic block model (MMSBM) (Airoldi et al., 2008) instead equips each node with a K dimensional mixed membership vector $\mathbf{z}_i = (z_{i1}, \dots, z_{iK})$ on the $K - 1$ simplex such that $z_{ij} > 0$ for all $j \in \{1, \dots, K\}$ and $\sum_{j=1}^K z_{ij} = 1$.

These models assume that the nodes belong to some latent space, with restrictions on the geometry of that space, with the MMSBM resulting in a more flexible community structure than the SBM model. It is also possible to relax this further, requiring only that the nodes have a position in some arbitrary latent space. Although other geometries have been considered (McCormick and Zheng, 2015; Smith et al., 2019), Euclidean space is often used. Models of this form for social network data were first introduced by Hoff et al. (2002). These models suppose that the probability of an edge between two nodes is then a function of their position in this latent space. For example, if $\mathbf{z}_i, \mathbf{z}_j \in \mathbb{R}^d$ then this can be written as

$$A_{ij} \sim \text{Bernoulli}(p_{ij})$$

where

$$\text{logit}(p_{ij}) = \alpha + s(\mathbf{z}_i, \mathbf{z}_j).$$

Here α gives a baseline probability for edge formation while $s(\mathbf{z}_i, \mathbf{z}_j)$ is a measure of similarity between the position of the two nodes. This model assumes that edges between any two nodes are conditionally independent, given the latent positions. For example, s could be the negative distance between nodes in \mathbb{R}^d . This choice of s will naturally encode transitivity, a phenomenon observed in many real social networks (Hoff et al., 2002). Transitivity in networks states that if there is an edge between node i and node j , and there is also an edge between node j and node k , then it is likely that there is an edge between node i and node k . When distance is used as a similarity measure then transitivity follows directly from the triangle inequality. Such latent space models are commonly fit using Euclidean space of dimension $d = 1$ or $d = 2$. This allows a natural visualisation of the nodes in the network in a latent space, as seen in Figure 2.1.

2.3 Inference for Network Data

Given a chosen latent variable model, an inference scheme needs to be devised to estimate the parameters of such a model given the data. There are several key inference methods of interest when fitting statistical models to social network data, which are reviewed briefly below.

Spectral Clustering We first highlight spectral clustering, which is a key tool for the explicit goal of community detection in network data. Spectral clustering for an adjacency matrix A consists of constructing an embedding of the network using the eigenvectors of the graph Laplacian based on A . The eigenvectors corresponding to the k smallest eigenvalues are used to construct $U \in \mathbb{R}^{n \times k}$, with some clustering method (commonly k -means) then applied to the rows of U . Under mild conditions, Spectral Clustering can recover the underlying community structure in data generated from a SBM (Lei and Rinaldo, 2015). Similarly, spectral clustering and spectral methods more generally have well established theoretical properties in other settings (Rubin-Delanchy et al., 2017; Von Luxburg et al., 2008). However, this procedure is computationally intensive and

Latent Space Model for Dolphin Data

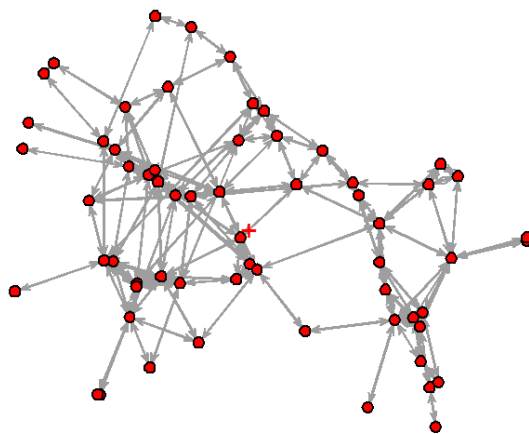


Figure 2.1: The output of a two-dimensional latent space model fit to the dolphin interaction data of Lusseau (2003).

computing the decomposition of the graph Laplacian is $O(n^3)$, limiting the applicability of this method to the very large networks which occur in modern applications.

Model Inference While spectral clustering has been widely used and shown to be consistent at community detection for some latent variable models for network data, often the goal is to also infer the parameters of the chosen latent model. For example, when fitting a latent space model, we are interested in estimating the latent positions for each node. A common theme in statistical models for network data is that the likelihood function may not be tractable for even small networks, making likelihood based methods challenging. While EM algorithm based methods have been implemented in some settings (Snijders and Nowicki, 1997), Bayesian methods utilising

Markov Chain Monte Carlo (MCMC) are also commonly used for network models. Nowicki and Snijders (2001) describe a Gibbs sampler for the SBM model, under suitable Dirichlet and Beta priors for π and B_{z_i, z_j} . Similarly, Hoff et al. (2002) use a modified Metropolis Hastings procedure to fit a latent space model for network data, accounting for the invariance present in the distance between a set of points in Euclidean space.

MCMC methods have been applied to network data, however they are often unsuitable for large networks, which are commonly obtained in modern applications. While there has been work aimed at modifying these methods, allowing them to scale to large networks (Raftery et al., 2012; McDaid et al., 2013), a popular alternative has been to utilise Variational inference schemes.

Variational Inference Methods Although Markov Chain Monte Carlo methods have been common in statistical inference for several decades, variational inference has emerged more recently as a tool for fast inference in complex models (Blei et al., 2017). Variational inference posits a family of approximate densities over the latent variables in a model, \mathbf{z} . Then, it seeks to find the member of that family which best agrees with the true posterior, $p(\mathbf{z}|y)$. In particular, a variational family $q(\mathbf{z}) \in \mathcal{Q}$ is chosen and the “best” choice of q is the variational approximation. The best choice of q is the distribution closest to the true posterior in Kullback-Liebler (KL) divergence. However, this is not a tractable problem and instead q is chosen by maximizing the evidence lower bound (ELBO), which for a variational family q is given by

$$ELBO(q) = \mathbb{E}_q[\log p(y, \mathbf{z})] - \mathbb{E}_q[\log q(\mathbf{z})]. \quad (2.1)$$

For a model with many latent variables $\mathbf{z} = (z_1, \dots, z_n)$ then specifying a variational family across these variables can be challenging. Commonly, a mean field approximation is used, which assumes that each of the latent variables are mutually independent, giving

$$q(\mathbf{z}) = \prod_{i=1}^n q(z_i).$$

This allows the variational posterior to be specified instead by a sequence of independent univariate distributions. To illustrate this, we consider Variational Inference for the SBM model. Given that the latent community assignments $z_i \in \{1, \dots, K\}$ a natural variational family for z_i is

$$q(z_i) = \mathcal{M}(1, \tau_{i1}, \dots, \tau_{ik}),$$

where $\mathcal{M}(1, \tau_{i1}, \dots, \tau_{ik})$ denotes a single draw from the multinomial distribution with event probabilities $\tau_{i1}, \dots, \tau_{ik}$. Given this family, we can write the ELBO as

$$ELBO(q) = \sum_{i \neq j} \sum_{k,l=1}^K \tau_{ik} \tau_{jl} [A_{ij} \log B_{kl} + (1 - A_{ij}) \log(1 - B_{kl})] + \sum_{i=1}^n \sum_{k=1}^K \tau_{ik} \log \left(\frac{\pi_k}{\tau_{ik}} \right)$$

Then, given this representation a Variational EM (VEM) procedure can be implemented. In the variational E step, the ELBO is maximised with respect to τ . Then, given these estimates of τ , the M step maximises the ELBO with respect to both π and B respectively. This VEM procedure for the SBM was first implemented by Daudin et al. (2008).

Variational Inference can work well in this setting, and it can also be readily modified to scale to large data, a challenge for all inference methods for real networks. Hoffman, Blei, et al. (2013) introduce Stochastic Variational Inference (SVI), leveraging tools from stochastic optimization to scale variational inference to massive data. This is done by directly considering optimisation of the ELBO, taking estimates of the gradient using a subsample of the data. In the SBM setting, we can readily compute the gradient of this ELBO in terms of each parameter. Then, we can update the parameters $\theta = (z, B, \pi)$ by moving in the directions of these gradients with an appropriate choice of step size. Gopalan and Blei (2013) consider such a procedure to fit a MMSBM to large real networks, identifying interpretable community structure in citation networks of physics papers and US patents.

Identifiability in Network Data Identifiability is a common challenge when fitting statistical models to network data. In particular, for the latent position model of Hoff et al. (2002), there are an infinite number of latent positions which will all give the same log-likelihood. Similarly, Allman et al. (2011) study the issue of identifiability in general random graphs. In practice post-processing is often needed to deal with these concerns. For example, Handcock et al. (2007) use both a Procrustes transformation of the latent positions and also the relabelling algorithm of Celeux et al. (2000) to consider the best permutation of cluster labels.

While the majority of theoretical results for network models have focused on applying spectral clustering and other community detection methods and showing they can recover the communities in an SBM, other approaches have also been considered. Amini et al. (2013) show a pseudo likelihood approach can consistently recover the communities in an SBM, while Celisse et al. (2012) show the consistency of both the maximum likelihood and variational estimators of the block matrix and the group proportions for an SBM. Rubin-Delanchy et al. (2017) discusses the connection between spectral embeddings and latent position models.

2.4 Network Data across Time

While much work in statistical network analysis has focused on static networks, which can be represented by a single adjacency matrix, interest in relationships on a network observed across time has grown in recent years. In reality, many of the classical examples of static network data are actually constructed from data with some underlying temporal component. For example, the original data of Lusseau (2003) consisted of observing which animals appeared together over many observation periods, which was then aggregated to form a single network. Given the close connection between static network data and networks observed across time, it is unsurprising that many methods for data of this form are extensions of existing models. Before highlighting some existing methods, we wish to make clear the distinction between the different types of network data which can be observed across time.

What do we mean by networks across time? Several formulations of network data across time have been considered in the literature. The most immediate extension from a static network model is to consider repeatedly observing a complete network at discrete time points. We shall refer to this formulation as a *dynamic network*. For dynamic network data we have discrete time points $t = 1, \dots, T$ and the observed data consists of the presence or absence of an edge at a given t . Instead of a single adjacency matrix A we now observe a sequence of such matrices $\mathcal{D} = (A_1, \dots, A_T)$ where $A_{ijt} = 1$ if there is an edge from node i to node j at time t . This means that the presence of an edge between nodes may vary across time.

Alternatively, instead of repeatedly observing an adjacency matrix at discrete time points, we can instead observe continuous time information, describing the exact event time for an interaction between two nodes. In particular, we will denote by t_m^{ij} an event occurring from node i to node j at time $t_m \in [0, T]$. Our observed data will then be

$$\mathcal{D} = \left\{ \{t_m^{ij}\}_{m=1}^{M^{ij}}, i, j = 1, \dots, n, i \neq j \right\},$$

where M^{ij} total interactions are observed from node i to node j across the observation period. We shall refer to data of this form as *network event data*. Data of this form is called *longitudinal network data* by Matias, Rebafka, et al. (2018). Note that if we observe network event data, this could be aggregated to form dynamic network data. When the true event times are known, this will lead to a loss of information and can limit inference. We also note that there have been other models considering time and network data. Rastelli and Fop (2020) considered a block model in the context of a single interaction between nodes, observing both the time of that interaction and the interaction length.

2.5 Models for Dynamic Networks

Several extensions of the popular latent variable models for static networks have been extended to the dynamic setting. Here we highlight some of these which connect to the static models previ-

ously discussed, with Kim et al. (2018) providing a detailed review of these and other methods.

Dynamic Blockmodels There are several ways that Block models can be extended to model network data in the dynamic setting. Recall that for the SBM model, the adjacency matrix of a network, A is parameterized by community assignment vectors $\mathbf{Z} = (z_1, \dots, z_n)$ and a block matrix B . In the dynamic setting, we now observe repeated matrices A_1, \dots, A_T . Here, the network can evolve in time and this could be driven by either the community membership evolving over time and/or the block probability matrix evolving over time. Yang et al. (2011) first considered this problem, leaving B fixed and allowing only the community assignment to change across time. With \mathbf{Z}_t denoting the community assignment of the nodes at time t , then A_t is a standard SBM generated from \mathbf{Z}_t and B . The evolution of the community assignment from t to $t + 1$ is then defined by a transition matrix $C \in [0, 1]^{K \times K}$, where C_{kl} gives the probability that node moves from community k to community l at the next time point. Yang et al. (2011) implement a Variational EM algorithm, along with an inference procedure utilising Gibbs sampling along with simulated annealing. They investigate the changing community structure in paper co-authorship over two-year periods.

Matias and Miele (2017) discuss some challenges for such models. In particular, whether the group memberships or the connection parameters vary over time (with the other fixed) can result in similar dynamics. Restricting to the connection probabilities being static across time, Matias and Miele (2017) identify interpretable evolving community structure in human and animal social networks. Xu and Hero (2014) consider this setup also, but did not account for the difficulties that occur due to label switching when both the probabilities and memberships can vary.

Dynamic Latent Space Models Recall that the latent space model for a static network supposes that each node has a position in some (possibly non-Euclidean) latent space. Much like the extension Yang et al. (2011) consider for the SBM, the latent space model can be extended to the dynamic setting, by controlling how the latent positions can change over time. Sarkar and Moore (2006) assume that A_t depends only on the current position of the nodes, \mathbf{Z}_t , with the latent positions following a Gaussian random walk with $Z_t | Z_{t-1} \sim \mathcal{N}(Z_{t-1}, \sigma^2 I)$. They illustrate

the evolution of the latent positions of conference authors over three time periods. Sewell and Chen (2015) consider a similar model, incorporating global popularity and activity effects and are able to consider directed networks. This model is fit using a Metropolis-Hastings within Gibbs procedure, using a Procrustes transformation to deal with invariance in the latent positions. This is used to examine the social dynamics seen in a school social network and also in congressional co-sponsorship.

2.6 Point Processes and Models for Network Event Data

Although dynamic networks have been widely studied, models for network event data are considerably less common, although this is often the more appropriate setting for real data. Such models consider event arrival times across the network, utilising point processes to model such data. Here we review the important components of general models of this form, along with existing methods for such data and the inference procedures considered.

Point process models Before describing models for events on networks we first consider univariate event arrival time data that consists of all event history up to a observation time T : $\mathcal{H}(T) = \{t_m\}_{m=0}^M$, where $t_0 = 0$, $t_M = T$, and M is the total number of events. An equivalent representation of this event history $\mathcal{H}(T)$ is via a *counting process*, $N(t)$, where $N(t)$ is a right-continuous function that records the number of events observed during the interval $(0, t]$. The associated stochastic property is usually specified by its conditional intensity function $\lambda(t|\mathcal{H}(t))$ at any time $t \in (0, T]$, conditioning on current history $\mathcal{H}(t)$,

$$\lambda(t|\mathcal{H}(t)) = \lim_{\Delta t \rightarrow 0} \frac{\Pr(N(t + \Delta t) - N(t) = 1|\mathcal{H}(t))}{\Delta t}.$$

This is the instantaneous expected rate of events occurring around a time t given the history. Inference on the intensity function is conducted by evaluating the likelihood function for a sequence of events up to time T , $\mathcal{H}(T)$, which can be expressed as (Daley and Jones, 2003)

$$\prod_{m=1}^M \lambda(t_m | \mathcal{H}(t_m)) \exp \left\{ - \int_0^T \lambda(s | \mathcal{H}(s)) ds \right\}. \quad (2.2)$$

The simplest such point process model is to assume that the intensity function is constant across time, resulting in a homogeneous Poisson process with rate $\lambda(t) \equiv \lambda$. That the intensity does not vary in time is often a strong assumption and in many real examples a homogeneous Poisson process is not sufficient to describe observed event times. As such, models have been developed which allow temporal heterogeneity in the intensity function. A *Hawkes process* (Hawkes, 1971) is a linear self-exciting process that can explain bursty patterns in event dynamics. For a univariate model, the intensity function with exponential triggering function is defined as

$$\lambda(t) = \lambda_1 + \sum_{t_m < t} \alpha e^{-\beta(t-t_m)}, \quad (2.3)$$

where $\lambda_1 > 0$ specifies the baseline intensity, $\alpha > 0$ calibrates the instantaneous boost to the event intensity at each arrival of an event, and $\beta > 0$ controls the decay of past events' influence over time.

Network point process models Now we extend the formulation of a univariate point process to a network, consisting of a fixed set of n nodes, $V = \{1, 2, \dots, n\}$. For each directed pair of nodes (i, j) , the observations of interactions between them up to terminal time T includes the sender i , the receiver j and a sequence of event times $\mathcal{H}^{ij}(T) := \{t_m^{ij}\}_{m=0}^{M^{ij}}$. Hence, a network Hawkes process model has a conditional intensity function for each pair (i, j) at time t given by $\lambda_{ij}(t | \mathcal{H}^{ij}(t))$. The likelihood of the observed interactions on the whole network is then

$$\prod_{i=1}^n \prod_{j \neq i}^n \prod_{m=1}^{M^{ij}} \lambda_{ij}(t_m^{ij} | \mathcal{H}^{ij}(t_m^{ij})) \exp \left\{ - \int_0^T \lambda_{ij}(s | \mathcal{H}^{ij}(s)) ds \right\}.$$

This is the general formulation for considering event data between nodes on a network. Given this, we wish to consider how to parameterize the conditional intensity functions, λ_{ij} , in the context of inferring social structure present in the network and between the nodes. There has been

some recent work in this direction, largely building off the existing structures considered for static network data.

Blockmodels Building on the literature for static and dynamic networks, a natural approach to consider community structure for data is the use of block models. Matias, Rebafka, et al. (2018) first extended the Stochastic Block model to this setting. Each of the n nodes belongs to a latent community z_i and events between node i and node j are modelled by a conditional inhomogeneous Poisson process, the intensity depending only on the latent groups z_i, z_j . A variational EM procedure is described to fit this model, with the E step used to estimate the variational group assignments. Meanwhile, for the M step which estimates the non-parametric intensity, both a histogram and kernel based M step are proposed. Applying this to London bike share data, Matias, Rebafka, et al. (2018) capture geographically interpretable clusters, which agree with patterns of commuting.

Arastuie et al. (2020) consider a similar block structure, restricting the intensity function to the Hawkes process, calling this the *Community Hawkes Independent Pairs* model. They show that when restricted to this intensity, spectral clustering on the aggregate node pair counts results in consistent community detection.

Other Models Outside of block models, other network models have also been considered for network event data in the context of social networks. Fox et al. (2016) model email interactions networks, examining the relationship between the rate of events from each node in the network and the known leadership role of nodes. Here, reciprocity is incorporated by allowing the rate at which an individual replies to an email to depend on whom they received it from. Fox et al. (2016) examine the correlation between the parameter estimates and the perceived leadership ranking of the nodes by other members of the network.

Miscouridou et al. (2018) also incorporate reciprocity in this context, with a conditional intensity function of the form

$$\lambda_{ij}(t) = \mu_{ij} + \int_0^t g_\phi(t-u) dN_{ji}(u).$$

Here the base rate is a function of the latent affiliation of the nodes to each of K latent communities with $\mu_{ij} = \mu_{ji} = \sum_{k=1}^K w_{ik}w_{jk}$. The w 's are latent affiliations modelled by a compound continuous random measure. Inference for this model is challenging, particularly for the procedure to scale well as the number of interactions increases. Miscouridou et al. (2018) propose a two-step Bayesian estimation scheme, utilising existing algorithms to infer the parameters of the base rate, with a standard Metropolis Hastings algorithm for the Hawkes parameters. This model is shown to have good performance in predicting the number of future events, while the latent affiliation parameters appear to identify community structure.

Passino and Heard (2021) extend reciprocity to incorporate mutual excitation in these models. In particular,

$$\lambda_{ij}(t) = \alpha_i(t) + \beta_j(t) + \gamma_{ij}(t),$$

where $\alpha_i(t)$ is a self exciting component relating to all events originating from node i while $\beta_j(t)$ is a self exciting component depending on all events received by node j . $\gamma_{ij}(t)$ encodes only the history of events from i to j . This model is fit and evaluated on an IP network. Fan et al. (2021) also considered a similar setting, where mutual excitation was possible, with kernel functions based on Gaussian processes, where the excitation functions can be shared across nodes.

Finally, Sit et al. (2021) consider estimation where the mean function of the counting process from node i to j is of the form

$$\mathbb{E}\{dN_{ij}(t)|Z_{ij}(t)\} = \exp\{\beta_0^T Z_{ij}(t)\} \lambda_0(t) dt,$$

where $Z_{ij}(t)$ are covariates and interest is in the estimation of β_0 in the presence of an unknown baseline function $\lambda_0(t)$. A pseudo partial likelihood approach is developed and the theoretical properties of this procedure are established.

Challenges There are several challenges when fitting statistical models for network event data. Computational complexity hampers network models for static data, and this is very much the case also when event data across the network is incorporated. Many of the existing methods fit complex intensity functions to this data, fitting these functions across all possible node pairs. As such, tools which could improve the computational performance of models for event data for networks are essential. Similarly, while many methods have been established for data of this form, a common challenge is out to evaluate the fit of these models. While much work has been devoted to the task of model fitting and model checking for network data, further tools are required, as described in Ward et al. (2021). However, considering event data on a network and parameterizing such models through point processes has many natural advantages here. There are a wide range of well established diagnostics for univariate temporal point processes, such as the time rescaling theorem of Brown et al. (2002). For event times $0 < t_1 < \dots < t_i < \dots < t_n < T$ that are a realisation from a point process with conditional intensity function $\lambda(t|\mathcal{H}(t))$ then we can define

$$\Lambda(t_i) = \int_0^{t_i} \lambda(u|\mathcal{H}(u))du.$$

The result of Brown et al. (2002) shows that these $\Lambda(t_i)$ are realisations from a homogeneous Poisson process with unit rate. This implies that the difference between these terms,

$$\Lambda(t_i) - \Lambda(t_{i-1}),$$

are exponentially distributed with unit rate. This means to evaluate the fit of a point process to data, we can assess the distribution of these quantities. Classical tests, such as a Kolmogorov-Smirnov test, can be used for this task. Similarly, we can also compute a residual from fitting an estimated point process to some data. Clements et al. (2011) construct a normalized residual for temporal point processes, defined at time t as

$$PR(t) = \sum_{t_n < t} \frac{1}{\sqrt{\lambda(t_n|\mathcal{H}(t_n))}} - \int_0^t \lambda(s|\mathcal{H}(s))ds.$$

Under the true conditional intensity function then these residuals will have mean 0 and variance t , i.e, the residual does not depend on the intensity function. These results provide convenient tools for evaluating the fit of temporal point processes and Sun et al. (2021) provides a documented R implementation.

In this thesis we will consider events observed across networks and so we can utilise the existing model checking tools for temporal point processes in this context, providing natural pairwise residuals and test statistics for network data. This was first considered by Wu, Smith, et al. (2021). In particular, having fit a point process model across a network, we can compute a matrix of residuals, corresponding to events between each node. Evidence of structure in these residuals can then be used to motivate model refinement and iteration.

2.7 Social Dynamics on Networks

To finish this chapter, we wish to review some of the common social dynamics that are of interest in network data and which of these dynamics can be captured in social network models.

Transitivity, which was introduced in Chapter 2.2, is widely observed in large real networks. Models such as the latent space model of Hoff et al. (2002) can naturally incorporate such properties in a statistical model. A related idea, if nodal covariates are observed, is the concept of *homophily*. That is, nodes which have similar covariates should be more likely to interact. Edge covariates have been incorporated in the original latent space model of Hoff et al. (2002) and in more recent work, such as Ma et al. (2020). Similarly, nodal covariates have been considered for the SBM (Huang and Feng, 2018). Real networks can also be sparse, with most nodes only interacting with a small proportion of the overall network. Similarly, the degree distribution of nodes in a real network is often heterogeneous, as is usually seen when fitting the SBM model to real networks (Karrer and Newman, 2011). In particular, the classic SBM assumes that all nodes in a community have the same expected degree. To allow the degree between nodes to vary, each node can be given a parameter controlling its expected degree. This degree correction is important in correctly capturing the properties of real networks and has been widely implemented.

Chapter 3: Latent Ranking Models for Social Animal Interactions

3.1 Introduction

In this chapter we consider the problem of providing a general model-based framework for exploring the unobserved social hierarchy among a group of mice through their observed repeated aggressive interactions. We do this using data from the study conducted by Williamson et al. (2016), in order to address unsolved questions in that work. In particular, describing the dominance structure behind such interactions well is a difficult task, and existing methods cannot adequately capture all possible dynamics, quantify uncertainty in the ranking, or provide insights into how a power hierarchy is formed or the distribution of that hierarchy. There is no existing method which can describe how the observed interactions are generated from the underlying social hierarchy. Similarly, how mice are able to recognize their social status relative to other mice and how this recognition facilitates hierarchy formation and maintenance remains an unanswered question. The temporal dynamics in these interactions are driven by the need of the mice to explore, recognize, maintain, and exploit their positions in such a hierarchy, through mechanisms that are not fully understood. Section 3.2 presents an overview of existing well-known methods for dominance ranking and their properties. These existing methods, which generally utilise aggregate data, suffer several common issues, including the inability to rigorously evaluate the estimated ranking and the inability to deal with the temporal component of these interactions, which is likely influenced by the animals' gains of social information about their group's structure (Hobson et al., 2021). Specifying statistical generative models therefore provides a natural way to characterize the structure of these social groups more generally. One focus of the models we develop here is the ability to capture the latent stable dominance hierarchy via modelling the temporal and network dynamics of these social interactions. Previous work in the animal behaviour literature has indicated that there

is some influence between hierarchy formation and prior attributes of these animals (Chase, Tovey, et al., 2002; Chase and Lindquist, 2017). When these animals are placed together, a realized dominance hierarchy materializes over time, as the animals navigate their place in the social structure and gauge their own abilities (Hobson, 2020). During this process, animals may exhibit different activity levels and interaction patterns, given a stable inherent hierarchy, due to the different needs of exploration and exploitation. Existing methods in the animal behaviour literature attempt to describe the inherent social structure, by using the win/loss matrix directly (David, 1987). However, this matrix, as a noisy and dynamic realisation of this structure, is driven mostly by the more active animals in a group. In particular, ranking algorithms that optimise some function of win/loss matrices directly do not offer a generative statistical model that can be used for further inference. Our model aims to infer a static latent ranking behind the observed dynamic interactions. In Section 3.3, we take advantage of the timestamps of these interactions and propose three network point process models: the cohort Hawkes process model (C-HP), the cohort degree-corrected Hawkes process model (C-DCHP) and the cohort Markov-modulated Hawkes process model (C-MMHP). We construct these models such that these point processes are a function of a set of latent rank variables. These latent rank variables are a powerful feature of our models, allowing us to incorporate various known traits of animal behaviours in a social hierarchy into our model. We develop these models in a Bayesian framework to capture uncertainty estimates and to better model pairs which contain few interactions. We iteratively develop each model from the previous to better account for dynamics seen in animal data. This results in our final Cohort Markov-Modulated Hawkes Process (C-MMHP) model. In Section 3.4, these models are compared, using simulated and real data, to existing methods for understanding animal dominance ranking, highlighting how different methods capture different behaviour components, leading to different rank estimates. We illustrate that our final model is flexible and adequately captures dynamics driven by the group’s inherent dominance hierarchy by showing results on rank inference, prediction performance and residual analysis. These point process models, therefore, provide a new utility for future research that can lead to better understandings of the dominance hierarchies among animals and be used to generate

further research questions. Section 3.5 summarizes this work and discuss future directions for our proposed model.

3.2 Background

Here we review the literature on social hierarchy for group-living animals. Empirical studies of the social hierarchy of animals that live in a group are generally developed based on the observations of dyadic, or pairwise, agonistic interactions. In Williamson et al. (2016), the agonistic interactions include fighting, chasing and mounting behaviours. We consider all such aggressive interactions without differentiating the type, as is often done in this area (Lee, Fu, et al., 2019). We denote the interactions between N animals as a matrix W , where W_{ij} is the number of aggressive interactions won by animal i against animal j . In So et al. (2015) and Williamson et al. (2016), this is also called a *win/loss* matrix.

Two approaches are generally considered in the animal behaviour literature to analyse this win/loss matrix (Drews, 1993): *functional* methods and *structural* methods. *Functional* methods aim to directly infer a ranking of animals from this win/loss matrix by rearranging this matrix in an attempt to best capture behavioural patterns, expected in a social hierarchy. The rank is therefore inferred directly from the observations recorded in the *win/loss* matrix. If $W_{ij} > W_{ji}$ then functional methods infer that i dominates j . Alternatively, *structural* methods propose an indirect model-based approach, associating a latent ranking variable F_i with individual i . If $F_i > F_j$ then these *structural* methods infer that i dominates j . These latent variables are constructed to satisfy a set of a priori assumptions, and structural models attempt to estimate these latent ranks to best align with the behaviour captured in W .

An important concept in dominance ranking is *linearity*. Under a strict linearity assumption, for any three individuals, i, j , and k , if i dominates j and j dominates k , then i is assumed to dominate k . In social network research, this closed triad relationship is also called *transitivity*. For *functional* methods, the linearity assumption intuitively follows from observational studies of group-living animals. However, phenomena that violate a strict linearity assumption are often

observed. In these cases, *functional* methods aim to find a nearly linear ranking that is most consistent with the observed wins and losses. Meanwhile, in *structural* methods, the linearity assumption is not directly observable but is incorporated as a property of the latent parameter F . In particular, each animal i has a latent parameter f_i and as such, we cannot directly observe the relationship between f_i and f_j . Similarly, even if this linearity assumption is satisfied by the latent variables, the realised events are a random sample from these latent values and may not result in linearity. The goal for *structural* methods is to study the model that can mostly reflect the potential formation mechanisms of dominance hierarchy.

One popular functional model is the I&SI method of Vries (1998). The I&SI method is a matrix-reordering method that identifies ordinal rankings of individuals that are most consistent with a linear hierarchy, by iteratively minimizing two criteria: the number of inconsistencies (I) and then, conditionally, the total strength of the inconsistencies (SI) without increasing I . The number of inconsistencies (I) is the number of pairs in which the lower-ranked individual wins more frequently than the higher-ranked individual in a given *win/loss* matrix, \tilde{W} ,

$$I = \sum_{i>j} \mathbb{1}_{\{\tilde{W}_{ij}>\tilde{W}_{ji}\}},$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function. The matrix \tilde{W} is generated by reordering the original win/loss matrix W according to a ranking of the individuals. The *strength* of a single inconsistency is the absolute rank difference of the inconsistent pair. Then, the total strength of the inconsistencies (SI) is the sum of strengths of all inconsistencies in \tilde{W} ,

$$SI = \sum_{i>j} |i - j| \mathbb{1}_{\{\tilde{W}_{ij}>\tilde{W}_{ji}\}}.$$

An example is shown in Figure 3.1. The original *win/loss* matrix in the example is W , which corresponds to $I = 3$ and $SI = 7$. According to the I&SI ranking method, the matrix is reordered to yield \tilde{W} , the rightmost matrix in Figure 3.1, in which $I = 1$ and $SI = 3$. Intuitively, the I&SI method finds the order of the rankings that is most consistent with a linear hierarchy. Although

such a perfect linear hierarchy usually does not exist, I&SI aims to find a ranking where any inconsistencies take place between individuals that are close in rank. In other words, the I&SI method is most likely to allow for inconsistent dyads near the diagonal.

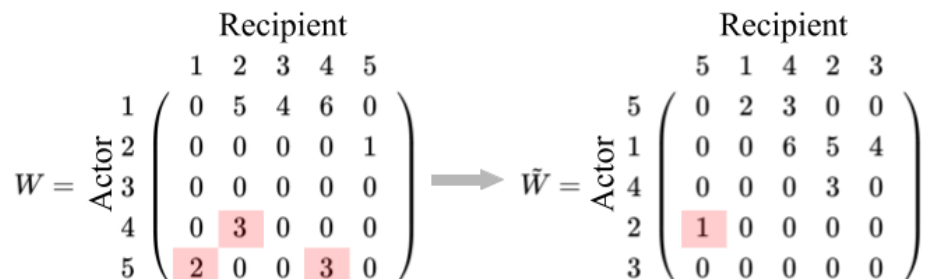


Figure 3.1: An example of a *win/loss* matrix and the corresponding reordered matrix according to the I&SI method. The entries shaded in red in the matrix are the *inconsistencies*, where the lower-ranked individual wins more frequently than the higher-ranked individual.

This method suffers from the problem that the algorithm is not guaranteed to converge to a unique optimal solution (Vries and Appleby, 2000). In particular, when there is a tie in the number of wins/losses ($W_{ij} = W_{ji} > 0$) or an unknown relationship (i.e., where there is little information, $W_{ij} = W_{ji} = 0$), the result highly depends on the choice of rules for assigning rankings. Another reason for the divergence is the method’s reliance on only an asymmetric relationship between the number of wins and losses, instead of the absolute difference. Such a simplified binary dominance measure ignores important information in the data – the total number of fights. It is often observed that the distribution of dominance power exhibits a high discrepancy (Chase, Tovey, et al., 2002), such as when highly ranked animals win a larger number of fights against intermediately ranked animals than these intermediate animals win against lowly ranked animals. This is seen in a *win/loss* matrix with large variation in the values of W_{ij} , but an ordinal ranking from a binary dominance measure is not discriminative enough to demonstrate that. Williamson et al. (2016) provide an analysis of monopolization of the most dominant mouse in each cohort, which suggests the necessity for considering a real-valued score instead of the ordinal rank.

Winner-loser models are an important family of structural methods that aim to explain the formation of linear dominance hierarchy (Lindquist and Chase, 2009). Commonly, the models

assume an innate power parameter for each individual i , denoted as F_i (some models may assume a time-variant version, i.e. $F_i(t)$) (Bonabeau et al., 1999; Dugatkin, 1997; Hemelrijk, 2000). Although different models have their own specific formulations, common components they share are: an interaction probability and a dominance probability. Both are functions of innate power. The mathematical formulation of the models will not be discussed here, but some assumptions used in these model are of interest. One essential idea is the *winner effect*, the phenomenon in which an animal that has experienced previous wins will continue to win future aggressive interactions with increased probability. Although the extent and effectiveness of these winner effects remains unclear, evidence from experiments show that they exist and vary in different groups and species (Dugatkin, 1997; Dugatkin and Earley, 2003; Hsu and Wolf, 1999). Experimental evidence also shows patterns that are challenging to capture through winner-loser models, such as *bursting* and *pair-flips*. Bursting means that higher-rank animals often exhibit successive fighting of lower-rank ones in an extended period of time. Pair-flips describe the situation when a pair of animals exchange the direction of their aggressive acts before a stable dominance relationship is established. A potential model for learning a latent hierarchy should therefore be able to incorporate these characteristics. In Section 3.4 we describe several existing structural and functional models in detail, which we use for comparison with our proposed methods.

3.2.1 Issues with conventional approaches

In summary, although the current methods have led to important insights on social structures among animals (So et al., 2015; Williamson et al., 2016; Hobson et al., 2021), they suffer from several issues that prevent them from being of utility for today’s increasingly available data with temporal information.

For functional methods, the use of an ordinal ranking alone may not be informative enough to describe the unequal distribution of dominance power which is commonly observed. The observed win/loss matrices are a noisy realisation of the true underlying dominance ranking among animals. Algorithms that attempt to directly derive the ranking by rearranging W can therefore be unstable

and unable to account for even small deviations from expected behaviour. In addition, uncertainty measures are not available for the dominance ranking produced.

Structural models for such data provide much potential to understand the underlying drivers of animal behaviour. However, methods to evaluate these models have been limited, as seen in Lindquist and Chase (2009). These models are generally not built under the statistical framework of generative models. They fall short of providing a probabilistic connection between the observed social interaction timestamps and the underlying dominance hierarchies, and as such are unsuited to rationalize noisy patterns against the subject-matter assumptions of these models. As a result, a more systematic solution is needed for modeling the temporal dynamics of the dominance hierarchy, instead of relying on empirical scores. The timestamps of interactions among a group contain information about how the particular hierarchy formation and social information of this hierarchy are associated with social interaction patterns over time. This framework can provide better insights into the questions of Williamson et al. (2016) such as agnostic interactions in uncommon directions. The utilisation of a probabilistic generative model for latent ranking further provides the potential to rigorously assess model fit and help formalise scientific hypotheses. In the next section we will develop generative point process network models for this data, which we then compare with these existing methods in Section 3.4.

3.3 Latent ranking structured network point process models

In this section, we propose a series of probabilistic generative models to address common issues with conventional approaches, as discussed in the previous section. Inspired by theories on social hierarchy among group-living animals, there are various properties that we want to take into account when constructing these models: inconsistencies lying between the interactions and rankings, the time-evolving nature of the interaction dynamic, the winner effect, bursting and pair-flips phenomena. For modeling the observed timestamps of social interactions, our models address three contributing mechanisms: the unobserved social hierarchy, the developmental influence from the animals' social information about the social hierarchy, and temporal dependence on historical

interactions.

We propose three network point process models based on latent (structural) rankings. We motivate the development of each of these models in turn by examining the properties each model fails to capture in one cohort of mice interaction data (described in Section 3.4.4), using the inference procedure described in Section 3.4.2.

Animal aggressive interaction data is essentially network data, where the senders are the winners of the fights and the receivers are the losers. To consider the necessary information lying in the timestamps of interactions, we utilise point process models on networks (introduced in Chapter 2.6) to describe the aggressive events between animals.

3.3.1 Latent ranking structured models for network point processes

Note that in this chapter to aid clarity we use the notation $\lambda^{i,j}$ to denote a Point Process from Node i to Node j , where elsewhere we use λ_{ij} .

Motivated first by the *winner effect* reviewed in Section 3.2, we model the conditional intensity of directed winning interactions between a given node pair as a function of their event (winning) history. Although experimental observations cannot explicitly verify the extent or persistence of influence of historical events, the intensity formulation in the Hawkes process (2.3) can help us model this *winner effect* flexibly. In a Hawkes process, α describes the extent to which previous wins influence the tendency to engage in a new fight. β represents the persistence – how fast this effect decays over time. A large β means that the winner effect decays quickly, and only the most recent wins influence the tendency to engage in aggressive interactions at the present time.

For a directed pair (i, j) , the Hawkes process intensity is

$$\lambda^{i,j}(t) = \lambda_1^{i,j} + \alpha^{i,j} \sum_k \exp(-\beta^{i,j}(t - t_k^{i,j})),$$

where $\lambda_1^{i,j}$, $\alpha^{i,j}$ and $\beta^{i,j}$ are pair-wise parameters in the Hawkes process. For all pairs, we will introduce structure in these pair-wise parameters below by assuming a latent rank variable, $f_i \in$

$[0, 1]$, $i = 1, 2, \dots, N$. This is similar to the latent characteristic concept used in the winner-loser models (Lindquist and Chase, 2009) and the latent rank in the aggregate-ranking model (De Bacco et al., 2018) reviewed in Section 3.4.1. The latent rank variable essentially embeds each individual in a one-dimensional unobserved ranking space. We constrain the pair-wise intensity function by bounding the latent rank in order to avoid issues with model identifiability. This latent rank variable is a powerful feature of our model as it allows us to incorporate various model assumptions on how the latent dominance hierarchy and historical events (i.e., social information) influence social interaction dynamics by specifying particular forms of the parameters $\lambda_1^{i,j}$, $\alpha^{i,j}$ and $\beta^{i,j}$ in the intensity function, as we will discuss in Section 3.3.1, 3.3.1 and 3.3.1.

Cohort Hawkes Process (C-HP) Model

In the first model, we assume a baseline intensity, λ_1 , and that the rate of decay for historical events, β , is constant across pairs. We structure the impact of historical events on each pair as a function of the pair's latent ranks f_i, f_j and parameters η , i.e. $\alpha^{i,j} := g_\eta(f_i, f_j)$. Inspired by the inconsistency and strength of inconsistency concepts in the I&SI method, we expect that the function $g_\eta(f_i, f_j)$ satisfies the following: (1) $g_\eta(f_i, f_j) > g_\eta(f_j, f_i)$ when $f_i > f_j$; (2) $g_\eta(f_i, f_j)$ is a decreasing function of $|f_i - f_j|$ when $f_i - f_j < 0$. Hence, we consider,

$$g_\eta(f_i, f_j) := \eta_1 f_i f_j \exp(-\eta_2 |f_i - f_j|) \text{logistic}(\eta_3 (f_i - f_j)),$$

where $\eta := (\eta_1, \eta_2, \eta_3)$.

Each component here is motivated by empirical evidence of observed behaviours seen in animal data. In particular, we describe each term in this excitation function in detail:

- $\eta_1 f_i f_j$ encodes the overall tendency of animals to continue fighting as a function of the product of their latent rankings. This results in pairs of higher ranked individuals being more likely to continue fighting. For higher ranked animals there is added incentive to clearly establish their dominance over similarly ranked animals, hence more repeated interactions.

Meanwhile, for a pair of lower ranked animals, there is often less incentive to move from (say) the lowest rank of the hierarchy to the second lowest. This behaviour is observed empirically in mice, with the higher ranked animals being most active throughout. Modeling the excitation parameter with the product of latent rankings allows us to capture this variation across pairs.

- $\exp(-\eta_2|f_i - f_j|)$ captures important properties found in existing methods within our model. This component gives an excitation parameter of the Hawkes process that is a decreasing function of $|f_i - f_j|$ when $f_i < f_j$. With this condition, this means that a weaker animal is less likely to continue winning fights against a stronger animal, with the likelihood of these events decreasing as the discrepancy between their latent rankings increases. This is motivated by the strength of inconsistencies component of the I&SI method and agrees with observed behaviour.
- $\text{logistic}(\eta_3(f_i - f_j))$ ensures that $g_\eta(f_i, f_j) > g_\eta(f_j, f_i)$ when $f_i > f_j$. This condition means that wins are more often from a dominant animal to a submissive one. Again, this aligns with the inconsistency concept of the I&SI method while being a property observed in real data.

Figure 3.2-(a) shows the contour plot of $g_\eta(f_i, f_j)$, with $\eta_1 = 4.46$, $\eta_2 = 0.18$, and $\eta_3 = 1.46$, which are estimated values from real data analysed in Section 3.4.4. Note that the x -axis in Figure 3.2-(a) is decreasing from left to right, in order to be consistent with the arrangement of a win/loss matrix, where the interactions between the most dominant pairs are displayed in the top left. We notice that the function takes higher values when $f_i > f_j$ (upper right triangle of Figure 3.2-(a)), compared to values when $f_i < f_j$ (lower left triangle of Figure 3.2-(a)). This ensures that winning interactions are directed more frequently from a dominant individual towards a submissive individual, mirroring the *inconsistency* concept in I&SI method. The contour plot also shows that the form of this function agrees with the idea of minimizing the total strength of the inconsistencies in the I&SI method: it has a smaller value when $f_i < f_j$ and $|f_i - f_j|$ is larger (moving from the diagonal to lower left triangle of Figure 3.2-(a)).

Now, the intensity in the C-HP model is

$$\lambda^{i,j}(t) = \lambda_1 + g_\eta(f_i, f_j) \sum_k \exp(-\beta(t - t_k^{i,j})).$$

Figure 3.2-(a) shows the contour plot of $g_\eta(f_i, f_j)$, with $\eta_1 = 4.46$, $\eta_2 = 0.18$, and $\eta_3 = 1.46$, which are estimated values from real data analysed in Section 3.4.4. Note that the x -axis in Figure 3.2-(a) is decreasing from left to right, in order to be consistent with the arrangement of a win/loss matrix, where the interactions between the most dominant pairs are displayed in the top left. We notice that the function takes higher values when $f_i > f_j$ (upper right triangle of Figure 3.2-(a)), compared to values when $f_i < f_j$ (lower left triangle of Figure 3.2-(a)). This ensures that aggressive behaviours are directed more frequently from a dominant individual towards a submissive individual, mirroring the *inconsistency* concept in I&SI method. The contour plot also shows that the form of this function agrees with the idea of minimizing the total strength of the inconsistencies in the I&SI method: it has a smaller value when $f_i < f_j$ and $|f_i - f_j|$ is larger (moving from the diagonal to lower left triangle of Figure 3.2-(a)).

Now, the intensity in the C-HP model is

$$\lambda^{i,j}(t) = \lambda_1 + g_\eta(f_i, f_j) \sum_k \exp(-\beta(t - t_k^{i,j})).$$

To assess the goodness-of-fit of point process models, according to the time rescaling theorem (Brown et al., 2002), we can test whether the *rescaled-inter-event* times $\{\Lambda_m := \int_{t_{m-1}}^{t_m} \lambda(s) ds\}_{m=1}^M$, are independently distributed following an exponential distribution with rate 1. We fit this model to data corresponding to interactions between a group of 12 mice, using the inference procedure of Section 3.4.2 We describe this data in more detail in Section 3.4.4. For each pair (i, j) , we conduct a Kolmogorov-Smirnov test on the *rescaled-inter-event* times $\{\Lambda_m^{i,j} := \int_{t_{m-1}^{i,j}}^{t_m^{i,j}} \lambda^{i,j}(s) ds\}_{m=1}^{M^{i,j}}$ and show the test statistics result in Figure 3.2-(b). The background colour indicates the value of the K-S statistics. This indicates good model fit for the C-HP model for the highly ranked animals (the top three rows of (Figure 3.2-(b))). The values of these K-S statistics increases, and there is

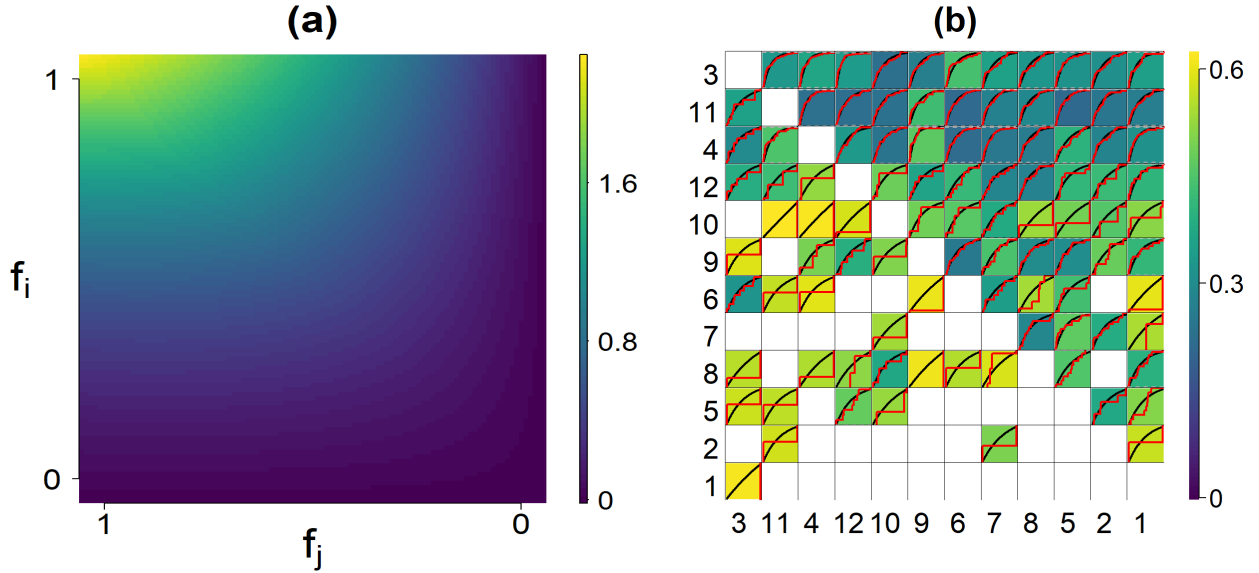


Figure 3.2: (a) Contour plot for $\alpha^{i,j} := g_\eta(f_i, f_j)$ where $f_i, f_j \in [0, 1]$. (b) Matrix of K-S statistics after fitting the C-HP model to the real data (reordered by I&SI ranking). The rows and columns of this matrix correspond to senders and receivers of an agonistic behaviour, respectively. Color shading reflects the values of the K-S test statistics. Red lines are empirical cumulative distribution functions of *rescaled-inter-event* times and black lines are cumulative distribution functions of exponential random variable with rate 1.

evidence of a lack of fit as we move from the top rows of Figure 3.2-(b), particularly in the lower left diagonal. This is unsurprising as a majority of all interactions in this cohort are instigated by the 3rd animal (first row). As such, in the C-HP model, these interactions lead to a much larger value of λ_1 then would be suited to describe interactions between other nodes. This suggests that this model does not adequately address individual baseline event intensities when the number of interactions between pairs can vary widely. We introduce a correction to account for this next.

Cohort degree-corrected Hawkes process (C-DCHP)

The cohort Hawkes process model (C-HP) model assumes a constant baseline rate λ_1 and is incapable of modeling the degree heterogeneity of the observed nodes. However, it can be observed from Figure 3.2-(b) that this model tends to consistently fit poorly for pairs which include certain individuals, for example those pairs in which the sender is individual 10 or the receiver is

individual 11. To address this issue, we extend the C-HP model, allowing varying baseline intensity rates across pairs. We accommodate degree heterogeneity in the pairwise baseline rate $\lambda_1^{i,j}$ by introducing a set of non-negative out-degree-correction parameters γ_i and in-degree-correction parameters ζ_j , $i, j = 1, 2, \dots, N$. With the baseline rate defined as $\lambda_1^{i,j} = \gamma_i + \zeta_j$, we have the intensity function as

$$\lambda^{i,j}(t) = \gamma_i + \zeta_j + g_\eta(f_i, f_j) \sum_k \exp(-\beta(t - t_k^{i,j})).$$

This model introduces degree-correction parameters in the baseline rate of the C-HP model, hence, we refer to it as the cohort degree-corrected Hawkes process model (C-DCHP).

Figure 3.3(a) shows the baseline intensity matrix after fitting this model to the same cohort in Figure 3.2-(b).

These estimates suggest that a model which allows more flexible intensities may indeed be needed to capture the heterogeneity in behaviour across individuals. We can see that this degree correction successfully adjusts for less active actors or recipients. For example, consider individuals 3 and 6; as shown in Figure 3.3, their estimated $\lambda_1^{i,j}$ s are relatively large indicating that both individuals are more active in terms of interaction frequency; individual 6 tends not to initiate many fights and individual 3 tends not to be the recipient of many fights. Importantly, these differences in activity level are individual-level attributes and require separate consideration when we are interested in learning about dominance hierarchy from dyad-level agonistic interaction data. The resulting inferred ranks from the C-DCHP model for such individuals are more consistent with the ranking obtained from existing methods, compared to the inferred ranks from the C-HP model. Similarly, to compare the fit of these models, we can look at the Pearson residuals (Wu, Smith, et al., 2021) from fitting each model to this data. We see, in Figure 3.3(b) that there is significant structure in the residuals after fitting the C-HP model, with large positive residuals for all interactions from the top ranked animal. Fitting the C-DCHP model removes much of the structure in these residuals and much better captures the heterogeneous nature of these pairwise interactions.

However, the C-DCHP model consistently assigns a very large out-degree parameter value to the most dominant individual, as shown in Figure 3.3(a) and seen for all real data examples we consider. This complicates inference for the latent ranks of high-ranked individuals and leads to poor model performance for any interactions not involving this individual. In observed data, these interactions excluding this dominant individual more often exhibit more sporadic behaviour, with long periods where no events are observed, a feature we consider in the following model.

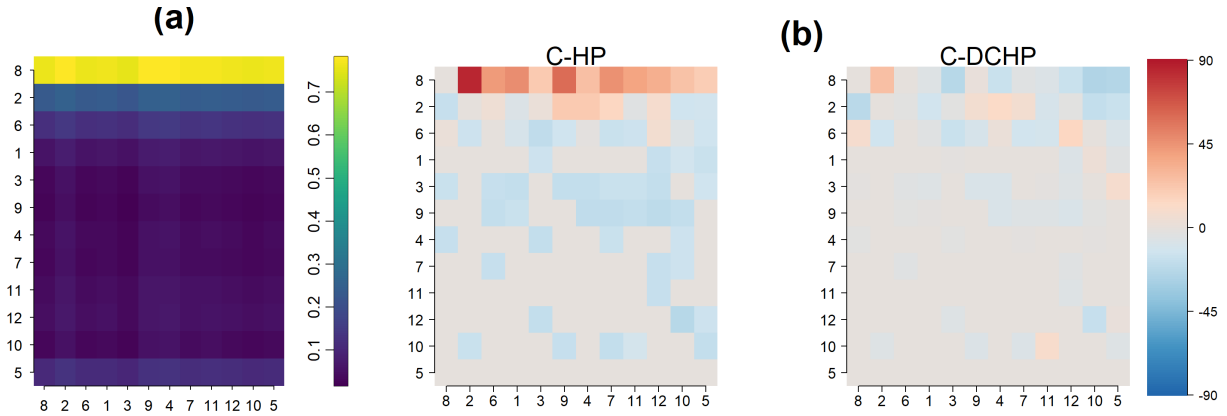


Figure 3.3: (a) Matrix of baseline rates $\lambda_1^{i,j}$ (reordered by I& SI rankings). These degree-corrected baseline rates allow for a more flexible node level model, clearly seen in the top row (a mouse which is involved in starting a large number of fights) and the bottom row (a mouse which does not start any fights but is often fought). (b) Pearson residuals for the C-HP and C-DCHP models.

Cohort Markov-modulated Hawkes Process (C-MMHP)

In Wu, Ward, et al. (2022), the Markov-modulated Hawkes Process (MMHP) is proposed to model general sequences of sporadic and bursty event occurrences. The model utilises a latent two-state continuous-time Markov chain (CTMC) $Z(t)$ to better describe event dynamics. In state 1 (the *active* state), events occur according to a Hawkes process, while in state 0 (the *inactive* state), according to a homogeneous Poisson process. The transition of $Z(t)$ is modelled through an infinitesimal generator matrix with parameters $\{q_1, q_0\}$, with the first row corresponding to

transitions from the active state, such that,

$$Q = \begin{bmatrix} -q_1 & q_1 \\ q_0 & -q_0 \end{bmatrix}. \quad (3.1)$$

Hence, for one MMHP, the conditional intensity function given the latent Markov process $Z(t)$, history events $\mathcal{H}(t)$ and parameter set $\Theta = \{\lambda_0, \lambda_1, \alpha, \beta, q_1, q_0\}$ is

$$\lambda(t|Z(t), \mathcal{H}(t), \Theta) = \begin{cases} \lambda_0, & \text{when } Z(t) = 0, \\ \lambda_1 + \alpha \sum_k \exp(-\beta(t - t_k)), & \text{when } Z(t) = 1. \end{cases}$$

Implicitly, the intensity function has the form

$$\lambda_0 + (\lambda_1 - \lambda_0)Z(t) + \alpha Z(t) \sum_k \exp(-\beta(t - t_k)).$$

Thus, the latent process provides substantial flexibility in modeling the baseline rate as well as the extent of historical event influence.

We can readily extend this model to the network setting, where for directed wins between each pair (i, j) , the intensity follows,

$$\lambda^{i,j}(t|Z^{i,j}(t), \mathcal{H}^{i,j}(t), \Theta^{i,j}) = \begin{cases} \lambda_0^{i,j}, & \text{when } Z^{i,j}(t) = 0, \\ \lambda_1^{i,j} + \alpha^{i,j} \sum_k \exp(-\beta^{i,j}(t - t_k^{i,j})), & \text{when } Z^{i,j}(t) = 1. \end{cases} \quad (3.2)$$

Here $\Theta^{i,j} := \{\lambda_0^{i,j}, \lambda_1^{i,j}, \alpha^{i,j}, \beta^{i,j}, q_1^{i,j}, q_0^{i,j}\}$ is the parameter set for pair (i, j) . $q_1^{i,j}$ and $q_0^{i,j}$ are the instantaneous transition probabilities for the latent CTMC $Z^{i,j}(t)$ of a pair (i, j) . $Z^{i,j}(t)$ are independent across pairs. The transition probability $q_1^{i,j}$ ($q_0^{i,j}$) represents the probability that pair (i, j) transitions out of the active (inactive) state and is modelled as a function of the latent ranks, f_i, f_j . To understand the behaviour of these latent state transition parameters for each pair (i, j) , consider the stationary distribution of the latent CTMC, $Z^{(i,j)}(t)$. For an irreducible and recurrent

CTMC $Z(t)$ with infinitesimal generator as shown in (3.1), a stationary distribution π satisfies $\pi^T Q = 0$ (Yin and Zhang, 2012). Hence, for a pair (i, j) , the limiting behaviour of their latent state transitions dictates that they spend $\frac{q_0^{i,j}}{q_0^{i,j} + q_1^{i,j}}$ of their time in the active state, and all remaining time in the inactive state. With the hope that if i dominates j , i.e. $f_i > f_j$, the pair (i, j) will spend lots of time in the active state, we form the transition probabilities as,

$$q_1^{i,j} = \exp(-\eta_3 f_i)$$

$$q_0^{i,j} = \exp(-\eta_3 f_j).$$

Hence, when individual i is stronger than individual j , node i is more likely to start and continue winning against node j (i.e. large $q_0^{i,j}$ and small $q_1^{i,j}$) than to observe wins from j to i . This follows the *asymmetry* property of aggressive behaviour in group animals. The limiting distribution of time spent in state 1 is $\text{Logistic}(\eta_3(f_i - f_j))$.

Given the latent process between pair (i, j) , $Z^{i,j}(t)$, we assume that $\beta^{i,j}$ is a constant β across pairs. In a similar vein to the C-HP and C-DCHP models, we model the winner effect $\alpha^{i,j}$ as taking the form $\eta_1 f_i f_j \exp(-\eta_2 |f_i - f_j|)$. The third component of the excitation parameter in these first two models is now replaced by the latent state indicator $Z^{i,j}(t)$. As such, this equips the conditional intensity in the C-MMHP model with additional flexibility, alternating between a simpler homogeneous Poisson process and a self exciting Hawkes process, with the limiting distribution of the process aligning with the C-DCHP model. Markov modulation allows the self-exciting component of the intensity to dissipate at points during the observation period, a phenomenon that is often observed in animal behavioural studies where interactions can be sparse at certain times. In other words, this modulation is introduced to account for the often seen sporadic nature of interactions between animals, arising from initial exploratory aggressive interactions and later interactions which may determine the precise hierarchy, before potentially stabilising.

Like the C-DCHP, we again consider the degree correction described in the previous model

here, giving $\lambda_0^{i,j} = \gamma_i + \zeta_j$. $\lambda_1^{i,j}$ is defined by

$$\lambda_1^{i,j} = \lambda_0^{i,j} (1 + w_\lambda),$$

for a common $w_\lambda \geq 0$, to ensure that the base rate of the point process in the active state is greater than the inactive state. Hence, we have the intensity from i to j given by

$$\lambda^{i,j}(t) = \lambda_0^{i,j} + (\lambda_1^{i,j} - \lambda_0^{i,j}) Z^{i,j}(t) + \eta_1 f_i f_j \exp(-\eta_2 |f_i - f_j|) Z^{i,j}(t) \sum_k \exp(-\beta(t - t_k^{i,j})).$$

This proposed model therefore provides a potential mechanism to describe the establishment of a realised dominance structure, as an expression of a latent hierarchy, by incorporating each of the components of the C-DCHP model and allowing Markov modulation. This is a key step towards better understanding the true process behind such interactions, a key question highlighted by Williamson et al. (2016). The mechanism we propose here agrees with existing methods from the animal behaviour literature and captures properties commonly seen in data of this form. For example, bursty behaviour is commonly seen in aggressive and subordinate interactions (Lee, Fu, et al., 2019), which is readily incorporated into our model through the use of a Hawkes process. Through the parameterization of the Hawkes process, our proposed model is also designed to capture a linear hierarchy. This agrees with many of the existing methods in this area. However, these existing aggregate methods fail to adequately account for wins in unexpected directions, such as from lower ranked to higher ranked animals, which can and do occur. Interactions such as these are unsurprising, particularly when the animals are first placed together, as they fight to gain social information about the social hierarchy. These interactions may continue to occur also. It can be advantageous for animals to improve their position by beating similarly ranked animals, to better obtain limited resources and potentially dissuade lower ranked animals from being aggressive towards them. While this is only seen among strong mice, in other species it occurs throughout the hierarchy (Hobson et al., 2021).

We observe this only among the top ranked mice in our data, where the top ranked animal can

be defeated late in the observation period (Williamson et al., 2016). By equipping each directed interaction pair with a point process, there is always some likelihood that interactions will be observed between animals in an uncommon direction. Our C-MMHP model goes further than this. With a Markov Modulated Hawkes Process, we allow the likelihood for interactions to vary in time, alternating between a Poisson process and a Hawkes process. This additional flexibility is well suited for capturing interactions in an uncommon direction. In particular, interactions of this type are often sporadic, often occurring after long periods where there are no interactions between that pair or in that direction.

3.3.2 Model inference

Bayesian modeling. Throughout this chapter, we adopt a Bayesian framework for our model inference. Assuming a prior distribution for the model parameters and given a model likelihood, the posterior distribution for quantities of interest can help us calibrate the uncertainty in the model. This is an important aspect of our current research strategy. First, we need tools that can quantify uncertainty in rank inference. For example, in Williamson et al. (2016), the analysis shows that the *pair-flips* phenomenon exists in some cohorts, which means that the direction of aggressive interaction changed over time. In a Bayesian modeling framework, we can naturally capture this effect through uncertainty in the model parameters: we suspect that individuals that are involved in pair-flip phenomena should have larger posterior variances for their latent ranks. Second, in each cohort, there always exists some pairs that have few or no interactions across the observation time window. A Bayesian framework can achieve robust inference in such conditions, with the assistance of prior assumptions and by borrowing strength from the data of other pairs. In this chapter, we will use the Stan modeling language (Carpenter et al., 2017) to fit all models and to obtain posterior samples for model parameters. We describe further details of our inference procedure in Section 3.4.2.

3.4 Results

3.4.1 Comparison Models

Before analysing the performance of our models on both real and simulated data, we first describe in detail existing methods used to analyse dominance behaviour in animals, which we shall use for comparison of inferred rankings. As described above, these methods can be broadly classified as *functional* and *structural*.

Functional methods

Along with the I&SI method, So et al. (2015) and Williamson et al. (2016) use the Glicko rating system to calculate temporal changes in dominance scores of each animal in each cohort. This is a dynamic paired comparison system that calculates a temporal sequence of cardinal scores based on the history of dyadic wins and losses (Glickman, 1999). All individuals start with the same initial rating. After each observed fight between a pair, the winner (or the loser) gains (or loses) points according to a decreasing function of the difference between their previous scores. In this case, fighting between pairs whose scores differ a lot will not result in significant changes in the system. The calculation of the Glicko score depends on a predefined constant, which determines the volatility of the score changes. Since scores are computed after each fighting event, this method can capture the temporal dynamics of the dominance hierarchy, although it does not account for temporal components, such as the time between events. Williamson et al. (2016) also provides a clear visualization of the change in the dominance score based on this method, where the emergence and stabilization of the hierarchy can be easily deduced from the graph. However, this method is ad-hoc in the sense that there are no theoretical rules for researchers to choose important key aspects of this method, including the initial rating, the decreasing function for changing a pair's scores after an observed fight, or the constant controlling the volatility of score changes. Since the method focuses on summarizing the observations without any formal modeling, it can be hard to provide formal insights regarding the evolution of hierarchy dynamics. It is also not always

clear how to draw a conclusion about the hierarchy structure from the visualization of the rating system.

Structural methods

Lindquist and Chase (2009) apply winner-loser models to real experimental data of hens and show the lack of fit between these models and the data. However, this procedure is qualitative only, by comparing simulation results from the models with the real data. The probabilistic generative models we proposed in Section 3.3 are able to capture these important animal behaviour phenomena, including the *winner effect*, *bursting* and *pair-flips*. We also develop a corresponding statistical inference procedure which means that model-fitting can be assessed by rigorous statistical model diagnostics, rather than relying on simulations as in Lindquist and Chase (2009). We analyse our models using these diagnostics in Section 3.4.3 and Section 3.4.4.

A more recent structural model is De Bacco et al. (2018), which introduced a physics-inspired model to infer cardinal hierarchical rankings of individuals in directed networks. By assuming that individuals are more likely to interact with others of similar rank, they propose an optimization solution and a generative model to find real-valued ranks of individuals. For a pair (i, j) , with latent rank variables f_i and f_j , the aggregate-ranking model uses Poisson regression to model the aggregate counts between the pair over the entire observation period, denoted as $N_{i,j}$, as a function of the difference in their ranks. This is essentially the counting process evaluated at time T , $N^{i,j}(T)$, ignoring the exact event times. The only information used in the model is the existence and direction of the interactions in the network. We refer to this model as the *aggregate-ranking* model. Only using the aggregate counts of interactions makes it hard to address phenomena like the *winner effect*, *bursting* and *pair-flips* mentioned in Lindquist and Chase (2009). Event time data which records when the aggressive behaviours occur is highly detailed and contains information needed to describe the important phenomena mentioned earlier.

3.4.2 Model implementation

To perform inference for each of these models we perform Bayesian inference using the Stan programming language (Stan Development Team, 2020). We impose weakly informative priors for the model parameters where possible. In particular, for each model we use half- $N(0, 1)$ priors on η_1, η_2, η_3 and β , $U[0, 1]$ priors for f , and a half- $N(0, 1)$ prior for w_λ in the C-MMHP model, to ensure that the rate in the active state is greater than in the inactive state. For the degree-corrected models we place Laplace priors on γ and ζ , to account for nodes which do not win or lose any fights. Full details of the inference procedure to infer the latent states of the C-MMHP model are given in Wu, Ward, et al. (2022).

3.4.3 Synthetic results

We first wish to validate our proposed models using simulated data where we aim to recover the *ground truth* latent ranking vector. To compare the three proposed models, we simulate 50 independent C-MMHPs generating event winning times between 10 nodes with uniformly separated true latent ranks. By fitting the synthetic data with our proposed three models, we can obtain the inferred latent ranks as shown in Figure 3.4-(a). Inference using C-MMHP and the C-DCHP both recover the true latent ranks well, with the C-HP showing considerable bias for several nodes. It is unsurprising that there is little difference between the inferred rankings from the C-DCHP and C-MMHP model, given their similar limiting distribution. Figure 3.4-(b) shows an example of estimated intensity for one pair of individuals in one simulated process (from the top ranked to second ranked individual), indicating that the C-HP and C-DCHP models cannot capture the true intensity as well as the C-MMHP model. The C-HP incorrectly captures the true decay parameter while both the C-HP and C-DCHP model overestimate the intensity repeatedly for events which occurred in the inactive state, with both showing more over and underestimation, although the C-MMHP does incorrectly classify some events.

We also compare the inferred ranking from each of these models and the comparison models discussed previously with the known true ranking. We summarize this using the Spearman rank

correlation between the estimated ranking obtained from each method and the true ranking, as shown in Fig 3.5. We note that the C-MMHP model recovers the true ranking best, with the C-HP and C-DCHP models also performing well but showing more variability. As a structural method, the I&SI model recovers the true ranking reasonably well for simulations from the generative C-MMHP model. In this scenario a large proportion of fights occur in the active state of the simulated C-MMHP model and so agreement between the I&SI method and our proposed models is expected.

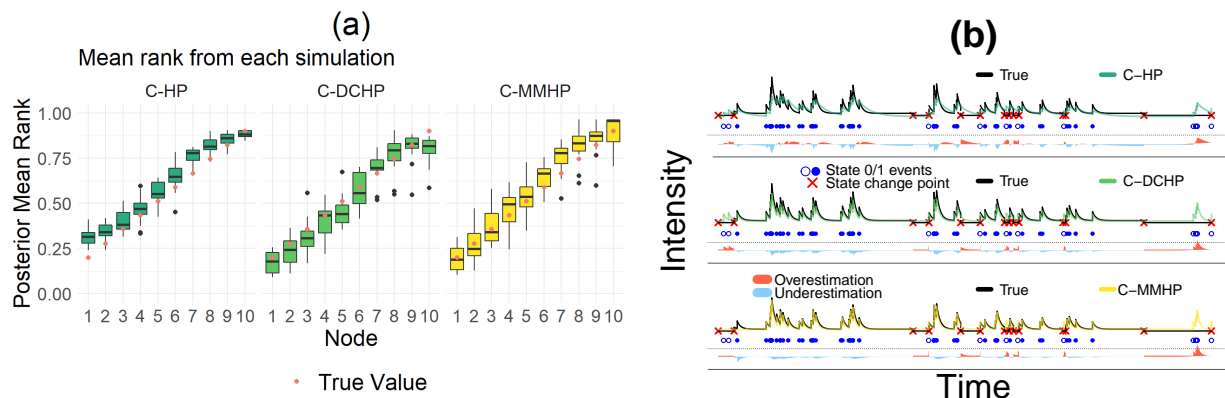


Figure 3.4: Simulation results from simulating 50 interaction datasets with common C-MMHP parameters. (a) Shows posterior inference of latent rank variable $f_i, i = 1, \dots, 10$ by C-HP, C-DCHP and C-MMHP. Each value is the posterior mean for f_i inferred from 50 independent simulations from a C-MMHP model with the same underlying parameters, with the true rank values overlaid in red. (b) Show the inferred intensity for one pair of individuals (top ranked to second highest ranked) in one simulation using three models. Here we fit each of the C-HP, C-DCHP and C-MMHP models to this data and plot the inferred intensity function for this pair. The events and the state they occurred in, along with the times the process changed state, are also shown in this plot. The red/blue shaded area underneath shows the magnitude of the error in the estimation of the intensity in terms of overestimation and underestimation.

3.4.4 Real data results

We next fit our models, C-HP, C-DCHP and C-MMHP to the ten mice cohorts studied by Williamson et al. (2016), which consisted of placing each cohort of twelve male mice in a large custom-built vivarium. Intensive behavioural observations were conducted for one to three hours per day during the dark cycle over twenty-one consecutive days. Trained observers recorded all occurrences of the behaviours (including fighting, chasing, mounting, subordinate posture and

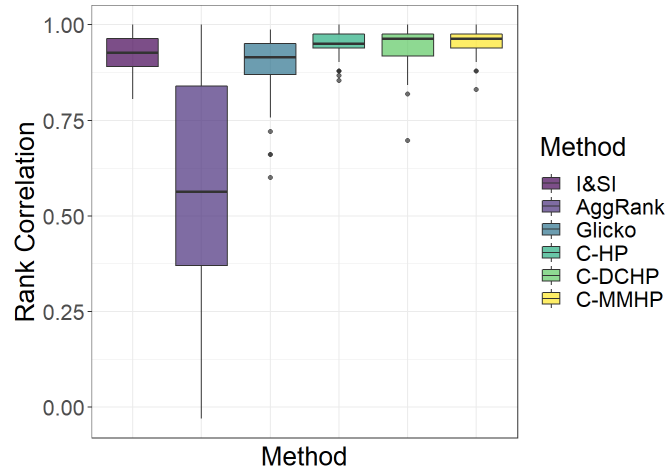


Figure 3.5: The Spearman rank correlation between the inferred ranking from each of these models, along with existing methods, and the known true ranking.

induced-flee). The details of each behavioural event are also recorded, identifying the actor which wins the interaction and the actor which loses the interaction, along with the timestamp and location.

Using various measurements in social hierarchy analysis and social network analysis, Williamson et al. (2016) demonstrates that these mice cohorts form significantly linear social dominance hierarchies. The work also examines the temporal changes in the mice social hierarchy and shows that in most of the ten cohorts, the dominance hierarchies emerge rapidly and become stable by the end of the second week. Although results of the quantitative analysis are thoroughly discussed and the patterns in the temporal dynamics are summarized qualitatively, there are still observations in some cohorts that disagree with the authors’ speculations, which have been unexplained in existing work but are naturally accounted for in our model. We have addressed how our model provides a potential mechanism for the connection between the establishment of a latent ranking of these animals with the observed social interactions and described how it captures interactions in an uncommon direction. We will verify that our model does describe this data well using multiple metrics, illustrating that a flexible probabilistic model incorporating these hypothesised processes can be constructed. We then also show how further information may be available from these models, such as information about the distribution of individuals’ dominance power. We also compare

the results from fitting the following existing models to this data: a dynamic social network in latent space model, and a Markov-modulated Hawkes process without incorporating any latent ranking structure between the nodes. We first briefly describe these models. Additional analysis results for each individual cohort are available in a supplemental file.

Dynamic social network in latent space model (DSNL)(Sarkar and Moore, 2006). This model is constructed for dynamic network data with binary links which is observed in discrete time steps. The model associates each node in the network with a latent space variable that can move in discrete time, and specifies that the move is Markovian. For node i at discrete time d , the latent variable is denoted as $f_i^{(d)}$. We tailor this model to our observed mice interaction data by changing the binary link assumption in the original model to allow for aggregate counts by using a Poisson link instead of a logistic link. We construct discrete time steps to be the ending time of each day in the observation time window, i.e. $t^{(d)}$. Hence, for each pair (i, j) , we have the count of their interactions during day d , $N_d^{i,j} := N^{i,j}(t^{(d)}) - N^{i,j}(t^{(d-1)})$, where $N^{i,j}(t)$ is the counting process for pair (i, j) evaluated at time t . Further details of this model will be omitted here.

Markov-modulated Hawkes process without network ranking structure (I-MMHP) (Wu, Ward, et al., 2022). In this model, we assume that the intensity function of (3.2) allows for different parameter values $\Theta^{i,j}$ across pairs. This means there is no structure between nodes and a latent ranking of the animals cannot be inferred. The independent structure of the parameters in this model is less constrained than our C-MMHP model, where we consider network structure between nodes to learn latent rankings.

Summary measures for evaluating model performance. Our real data analysis results will be summarized from several perspectives: (i) inference for the latent ranks and a summary of the properties captured by each of the different methods, prediction performance, both in terms of (ii) predicted events and (iii) predicted evolution of dynamics over time and (iv) additional insights available through the C-MMHP model, which may provide potential future research directions.

We compare the results of the C-HP, C-DCHP and C-MMHP models under the first three of these perspectives. Because the nature of the three other comparison models - aggregate-ranking, DSNL and I-MMHP - differs, they are fitted and compared from different perspectives. The aggregate-ranking model (and also the I&SI method we discussed previously) estimates a static ranking and will be discussed in terms of inference for the latent ranks only. The I-MMHP is a point process model and can be evaluated using the same point process methods as our latent ranking point process models. However, the I-MMHP cannot be used to infer a latent ranking. Finally, both the DSNL and I-MMHP models can perform prediction of events (or event counts) and can serve as comparison models in the prediction performance section.

Inference on latent rank

We fit the C-HP, C-DCHP, C-MMHP and aggregate-ranking models to our data of ten cohorts separately. Figure 3.6 shows the relationship between I&SI rank and posterior draws of latent ranks using our three models and aggregate-ranking model in two cohorts, Cohort 5 in Figure 3.6 (a) and Cohort 3 in Figure 3.6 (b). While we have ordered the animals by their estimated I&SI ranking, this is not to take it as some reference ranking but to highlight how our models capture a different ranking structure. These cohorts display the two types of common characteristics we observe across the 10 cohorts. Cohort 5 provides an example of general behaviour seen in 7 of the 10 cohorts studied here. In this example, there is general agreement between the rankings inferred from each of the Hawkes models, with the C-MMHP model displaying less uncertainty than the alternative models. The C-MMHP model identifies 3 approximate groups within the rankings, which is seen in several cohorts. As such, the bursty model assumption seems to agree with the dynamics evident here, with these bursty dynamics largely aligning with the power hierarchy. We also see animals such as # 7 and # 10 where the ranking from the C-MMHP model deviates from the I&SI ranking, as these animals are involved in relatively few fights and as such there are few periods of bursty fights. In this cohort all animals lose a similar number of fights, with the percentage of all fights which are classified as active by the Markov modulation being very high

for the whole cohort.

Different from the seven cohorts represented by cohort 5, Cohort 3 and another two cohorts exhibit another form of common behaviour. Here it is difficult to identify differences in the rankings for many of the animals. There is also weaker agreement between the C-MMHP model and the C-HP/C-DCHP models, while we also see disagreement between the I&SI ranking and the simplest of our models, the C-HP model. Interestingly, here all inactive wins originate from animal # 5, which is ranked significantly lower by the C-MMHP model than its I&SI ranking. Similarly, this animal has a large relative out degree estimate under both the C-DCHP and C-MMHP models. For these cohorts there are a large number of sporadic fights and a large percentage of wins attributed to a single animal, who seems to indiscriminately win fights against all other animals repeatedly. Although this behaviour is rewarded by traditional ranking methods, it is not clear that this should be an indication of dominance, with other animals perhaps unable to learn social information about the losers of these fights (Hobson, 2020). Our C-MMHP model places less emphasis on this behaviour, giving such animals a lower ranking than existing methods. Given that the Aggregate Ranking method and the I&SI method use the same win/loss matrix information, it is expected that they show strong agreement.

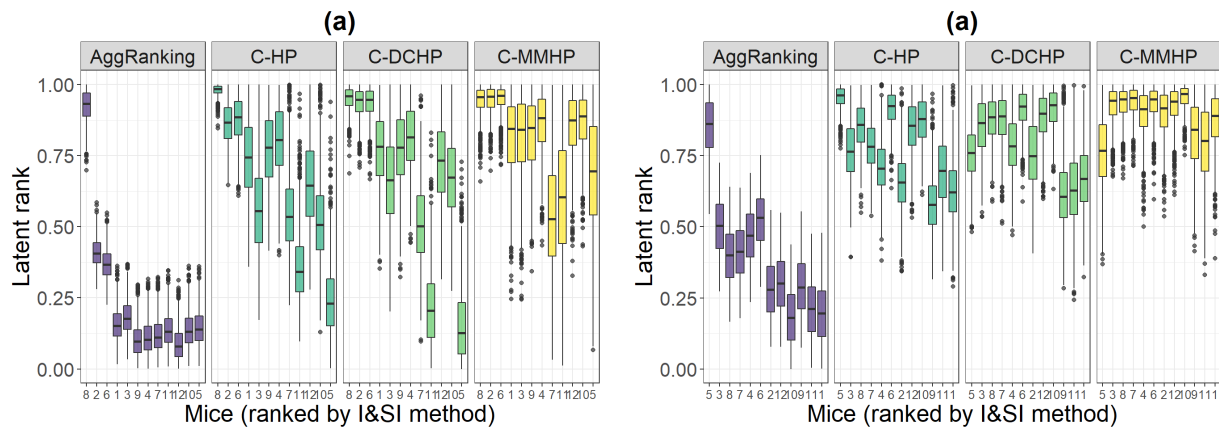


Figure 3.6: Real data fitting results. (a) Comparison of rank inference using different model with I&SI rank for Cohort 5. (b) Comparison of rank inference using different model with I&SI rank for Cohort 3.

Prediction

We can also use posterior predictive distributions to validate the models considered. For each model, we split the data into two time periods: (1) the first 15 days of data, $\mathcal{H}^{i,j}(t^{(15)})$, where $t^{(d)}$ is the ending observation time for the d -th day, which is used to estimate the model and (2) a prediction window from day 15 to day $t^{(d)}$, for $d = 16, \dots, 21$, the remaining observation period, which allows us to compare models across different prediction horizons. For each prediction horizon $t^{(d)}$, we generate a predicted point process separately over the time period $t^{(15)}$ to $t^{(d)}$, given each posterior draw of parameters and the historical events in the first 15 days. Hence, the predicted counting process $\hat{N}^{i,j}(t)$ is constructed by generating processes in each prediction period and adding these to the true process in the model-fitting period. For each prediction horizon and model, we generate 1000 posterior processes, corresponding to 1000 posterior draws from the posterior distribution for the model parameters. Following Sarkar and Moore (2006), we can also make predictions over these same time windows using the DSNL model.

Two aspects of the predictions are evaluated, the accuracy of predictions for the interaction counts and the prediction of the rankings.

For each point process model and for each different prediction horizon $d = 16, \dots, 21$, the number of total interactions for pair (i, j) during the prediction period can be estimated by $\bar{N}^{(i,j)}(t^{(d)}) - N^{(i,j)}(t^{(15)})$, where $\bar{N}^{(i,j)}(t^{(d)})$ is the average count of wins across 1000 posterior processes. We arrange the prediction counts in a matrix $\hat{A}^{(d)}$ such that each (i, j) entry is the predicted number of interactions for pair (i, j) from the end of the 15th day until the d th day. To quantify the accuracy of these predicted counts, we use the mean absolute error (MAE) of the difference between the estimated and real win/loss matrix $A^{(d)}$,

$$\frac{1}{n} \sum_{i,j} |\hat{A}_{ij}^{(d)} - A_{ij}^{(d)}|$$

The smaller the MAE, the closer the model's predictions of the interaction counts are to the observed data. Figure 3.7-(a) summarizes the result across all cohorts, by taking the median predicted

counts for each pair across each of 1000 posterior draws. The C-MMHP, C-DCHP and I-MMHP models provide the best predictions of interaction counts, with the smallest MAE across all prediction horizons, with C-MMHP slightly outperforming the other models.

We also infer a proxy predicted rank of individual i at prediction time $t^{(d)}$ by introducing the out-degree intensity

$$\hat{\lambda}_i(t^{(d)}) = \sum_j \hat{\lambda}^{i,j}(t^{(d)}).$$

The Glicko score ranking system serves as a benchmark for us to compare to, as it is a dynamic score. While this is not a true dynamic score, we can construct it for each of the models we consider here, allowing us to use it as a comparison across them, identifying agreement between each of these models and the evolution of the Glicko ranking. We compute the Spearman rank correlation of our inferred rank with the Glicko score at the end of the prediction day. Figure 3.7-(b) summarizes the result for all cohorts. The C-MMHP model closely aligns with the Glicko score, with rank correlation close to 1, while the unconstrained I-MMHP model also performs well in this scenario.

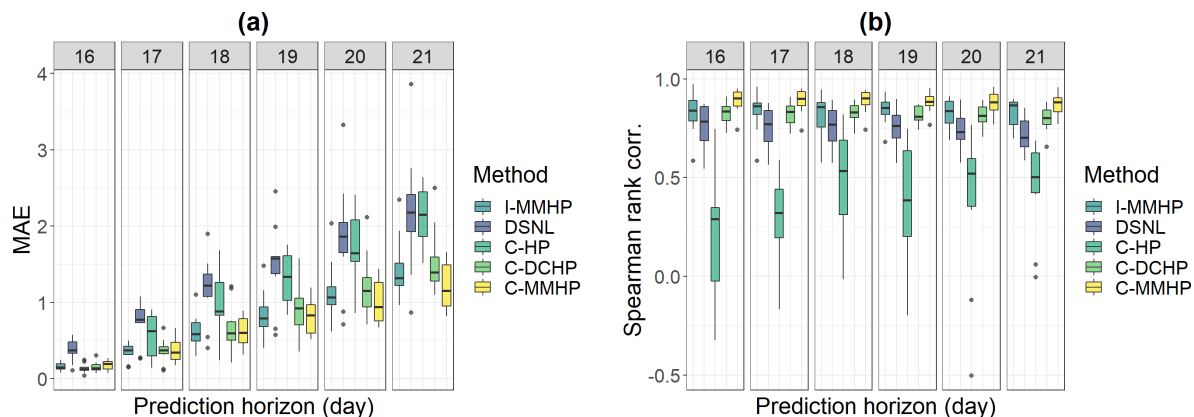


Figure 3.7: Prediction of events and rank. (a) shows the MAE of predicted error for all cohorts, using the median predicted count for each model for each cohort on each day. (b) Summarizes the Spearman rank correlation of predicted rank for all cohorts, where each cohort is predicted by the posterior mean of $\hat{\lambda}_i(t^{(d)})$.

Our posterior predictive processes can even be used to forecast the Glicko scores over future prediction windows, since we obtain the full event history from the generated process. In contrast,

the DSNL model can only provide day-level predictions, which we have evaluated previously. Figure 3.8-(a) shows the prediction of Glicko scores over days 19-21 when fitting the data in the first 18 days to the C-MMHP model. Our prediction bands can forecast temporal trends of Glicko ratings in the real data and provide an appropriate representation of the uncertainty in these predictions, which we illustrate in Figure 3.8-(a). These prediction bands correctly separate the rankings of most animals, particularly the highly ranked nodes, and capture the groups of rankings that seem to have formed for this cohort. Being able to predict rank evolution over time is not possible using existing methods and this could be of use in the experimental design of these studies to guide data collection, such as the ability to isolate mice who would be expected to rank similarly in the original group.

Additional insights from the C-MMHP model

Finally, we wish to highlight some additional insights which are available after fitting our C-MMHP model. Since our C-MMHP model can separate wins into active and inactive states, such separation can serve as a preprocessing step for the data. To illustrate this, we first fit the C-MMHP model to the data for one cohort and classify the wins into active and inactive states according to the estimated latent Markov process. The two types of interactions can then be fitted separately using other animal behaviour models. Wu, Ward, et al. (2022) shows that the wins in the active state more closely follow a linear hierarchy, as compared to the set of all wins or the set of inactive wins; this provides an explanation for the *pair-flips* phenomenon. During the active state, pairs are engaging in aggressive interactions and actively trying to navigate the social hierarchy, while in the inactive state, the wins are more or less random and lack specifically directed aggression seen in the active state. As an example, we fit the DSNL model to the set of overall events, active events and inactive events separately, and calculate the Spearman rank correlation between the latent ranks for each day as estimated by the DSNL model and the Glicko ratings at the end of each day. Figure 3.8-(b) shows these rank correlations on each day for the three types of wins. This suggests that the information contained in these wins varies over time, with the inactive state

showing a stronger correlation with the Glicko ranking in the start and middle of the observation period. Similarly, as Williamson et al. (2016) pointed out, the distribution of individual dominance power within animal groups is an important question. One way we can address this is by looking at the “out” degree estimates from our C-MMHP model. In particular, we see significant agreement between the out degree parameter estimates and whether Williamson et al. (2016) could identify the dominant animal after the first week. Where this animal could be clearly identified early on, the C-MMHP model resulted in a much larger out degree estimate for that animal than the rest of that cohort. This was not the case for the two cohorts where the dominant animal took longer to establish itself. This out degree parameter may therefore provide some insights into the distribution of individual dominance power within a given group, although further work is needed to further investigate this important question posed by Williamson et al. (2016). Interestingly, a similar connection is not seen with the less flexible C-DCHP model, which does not first classify events as active and inactive and base the inferred ranking on only the active events.

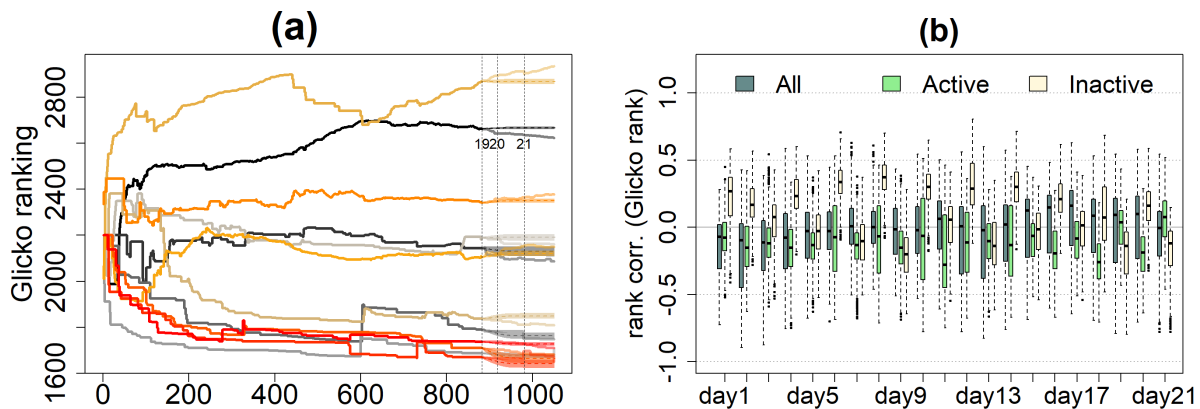


Figure 3.8: Further results on C-MMHP. (a) Glicko score ranking prediction of last three days using posterior draws, after fitting the first 18 days data in C-MMHP. True Glicko score ranking of all the time period is shown with the solid colored line, while the posterior prediction mean is in dashed line and one standard deviation is plotted in shaded color. The x-axis corresponds to the total number of interactions across the cohort. (b) Rank correlation between DSNL inferred latent rank and Glicko score ranking for each day in one cohort. Three colored bar indicated the performance of three inferred rankings conducted on the overall interactions, active and inactive respectively.

3.5 Discussion

In this chapter, we propose a series of statistical models that can uncover latent social dominance hierarchy among a group of animals from interaction win event times. Williamson et al., 2016 state that "How mice are able to recognize their social status relative to other mice and how this *recognition* facilitates hierarchy formation and maintenance remain unanswered." Like other models in the literature, our proposed model does not attempt to directly answer the question of *recognition* in these animals, or how it is related to the phenomena that are observed. We build a generative model utilising existing conjectures on behaviour patterns (e.g., self-excitation) that could be driven by recognition, such as animals modifying their behaviour based on their perceived position in the hierarchy. It is a statistical model that can incorporate such phenomena as a function of a latent ranking score, providing a natural modeling tool for such data. While existing work has also taken this approach, of modeling behaviour as a function of some latent ranking, we are the first to use such a model to model the exact event times directly. Model estimates and the overall goodness of fit provide evidence that our model is reasonable, including that it better describes such data compared to all existing models. That a model where the event times are driven by a latent hierarchy describes the data well provides support for the concept of recognition. Evidence for recognition, along with the other insights from our model discussed in the real data analysis allows better understanding of animal behaviour and helps illuminate potential future research directions. This makes our model an important tool for insight into the data generating process in these interactions, animal behaviour analysis and provide insights into important questions of recognition and its role in hierarchy formation.

To accomplish this, we formalise a point process model for continuous-time directed social network data. Three such models are developed: the cohort Hawkes process model (C-HP), the cohort degree-corrected Hawkes process model (C-DCHP) and the cohort Markov-modulated Hawkes process model (C-MMHP). The Hawkes process incorporates the winner effect and bursting patterns of aggressive behaviours, which are regularly observed in patterns of aggressive interactions

across animal species. The degree correction allows the model to better capture individual level heterogeneity that is commonly observed in data of this form. Finally, Markov-modulation accounts for pair-flip situations and allows for asymmetry in interactions between pairs of animals, by separating these interactions into active and inactive states. Performing inference for these models in the Bayesian paradigm allows us to accurately quantify the uncertainty in the inferred rankings and to better infer the ranking of nodes involved in few interactions, components that have been lacking in existing models for animal ranking.

The simulation study demonstrates that inferences from these models are reasonable and that the true ranking of nodes can be recovered. The mice cohort study serves as a real data example and demonstrates that the C-MMHP model performs best overall, in terms of providing insightful latent rank inference, prediction of both events and rank over time and potentially being of interest in generating future research directions. Although we do not have a ground truth for rankings in real data, we have described how our model complements existing ranking methods in the literature and the potential value of the additional inference available. That the event dynamics can be described by a latent ranking provides evidence that how these mice interact and explore their social structure is driven by their position in a hierarchy, a key idea in recognition.

We explore the dynamics of animal behaviour using this model motivated by observed and hypothesised phenomena in data of this form. They also highlight how the use of a new model may capture phenomena which existing models are not suited for. The results from our analysis of aggressive mice interactions provide insights on the agreement between model assumptions and the observed dynamics. This C-MMHP model can also be used to simulated future events, which could aid in designing studies of this form. Similarly, the state separation available in the C-MMHP model could lead to additional insights in conjunction with other models for animal behaviour.

Another key question in this area pointed out by Williamson et al., 2016 is the occurrence of agonistic interactions between pairs in unexpected directions. This is one key component of data of this form where existing models are inadequate. By utilising a point process, our proposed model is well suited to accounting for these interactions. Interactions which do not agree with

the final direction of the social hierarchy are unsurprising in animal data such as this. When these animals are first placed together they need to fight to determine their place in the hierarchy. As such we expect to see many inconsistencies initially, as animals are unsure of their place in the social structure and seek to identify it. Similarly, these animals are naturally trying to compete throughout the observed time period, and it is advantageous for them to improve their position, even if they are not ranked highly. This natural tendency to continue fighting is observed in real data, where we do occasionally see the top ranked animal being defeated at a later point (Williamson et al., 2016).

One potential extension of this model is to relax the assumption that the interaction dynamics are driven by a static latent ranking variable for each animal, which does not change in time. Previous work in the animal behavior literature has indicated that there is some influence between hierarchy formation and prior attributes of these animals (Chase, Tovey, et al., 2002; Chase and Lindquist, 2017). When these animals are placed together, a realized dominance hierarchy materializes over time, as the animals navigate their place in the social structure and gauge their own abilities (Hobson, 2020). As such, it is perhaps unclear whether additional information could be captured through the use of a dynamic latent ranking variable. To illustrate this we fit our C-MMHP model to the first two weeks of data and also the final two weeks of the 3 week observation period. We show the inferred latent ranking and their posterior distribution in each case for one such cohort below in Figure 3.9. The two sets of inferred latent rankings show a large amount of agreement, with little difference in the hierarchy, particularly among the top and bottom ranked animals. Most discrepancies occur among the animals ranked in the middle, where it can be hard to determine an exact ranking. This provides support for the choice of a static latent ranking in our proposed model, and suggests that key components of the rank order are established quickly, with only smaller departures from this in later observations. This is consistent with existing literature (Williamson et al., 2016), who show that the dominance hierarchy within a group forms quickly, with the dominant individual often evident after one week.

In the future, our model can be extended to incorporate the loser effect and bystander effect (Chase and Seitz, 2011) within the Hawkes process intensity function. The loser effect means

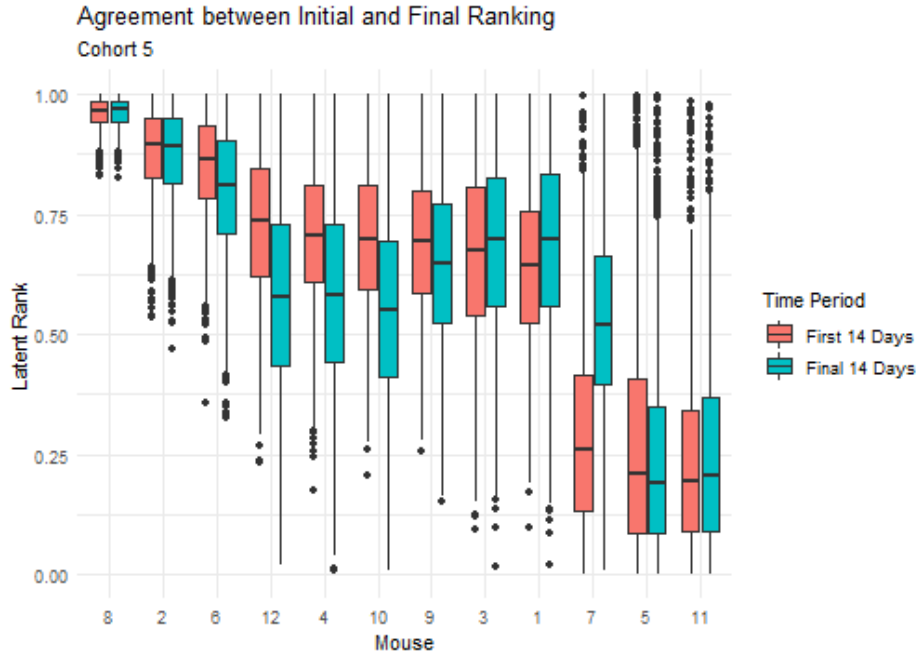


Figure 3.9: Boxplot of the posterior draws for the latent rank of animals in one cohort, fitting the C-MMHP model to both the first 14 days and last 14 days of observations separately. The mice are ordered by the rank from the first 14 days.

that an animal that has lost in earlier contests has an increased probability of losing subsequent contests with other individuals. The bystander effect describes the situation where an animal's behaviour might be influenced by observing an interaction or contest between two other animals. The extent of each effect can be estimated through a multivariate Hawkes process. The existence of such effects could be tested through the limiting distribution in Chen, Shojaie, et al. (2017). Alternatively, models which utilise reciprocating interactions, such as Blundell et al. (2012) could also be of interest. So et al. (2015) raises a question about the causal relationship between aggressive behaviour and gene expression. It is feasible to integrate these elements in our model by modelling the baseline intensities as a function of covariates that correspond to gene expression. We have also seen that certain global parameters in our models do not vary from Cohort to Cohort. As such, it would be of interest to design a hierarchical version of our model, borrowing strength from different datasets. Further work could also be done to make the degree estimates a function of the nodes latent rank, although there is no clear evidence from the literature as to the association

between the baseline activity level and the position in a realised hierarchy. This question therefore remains an important future problem.

Similarly, the model we have proposed here is a special case of a latent space model. Latent space models are an important tool in social network analysis and have been widely used in modelling both static network (De Bacco et al., 2018; Hoff, 2005; McCormick and Zheng, 2015) and dynamic network (Sarkar and Moore, 2006; Sewell and Chen, 2015) data. Although latent space models of discrete-time dynamic networks have been considered (Kim et al., 2018), along with continuous time dynamics across repeatedly observed matrices (Durante and Dunson, 2014), there has been little work in the context of continuous time events occurring on networks, and this remains an area for future research.

Chapter 4: Online Inference for Community Detection using Events on Networks

4.1 Introduction

An important task in statistical machine learning is to capture the structure present in large complex data occurring on networks. One common goal of many statistical network models is *community detection* (Zhao et al., 2012; Amini et al., 2013), which aims to uncover latent clusters of nodes in a network based on observed relationships between these nodes (Fortunato and Hric, 2016). However, many of these models assume that the edges, describing the relationship between these nodes, are simple, i.e., with interactions between nodes described by binary edges or weighted edges consisting of counts. In reality, for many real networks, activities between nodes occur as streams of interaction events which may evolve over time and exhibit non-stationary patterns. For example, social network data is commonly aggregated into binary edges describing whether there is a connection between two nodes, when in reality the true underlying interaction could have consisted of multiple messages or other interactions over a period of time. The binary edge might be constructed by considering if the number of such interactions is above an arbitrary cut-off. Aggregating these event streams and ignoring the time component to these interactions leads to a loss of information. Models which take advantage of the temporal dynamics of event streams therefore hold the potential to reveal richer latent structures behind these dynamic interactions (Matias, Rebafka, et al., 2018).

Example To illustrate the importance of utilising these event streams if they are available, we consider a simple example, similar to simulated data considered by Giorgi et al. (2018). In particular, we suppose data is generated from a block type model with $K = 2$ communities, where

the event times between nodes pairs are drawn from a temporal point process. The intensity of the process between these two communities changes over time in a simple way. In particular, we consider an intensity function from nodes in group k_1 to nodes in group k_2 of the form

$$\lambda_{k_1 k_2}(t) = \lambda_{k_1 k_2}^1 \mathbb{1}\{0 \leq t < T/3\} + \lambda_{k_1 k_2}^2 \mathbb{1}\{T/3 \leq t < 2T/3\} + \lambda_{k_1 k_2}^3 \mathbb{1}\{2T/3 \leq t < T\},$$

where the coefficient vectors $\lambda_{k_1 k_2} = (\lambda_{k_1 k_2}^1, \lambda_{k_1 k_2}^2, \lambda_{k_1 k_2}^3)$ for each block pair are of the form

$$\begin{pmatrix} \lambda_{11} \\ \lambda_{12} \\ \lambda_{21} \\ \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0.25 & 0.5 & 1 \\ 0.75 & 1 & 1 \\ 0.5 & 0.25 & 1 \\ 1 & 0.75 & 1 \end{pmatrix}.$$

The goal is to recover the communities these nodes belong to, given the events observed. Here, community detection methods which take account of the event times can readily capture the community structure, while methods which are designed for repeated observations of an adjacency matrix, and would need to aggregate this data, fail. In particular, we perform spectral clustering using the count matrix of total observed interactions and also by binning these interactions and constructing more flexible estimators (Pensky and Zhang, 2019). Both of these methods require aggregation of the data and as shown in Figure 4.1, are unable to recover community structure present (in terms of ARI), compared to the model we consider in this paper, which utilises exact event times.

Point processes are commonly used to model event streams, which can then be incorporated into network models to provide a community detection method which accounts for the dynamics of these event streams on the network. Notably, these models are able to characterize sporadic and bursty dynamics, which are ubiquitous in event streams on networks. Network models of this form have recently been developed, uncovering more expressive community structure. However, these methods suffer from the computational challenges associated with both network data and point

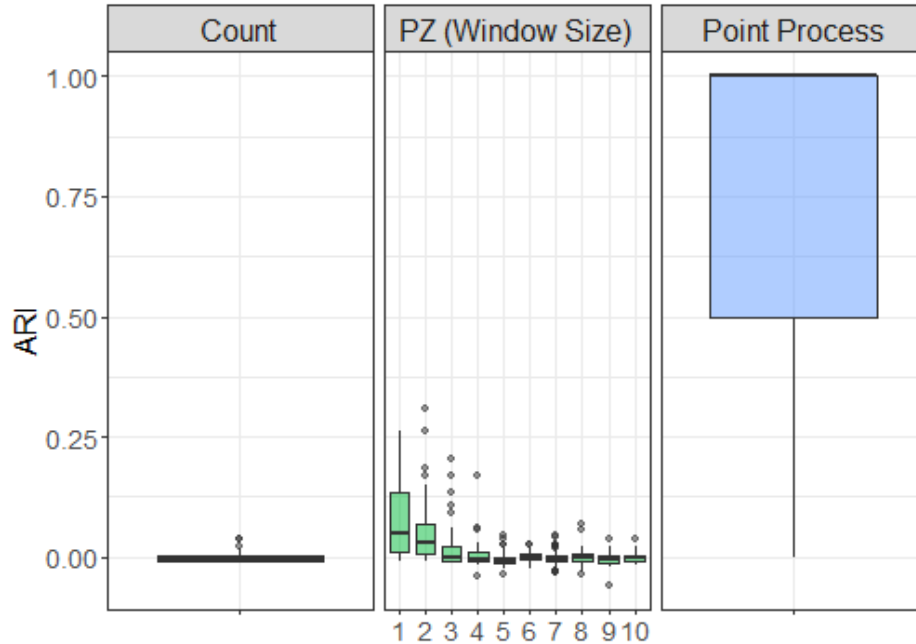


Figure 4.1: Community recovery in terms of Adjusted Rand Index (ARI) for events simulated from a point process block model. Aggregate methods which look at the overall count between nodes, or bin the data (PZ)(Pensky and Zhang, 2019) cannot recover the community structure. This is the case regardless of how we aggregate the data to form the PZ estimator (as given by the multiple box-plots). Modelling the exact event times through a point process can recover the true community structure well.

process methods, and it is computationally difficult to scale them to large networks. Moreover, to truly account for the streaming nature of edges, we would like to be able to perform community detection as events are observed on the network, updating our model in a flexible way. To do this, we propose an online variational inference framework and corresponding algorithms to learn the structure of these networks as interactions between nodes arrive as event streams.

We evaluate the performance of our proposed estimation scheme, comparing to non-online versions of our method. These experiments demonstrate comparable performance. We examine important implementation aspects of this online procedure, including monitoring the convergence of our estimators. This procedure performs well under various simulation settings, without the need to perform repeated expensive computations on the entire dataset. We investigate the empirical properties of our procedure, including the regret and online loss performance, comparing to expensive batch methods. Finally, the online procedure is compared to batch methods for the task

of link prediction on multiple real networks.

This chapter is organised as follows. In Section 4.2 we first formally define the required notation for modelling event streams using point processes and consider existing work which posits block type models of point processes to model event streams on networks. We also review existing results for online variational inference. In Section 4.3 we propose an online learning framework for models of this form. Section 4.4 outlines simulation studies comparing the performance of our procedure to more expensive batch methods. In Section 4.5 we implement our algorithm on multiple data sets of streaming events on networks, demonstrating comparable performance to non-online methods. Finally, in Section 4.6, summarise this work and briefly describe how this procedure could potentially be modified and applied in other contexts.

4.2 Online Streaming Data and Latent Cluster Assignment

Notation We use $N(t)$ to represent a general counting process and use $\lambda(t)$ to denote the corresponding intensity function. Here e and t are adopted to represent an event and a time stamp, respectively. Additionally, z is used for the latent class membership, while θ denotes the remaining “global” parameters of the model considered.

We first review the required framework of modelling event streaming data using point processes and describe previous work which has been done to incorporate such structure into existing network models.

Mathematically, streaming data can be described as $\{(e_1, t_1), \dots, (e_m, t_m), \dots, (e_{M^*}, t_{M^*})\}$, where e_m is the m th event and t_m is its corresponding time stamp. We observe M^* distinct events in some time window $[0, T]$. We have $e_m \in \mathcal{E}$ for $m = 1, \dots, M$ and $0 < t_1 < \dots < t_N < T$, where \mathcal{E} is the set of all possible different event types. Specifically, for event data on a network, we have $\mathcal{E} = \{(i, j) \in A \mid i, j \in [n]\}$ where (i, j) represents a directed event happening from node i to node j ; n is the size of the population, $[n] = \{1, \dots, n\}$ and A is the edge list, which encodes the network structure present. We use $|A|$ to denote the total number of directed interaction pairs of nodes in the network.

Had only the event times been observed, a natural way for modelling this type of streaming data is to use the machinery of counting processes. Under this framework, $N(t)$ is used to denote the counting process, the number of events observed up to time t . Along with this, the conditional intensity function is defined as

$$\lambda(t) = \lim_{dt \rightarrow 0} \frac{\mathbb{E}(N[t, t + dt] | \mathcal{H}(t))}{dt}, \quad (4.1)$$

where $N[t, t + dt)$ represents the number of events between time t and $t + dt$ and $\mathcal{H}(t)$ is the history filtration which is mathematically defined as $\sigma(\{N(s), s < t\})$ (Daley and Jones, 2003). The simplest counting process is the Poisson process, where the intensity function is constant in time, $\lambda(t) \equiv \lambda$. Naturally, this simple model is often insufficient to capture the time heterogeneity seen in event data, and processes which can vary in time are required. One such popular process incorporates *self-excitation*, where the intensity function is positively influenced by each historical event, with this influence decreasing as the length of time since that event increases. Among such processes, the Hawkes process has been widely used, including for modelling earthquake occurrences and financial data (Ogata, 1988; Hawkes, 2018).

Similarly, network models have been widely used to model social network data, describing the interactions (edges) between users (vertices/nodes) in a network. Network models consisting of binary or discrete edges between nodes are extensively studied in the statistical and machine learning literature. Perhaps the most widely used network model for binary networks is the stochastic block model. Stochastic block models assume that each node belongs to some latent cluster, with the probability of edges between nodes determined by their latent cluster assignment (Nowicki and Snijders, 2001).

When describing interactions between nodes in a network, it is often true that the underlying interactions are in fact observed in continuous time before then being aggregated into some discrete representation. For example, repeated interactions between nodes in a social network could be simply counted, with a binary link formed if the number of (directed) interactions is above some

threshold. One extension of these models for static networks that has been considered is to split the observations into multiple time windows with a static network constructed for each of these windows. In the context of messages on a social network, this would consist of constructing a static network based on the interactions between nodes in some time period (say, every week). Community detection methods have been developed for block models in this context (Pensky and Zhang, 2019). However, these methods still require compression of continuous time interactions into a static representation, which can fail to capture the true expressive dynamics between nodes. Similarly, the length of window used is subjective, and it is not clear in general how to choose the level of aggregation required. The direct modelling of repeated event streams on a network has not been as widely studied (Rossetti and Cazabet, 2018).

Recent extensions of stochastic block models have been used to model events on networks directly using point processes, the setting we consider here. This allows for community detection of nodes in a network which captures the temporal dynamics which describe events between nodes. Suppose that $\mathbf{z} = (z_1, \dots, z_n)$ is a vector representing the latent class memberships of n nodes in a network, where each node belongs to one of K possible classes. The latent classes are drawn from some vector π which gives the latent probability of each of the K classes. We assume that (directed) interactions between any two nodes in the network follow a point process, which has intensity $\lambda_{ij}(t)$. We impose a block model structure on these intensities, in that the intensity between two nodes is determined by the latent class of both the nodes. Given node i in latent class z_i and node j in latent class z_j then we have

$$\lambda_{ij}(t) = \lambda_{z_i z_j}(t).$$

This model was first considered by Matias, Rebafka, et al. (2018). In that setting, a block model was proposed where, conditional on the latent groups, interactions from any one node in the network to another follow an inhomogeneous Poisson process. The usual variational EM estimation procedure for binary networks was then extended to this setting, resulting in a variational semi-

parametric EM type algorithm. Given the current estimate of the cluster assignments, the conditional intensities are then estimated using a non-parametric M-step, consisting of either a histogram or kernel based estimate. A similar model has been proposed elsewhere (Miscouridou et al., 2018), where edge exchangeable models for binary graphs are extended to this setting. Here, the baseline of a Hawkes process encodes the affiliation of each node to the K latent communities, with a common exponential kernel for all interactions. Inference for this model is carried out using Markov chain Monte Carlo (Gilks et al., 1995).

While both these models are flexible and have been demonstrated to work well on real networks, they are both computationally intensive to fit. Each method requires multiple iterations over all events in the network to learn the community structure. Similarly, given the estimation procedures for these models, there is no immediate way to update these parameters in the context of streaming events, to readily incorporate the observation of new events. Given the continuous time nature of event streams we would like to be able to update our estimated community structure either in real time or, at least, without repeatedly using the entire event history. We would also like to avoid assuming that the total observation period is fixed. Below we provide a learning procedure for models of this form which avoids much of this computational burden and can more readily update the community structure given new observations.

We will consider point process block models of this form in this chapter. In particular, we will consider several possible formulations of the conditional intensity:

- **Block Homogeneous Poisson Process Model** The intensity function of block homogeneous Poisson process model is of the form

$$\lambda_{ij}(t) = B_{z_i z_j} \tag{4.2}$$

This intensity function only depends on individuals' latent community membership and does not depend on time.

- **Block Inhomogeneous Poisson Process Model** The intensity function of the block inhomogeneous

geneous Poisson process model is of the form

$$\lambda_{ij}(t) = \sum_h a_{z_i z_j}(h) f_h(t) \quad (4.3)$$

where $f_h(t) \in \mathcal{H}$ with \mathcal{H} being some functional space. This intensity function has the additive form, characterized by the linear combination of basis functions. Under this case, the intensity function depends only on the community assignment also, but these community wise intensity functions can vary in time.

- **Block Homogeneous Hawkes Process Model** The block homogeneous Hawkes is the extension of the original Hawkes model (Hawkes and Oakes, 1974). The intensity function is of the form

$$\lambda_{ij}(t) = \mu_{z_i z_j} + b_{z_i z_j} \int_0^t f(s) dN_{ij}(s), \quad (4.4)$$

where μ represents the baseline intensity, b represents the magnitude of the kernel function and f is the kernel function, which indicates the influence of previous events on the current intensity. A classical choice of f is $f(s) = \lambda \exp\{-\lambda s\}$ (Rizoïu et al., 2017), which we shall use throughout this paper, with a common λ across all nodes. Here, the parameters of this Hawkes process are determined by the community assignment, but the individual pair wise intensity function will depend on the history of events between that pair.

- **Block Inhomogeneous Hawkes Process Model** The intensity function of the block inhomogeneous Hawkes process model is of the form

$$\lambda_{ij}(t) = \mu_{z_i z_j}(t) + b_{z_i z_j} \int_0^t f(s) dN_{ij}(s), \quad (4.5)$$

where μ is no longer constant over time. Instead, $\mu_{kl}(t) = \sum_h a_{kl}(h) f_h(t)$ with $f_h(t) \in \mathcal{H}$ with \mathcal{H} being some functional space. That is, we assume the baseline function can be across

communities is characterized by the linear combination of certain basis function to capture different time patterns, while also depending on the event history for that node pair.

Online Variational Inference

In this chapter we consider online variational inference for estimating community detection in event data on networks. In the general formulation of online learning, data is observed sequentially in time with \mathcal{D}_m being the m -th such observation and there is a loss function $\ell(\mathcal{D}_m, \hat{\theta}_m)$ for a given parameter estimate $\hat{\theta}_m$. This estimate will be based on the past data $\mathcal{D}_{1:(m-1)} := \{\mathcal{D}_1, \dots, \mathcal{D}_{m-1}\}$. This loss function may vary depending on the inference procedure considered, but a natural choice is often the negative log likelihood, corresponding to online maximum likelihood estimation.

The aim of online inference is to find a parameter estimate which is close to the best overall estimate, had all the data been observed. We will denote such an estimate as θ^* which would minimize the generalization error $\mathcal{E}_*(\theta) = \mathbb{E}_{\mathcal{D} \sim P_*} \ell(\mathcal{D}, \theta)$, where P_* is the true data distribution. This quantity is unknown in practice and so interest instead lies in minimizing the cumulative error over time, $\sum_{i=1}^M \ell(\mathcal{D}_m, \hat{\theta}_m)$. Given this estimate, a commonly studied quantity is the *regret*, which is the difference between this cumulative error and the minimum cumulative error for a fixed estimate of the parameter, θ ,

$$\sum_{m=1}^M \ell(\mathcal{D}_m, \hat{\theta}_m) - \inf_{\theta \in \Theta} \sum_{m=1}^M \ell(\mathcal{D}_m, \theta).$$

Regret bounds can quantify the values of this quantity and have been obtained in certain settings (Shalev-Shwartz et al., 2012). Such bounds can also then be used to compute corresponding bounds on the generalization gap. Online learning has been considered for Bayesian inference, which is in some sense a natural setting for such a scheme. Chérif-Abdellatif et al. (2019) describe such a setting, where the “online” posterior can be written as

$$p_m^\eta(\theta) := \frac{1}{Z_m^\eta} \pi(\theta) e^{-\eta \sum_{i=1}^{m-1} \ell(\mathcal{D}_i, \theta)},$$

for some learning rate η , prior π and normalizing constant Z_m^η . In particular, if the loss function chosen is the log likelihood and $\eta = 1$ then this is exactly standard Bayesian inference, observing the data sequentially and updating the posterior. For $\eta < 1$ then this is tempered Bayesian inference (Alquier and Ridgway, 2020). Chérif-Abdellatif et al. (2019) extend this idea to variational inference, utilising gradient updates based on the loss for the parameter estimates. This is similar to the setting we propose here.

In particular, they consider three such formulations for gradient based updates to the variational approximation as data is observed in an online fashion. These differ in the objective function, which is composed of the gradient of the loss function and a KL term. In the first two cases, a regret bound can be derived for the corresponding updates, with one requiring the restriction that the variational family is mean field Gaussian. For the gradient update corresponding to natural gradient variational inference, a corresponding theoretical regret bound cannot be obtained. These results are investigated empirically on classical regression and classification problems, where there is no latent structure in the variational approximation.

4.3 An Online Learning Framework for Event Streams

Many methods in statistics and machine learning analyse large data in batches. This often involves processing large volumes of data at the same time and repeatedly, with long periods of latency. More recently, data streaming is widely used for real-time aggregation, filtering, and testing. This allows for real time analysis of data as it is collected and can be used to gain insights in a wide range of applications, such as social network (Bifet and Frank, 2010) and transit data (Moreira-Matias et al., 2013). Motivated by the aim of improved computational efficiency, we propose a scalable online learning method for the network point processes with latent block structure of Section 4.2.

4.3.1 Online Learning Algorithms for Network Point Processes

We denote by θ the model parameters we wish to learn and by $l(\theta)$ the objective function (the log-likelihood function in our setting). Let dT be a time window such that T , the total time for which the event stream is to be observed, can be subdivided into $M = T/dT$ time windows (we suppose T/dT is an integer without loss of generality). Following this subdivision into M time intervals, $l(\theta)$ can be rewritten as $l(\theta) = \sum_{m=1}^M l_m(\theta)$, where $l_m(\theta)$ is the objective corresponding to the m -th time window (in what follows, we use subscript m to denote a quantity computed in the m -th time window).

In a batch algorithm, the estimator $\hat{\theta}^b$ is defined as $\arg \max_{\theta} l(\theta)$, i.e. the best parameter estimate to achieve the maximum objective value. When $l(\theta)$ is taken as the log-likelihood function, $\hat{\theta}^b$ is also known as the maximum likelihood estimator (MLE). Unfortunately, such optimization may be computationally challenging when the data size becomes large and $l(\theta)$ contains latent discrete variables. Hence, we aim to construct an estimator $\hat{\theta}^o$ to approximate $\hat{\theta}^b$ with less computational burden, while also hopefully possessing the same properties as $\hat{\theta}^b$. To this end, we consider an online method for this optimization problem. The general scheme is described as follows.

Algorithm 1 General Online Optimization

Set initialization of $\theta^{(0)} = \theta_0$.
for $m = 1$ to M **do**
 Update θ by $\theta^{(m)} = \theta^{(m-1)} + \eta_m \frac{\partial l_m(\theta)}{\partial \theta}$.
end for
Output Set $\hat{\theta}^o = \theta^{(M)}$

However, under our setting, the general online scheme does not apply by noting that the true latent class label assignment is unknown. In other words, we need to integrate over all possible latent class configurations for computing the log-likelihood function, which is often intractable. In particular,

$$l(\theta) = \log \left\{ \sum_z \pi_z \exp(l(\theta|z)) \right\} = \log \left\{ \sum_z \pi_z \exp \left(\sum_{m=1}^M l_m(\theta|z) \right) \right\},$$

indicating that $l(\theta)$ can not be simply rewritten in the format of $l(\theta) = \sum_{m=1}^M l_m(\theta)$.

We use a variational approximation for the latent community assignments, which allows us to derive temporally estimates of the community structure, and the corresponding point process intensity functions. We take

$$q(z) := \prod_i q_i(z_i)$$

and $q_i(z) = \mathcal{M}(\tau_i)$ where $\tau_i = (\tau_{i1}, \dots, \tau_{iK})$ $\mathcal{M}(\tau)$ represents a multinomial distribution with parameter τ . This is the standard mean field variational approximation used for network models with latent community assignment (Celisse et al., 2012; Matias, Rebafka, et al., 2018) Given this, the remaining global parameters of our model are $\theta = (\pi, \lambda)$, where λ captures the parameters of the group level point processes and π the overall group proportions. Our proposed online method for network point processes with group assignment can then be described as follows.

Algorithm 2 Online Inference for Point Processes on Networks

Set initialization of $\theta^{(0)} = \theta_0$.

for $m = 1$ to M **do**

Update the latent distribution $q^{(m)}(z) = \prod_{i=1}^n q_i^{(m)}(z_i)$ by

$$q_i^{(m)}(z_i) \propto \pi^{(m-1)} \exp \left\{ \mathbb{E}_{q^{(m-1)}(z_{-i})} l_m(\theta^{(m-1)} | z) \right\} \cdot S^{(m-1)}(z_i), \quad (4.6)$$

 giving estimates $\hat{\tau}^m$.

Update the point process parameters by

$$\lambda^{(m)} = \lambda^{(m-1)} + \eta_m \frac{1}{|A|} \frac{\partial \mathbb{E}_{q^{(m)}(z)} l_m(\theta | z)}{\partial \lambda}.$$

Update the community proportions using

$$\pi_k = \frac{1}{n} \sum_{i=1}^n \tau_{ik}, \text{ for } k = 1, \dots, K.$$

end for

Output Set $\hat{\lambda}^o = \lambda^{(M)}$, $\hat{\tau}^o = \hat{\tau}^{(M)}$ and $\hat{\pi}^o = \hat{\pi}^{(M)}$.

Here $S^{(m)}(z_i) = S^{(m-1)}(z_i) \exp \{ \mathbb{E}_{q^{(m-1)}(z_{-i})} l_m(\theta^{(m-1)} | z) \}$ with $S^{(0)}(z_i) = 1/K$ for $z_i = 1, \dots, K$, while z_{-i} is a sub-vector of z with the i th entry removed. The quantity $S^{(m)}$ can be viewed as an n by K matrix which stores personal cumulative group evidence up to the current time window for

each individual i and latent class k . The step size η_m is the adaptive learning speed, which may depend on m .

One of the main contributions of our algorithm is that we update the distribution of latent profiles adaptively by using cumulative historical information. An individual’s latent profile is approximated by a sequence of probability distributions, $q^{(m)}(z) = \prod_{i=1}^n q_i^{(m)}(z_i)$, by assuming there is no dependence structure between the latent assignment of nodes. In the update of $q^{(m)}$ we do not need to go through past events, as all group information has been compressed into the cumulative matrix $S^{(m)}$.

This model is of a similar form to that proposed for online estimation of LDA, where documents arrive as streams (Hoffman, Bach, et al., 2010). In that setting, each of D known documents in the corpus is observed sequentially. After word counts of an individual document are observed, an E-step is performed to determine the optimal local parameters for the per-document topic weights and per word topic assignments. Then an estimate of the optimal global of the topic weights is computed, $\tilde{\lambda}$, as if the total corpus consisted of the current document observed D times. The actual estimate of λ , which parameterizes the posterior distribution over the topics, is estimated using a weighted average of the previous estimate and $\tilde{\lambda}$. This is similar in spirit to our proposed method, where we compute optimal values given the current observation data and update our overall estimates using these estimates from our current window. In our SBM type model, all parameters are global and must be updated with each window of data considered. Similarly, the online LDA procedure of Hoffman, Bach, et al. (2010) utilises the fact that the size of the total corpus is known. A similar technique is used for the general stochastic variational inference procedure (Hoffman, Blei, et al., 2013). Here, we want our model to be flexible to the total number of events which will be observed, which is controlled by T , the total observation time. In practice this will be unknown, and so we wish to consider an update scheme which allows for this. We also simply take direct gradient steps as each window of events are observed, without optimising the global parameters within each iteration of our online scheme.

We provide detailed algorithms for learning Poisson processes and Hawkes processes on net-

works of event streams. Specifically, Algorithm 3 describes the detailed online estimation procedure for the homogeneous Poisson process. It only requires storing the cumulative number of events without storing any event history. This aids in reducing memory cost. Similarly, Appendix A.1 describes the detailed online estimation procedure for the homogeneous Hawkes process with exponential kernel function. The procedure for an inhomogeneous process with (say) a step function base rate is similar. Also included is a support algorithm which describes the detailed procedure for keeping historical data by creating a hash map with the key being the pair of nodes and their history information. We only need to store the *sufficient statistics* (Lehmann and Casella, 2006) which already contains all information about model parameters. Specifically, we create a hashmap C , whose key is (i, j) ($i, j \in [n]$) and corresponding value is the sufficient statistic of the specific model. These values will be updated by incorporating new information, as new data in the current time window is processed. Hence, the proposed algorithm effectively optimizes computational memory costs.

4.3.2 Approximation via Variational Inference

We first wish to expand the discussion of the variational approximation being considered here. When the labels of individuals are known, the conditional log likelihood can be written explicitly as

$$l(\theta|z) = \sum_{(i,j) \in A} \left\{ \int_0^T \log \lambda_{ij}(t|z) dN_{ij}(t) - \int_0^T \lambda_{ij}(t|z) dt \right\}.$$

Then the complete log likelihood is

$$l(\theta, z) = \sum_{i=1}^n \log \pi_{z_i} + l(\theta|z). \quad (4.7)$$

Furthermore, the marginal log likelihood can be written as

$$l(\theta) = \log \left\{ \sum_z \left[\prod_{i=1}^n \pi_{z_i} L(\theta|z) \right] \right\}, \quad (4.8)$$

where $L(\theta|z) = \exp\{l(\theta|z)\}$ is the conditional likelihood.

As seen in (4.8), it is difficult to compute this likelihood directly, which requires summation over exponentially many terms. An alternative approach is by using variational inference (Hoffman, Blei, et al., 2013) methods to optimize the evidence lower bound (ELBO) instead of the log likelihood. The ELBO is defined as

$$\text{ELBO}(\theta) = \mathbb{E}_{q(z)} l(\theta, z) - \mathbb{E}_{q(z)} \log q(z), \quad (4.9)$$

where this expectation is taken with respect to z and $q(z)$ is some approximate distribution for z .

Using the multinomial variational family described above, the ELBO can then be written as,

$$\text{ELBO} = \sum_{(i,j) \in A} \sum_{k,l} \tau_{ik} \tau_{jl} \left\{ \int_0^T \log \lambda_{kl}(t) dN_{ij}(t) - \int_0^T \lambda_{kl}(t) dt \right\} + \sum_i \sum_k \tau_{ik} \log \pi_k / \tau_{ik}. \quad (4.10)$$

Note that

$$\mathbb{E}_{q(z)} l_m(\theta|z) = \sum_{(i,j) \in A} \sum_{k,l} \tau_{ik} \tau_{jl} \left\{ \int_{(m-1) \cdot dT}^{m \cdot dT} \log \lambda_{kl}(t) dN_{ij}(t) - \int_{(m-1) \cdot dT}^{m \cdot dT} \lambda_{kl}(t) dt \right\},$$

and therefore, we can write the first term as a sum over disjoint integrals

$$\text{ELBO} = \sum_{m=1}^M \mathbb{E}_{q(z)} l_m(\theta|z) + \sum_i \sum_k \tau_{ik} \log \pi_k / \tau_{ik}.$$

Hence, the new representation is in an additive form, which is more amenable to online optimization.

Define the estimator $\hat{\tau}_i^{(m)}$ to be the maximizer for the m -th time window of individual i , given

the previous estimates, as

$$\hat{\tau}_i^{(m)} \equiv \operatorname{argmax}_{\tau_i} \left\{ \sum_{w=1}^m \mathbb{E}_{q_i(z_i)} \mathbb{E}_{q^{(w-1)}(z_{-i})} l_w(\theta^{(w-1)}|z) + \sum_i \sum_k \tau_{ik} \log \pi_k^{(m-1)} / \tau_{ik} \right\}. \quad (4.11)$$

We then have the following result, to explain that the approximation step in our proposed algorithm is aiming to find the best approximate posterior distribution for each individual at each time window.

Lemma. *The optimizer of (4.11) is given by equation (4.6).*

Proof. By simplification, we have that

$$\begin{aligned} & \sum_{w=1}^m \mathbb{E}_{q_i(z_i)} \mathbb{E}_{q^{(w-1)}(z_{-i})} l_w(\theta^{(w-1)}|z) + \sum_i \sum_k \tau_{ik} \log \pi_k^{(m-1)} / \tau_{ik} \\ &= \sum_{k=1}^K \tau_{ik} \sum_{w=1}^m \mathbb{E}_{q^{(w-1)}(z_{-i})} l_w(\theta^{(w-1)}|z_{-i}, z_i = k) + \sum_k \tau_{ik} \log \pi_k^{(m-1)} - \sum_k \tau_{ik} \log \tau_{ik} + C_1 \\ &= \sum_{k=1}^K \tau_{ik} \log \left\{ \pi_k^{(m-1)} \exp \left[\sum_{w=1}^m \mathbb{E}_{q^{(w-1)}(z_{-i})} l_w(\theta^{(w-1)}|z_{-i}, z_i = k) \right] \right\} - \sum_{k=1}^K \tau_{ik} \log \tau_{ik} + C_1 \\ &= -KL(q_i \| p_i) + C_2, \end{aligned}$$

where C_1, C_2 are some constants free of τ_i and p_i is some multinomial distribution with

$$p_i(z = k) \propto \pi_k^{(m-1)} \exp \left\{ \sum_{w=1}^m \mathbb{E}_{q^{(w-1)}(z_{-i})} l_w(\theta^{(w-1)}|z_{-i}, z_i = k) \right\}.$$

Hence, the maximizer is achieved when $q_i = p_i$, that is

$$\tau_{ik} \propto \pi_k \exp \left\{ \sum_{w=1}^m \mathbb{E}_{q^{(w-1)}(z_{-i})} l_w(\theta^{(w-1)}|z_{-i}, z_i = k) \right\}.$$

Lastly, we denote $\exp\{\sum_{w=1}^m \mathbb{E}_{q^{(w-1)}(z_{-i})} l_w(\theta^{(w-1)}|z_{-i}, z_i = k)\}$ as $S^{(m)}(k)$, which can be computed

recursively by the formula

$$S^{(m)}(k) = S^{(m-1)}(k) \exp \left\{ \mathbb{E}_{q^{(m-1)}(z_{-i})} l_n(\theta^{(m-1)} | z_{-i}, z_i = k) \right\}.$$

This completes the proof. □

Algorithm 3 Online-Poisson

- 1: Input: *data*, number of groups K , window size dT , edge list A .
 - 2: Output: \hat{B} , $\hat{\pi}$.
 - 3: Initialization: S , τ , π , B .
 - 4: Set $M = T/dT$
 - 5: **for** window $m = 1$ to M **do**
 - 6: Read new data between $[(m-1) \cdot dT, m \cdot dT]$
 - 7: Create temporary variables $S_p \in \mathbb{R}^{n \times K}$, $B_{p1}, B_{p2} \in \mathbb{R}^{K \times K}$.
 - 8: Set learning speed: $\eta = \frac{K^2}{\sqrt{mm_t}}$, where m_t is the number of events between $[(m-1) \cdot dT, m \cdot dT]$.
 - 9: **for** events in current window **do**
 - 10: Compute B_{p1}, B_{p2}, S_p :
 - 11: $S_p(i, k) += \tau_{jl}$ for i, j in events
 - 12: $S_p(i, k) -= \tau_{jl} B_{kl} dT$ for i, j in A
 - 13: $B_{p1}(k, l) += \tau_{ik} \tau_{jl}$ for i, j in events
 - 14: $B_{p1} = B_{p1}/B$
 - 15: $B_{p2}(k, l) += \tau_{ik} \tau_{j,l}$ for i, j in A
 - 16: $S += S_p$.
 - 17: **end for**
 - 18: Compute the negative gradient: $grad_B = B_{p1} - B_{p2}$.
 - 19: Update the parameters:
 - 20: Update B by setting $B = B + \eta \cdot grad_B$
 - 21: Update τ by setting $\tau_{ik} = \frac{\pi_k S_{ik}}{\sum_k \pi_k S_{ik}}$ for $i \in [n]$ and $k \in [K]$.
 - 22: Update π by setting $\pi_k = \frac{1}{n} \sum_i \tau_{ik}$ for $k = 1, \dots, K$.
 - 23: **end for**
-

4.4 Simulation Studies

Given our proposed inference scheme, we first wish to thoroughly validate its performance in simulation studies. We shall evaluate our procedure in terms of both community and parameter recovery, while also investigating the empirical regret performance and monitoring the online loss.

One important consideration here is the choice of dT , the window size over which continuous data is processed. Here, throughout we use a fixed window size of $dT = 1$. Additional simulations in Appendix A indicate that in practice, once the number of windows, M , is sufficiently large, this choice does not impact the overall algorithm performance.

Recover communities We first demonstrate that we can correctly recover the true communities and that this can be achieved in an online fashion, as the data is observed. To do this, we simulate data from a fixed Poisson block model with $K = 2$ communities. Additional simulations for Hawkes processes are included in Appendix A.2. We consider a sparse setting and evaluate the performance of our proposed inference scheme on the final estimated community assignments, having learnt this structure in an online fashion taking a single pass through the observed events. Figure 4.2 demonstrates the performance, in terms of Adjust Rand Index (ARI) (Hubert and Arabie, 1985), as we increase the size of the network with $n = 100, 200, 500, 1000$. For each network size we simulate 50 sample networks and corresponding events across time. We see that as the number of nodes in the network grows, with the total observation period remaining fixed, we can better recover the true community structure of the nodes. In particular, for small networks it can be quite challenging to recover the true structure however once the number of nodes increases, along with the corresponding number of total events which occur on the network, we are better able to capture the true structure.

Online Community Recovery Given that we iteratively update our estimates as we observed events, we can also examine how the performance of our estimation scheme evolves over time. For example, we can investigate how quickly the community structure, z is captured as events on the network are observed. The online recovery of the other parameters, θ , is discussed below. Figure 4.3 illustrates this for the same simulation setting considered previously. As we observe and process the data in an online fashion we store

$$\hat{\mathbf{t}}^{(m)} = (\hat{\tau}_1^m, \dots, \hat{\tau}_n^m),$$

the estimated node labels for the m -th window. We then compute the ARI between the true communities and the estimates at intermediate time points. Figure 4.3 shows this performance for a range of network sizes. Unsurprisingly, we see that as the number of nodes in the network increases, the estimated community structure quickly agrees with the true assignments. While all network sizes show a large degree of variation in the ARI between the estimated and true labels initially, this decreases quickly as the number of nodes increases. In particular, for sufficiently large networks we are able to consistently recover the community assignments having observed only half of the total observation data.

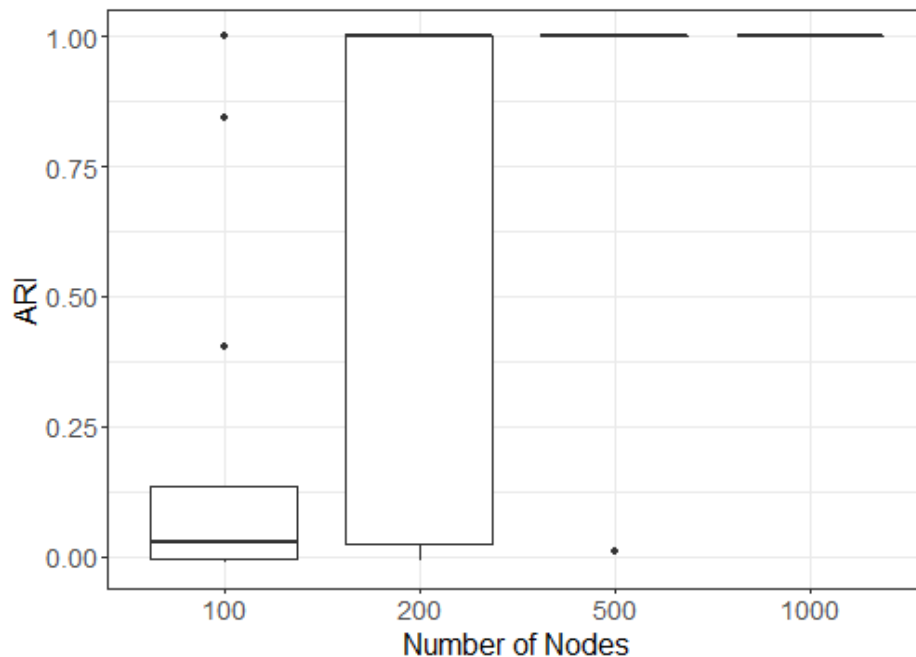


Figure 4.2: Overall community recovery for our online inference procedure as we increase the number of nodes in the network, with each box-plot corresponding to 50 simulations. For $n = 100$ community recovery is challenging, however as the number of nodes increases we can correctly recover the community structure in almost all simulations.

Similarly, while we can also investigate community recovery as we vary the number of communities, for a fixed number of nodes. Figure 4.4 illustrates that as we increase the number of communities, we remain able to recover the true community structure, with expected increased uncertainty.

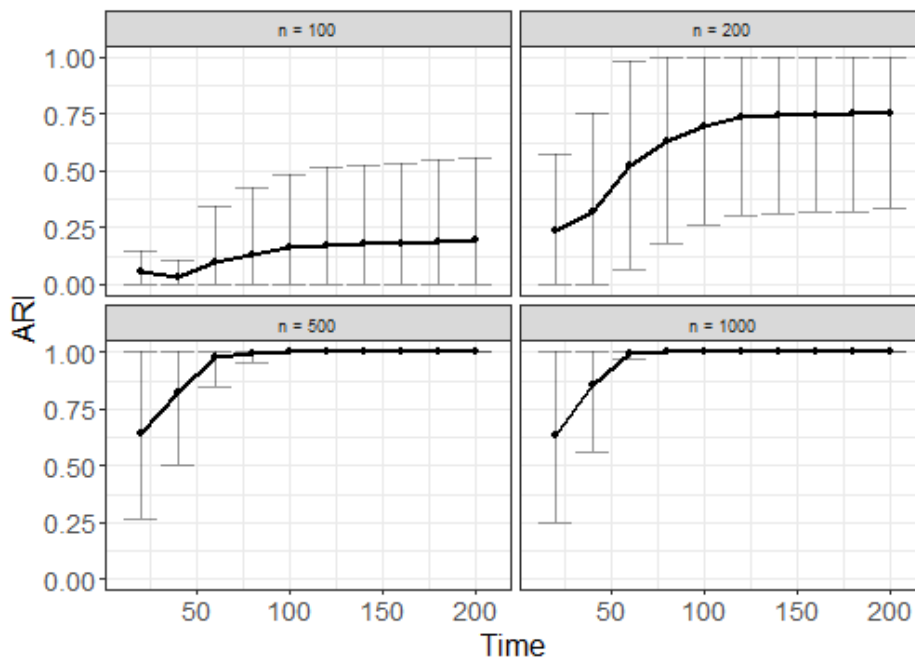


Figure 4.3: Demonstrating the ARI of the estimated clustering as we observe events in term for varying network size. Here we show the mean ARI with error bars corresponding to one standard deviation. We see that for all network sizes, we can identify community structure, with the larger networks doing so quickly and with less variability.

Monitoring Convergence A natural question in variational inference is how to identify whether the model has converged and whether it has converged to a local optima. For coordinate ascent variational inference convergence can be assessed by monitoring the ELBO and identifying when the change in this quantity from the previous iteration of the coordinate ascent scheme falls below some threshold (Blei et al., 2017). As we are observing the data sequentially here we cannot use this metric to assess convergence in practice. The ELBO will decrease as we observe new events.

Were the total observation period and all event times known in advance, we could compute the ELBO for the full data set, using the estimates we observe in an online manner (i.e, using only the data up to the current time point to form the parameter estimates, but then computing the ELBO for all data). In what follows, this quantity will be denoted the *full ELBO*. We illustrate this for a simulation setting with Figure 4.5, illustrating the corresponding estimates from fitting a batch estimate to this data. We see that the full ELBO from the online estimates quickly converges to a similar optimum, after having only observed events on the network for a short time.

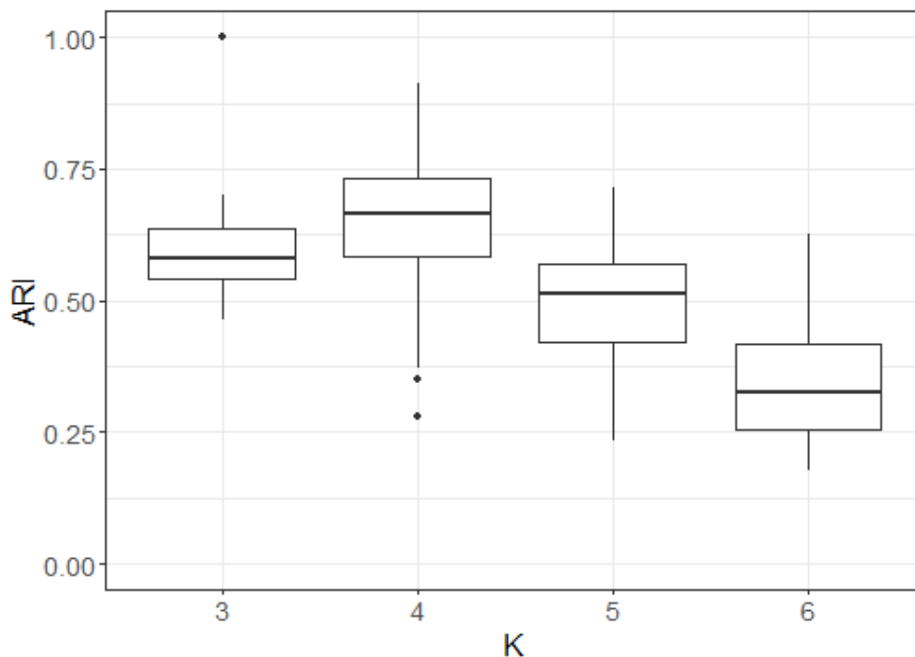


Figure 4.4: ARI as we consider network structure with an increasing number of communities. As we increase the number of communities in the network we remain able to recover the community structure, with increasing uncertainty as the number of communities increases.

Similarly, an advantage of the online procedure is that we obtain good parameter estimates using only a small number of events, compared to batch estimates which must use all events repeatedly. We illustrate this for simulated data in Figure 4.6. We again show the full ELBO against the percent of events used to calculate the corresponding parameter estimates. Similarly, we show the convergence of the batch procedure, which repeatedly uses all events to obtain a similar optimum.

In practice, we do not know the total number of events or the total observation time when fitting our method, and so cannot compute this full ELBO. As such, we need an alternative metric to assess convergence. Our online inference procedure gives us a “windowed” ELBO, corresponding to the current events observed in the current period, and the estimates of our parameters after observing data in that window,

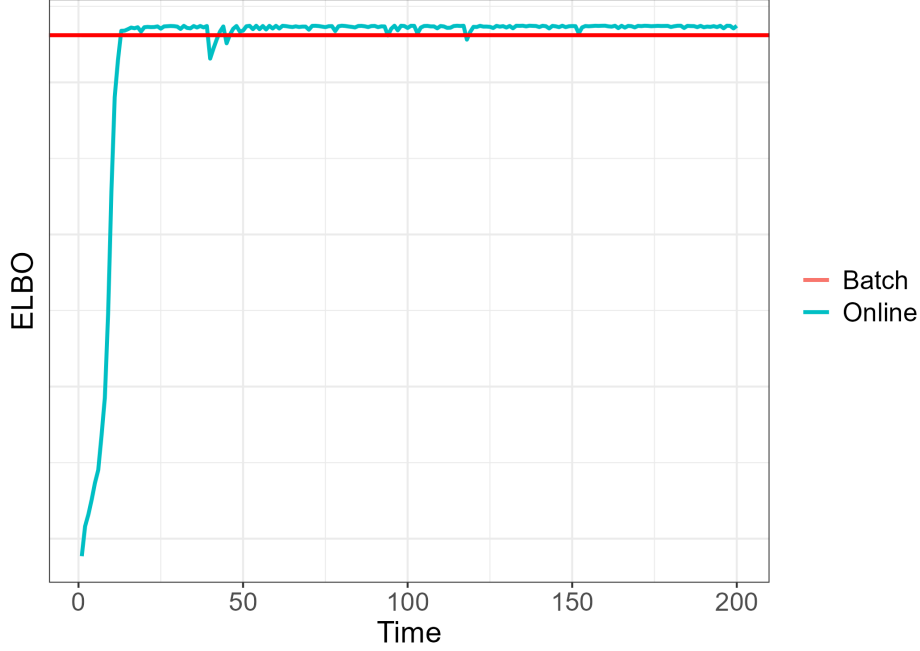


Figure 4.5: Computed ELBO based on the full data, using our online estimates. We see that the ELBO from the online estimates quickly converges to the optimum found by the corresponding batch estimate.

$$\begin{aligned}
 ELBO(m) = & \sum_{(i,j) \in A} \sum_{k,l} \tau_{ik}^{(m)} \tau_{jl}^{(m)} \left\{ \int_{(m-1) \cdot dT}^{m \cdot dT} \log \lambda_{kl}^{(m)}(t) dN_{ij}(t) - \int_{(m-1) \cdot dT}^{m \cdot dT} \lambda_{kl}^{(m)}(t) dt \right\} \\
 & + \sum_i \sum_k \tau_{ik}^{(m)} \log \pi_k^{(m)} / \tau_{ik}^{(m)}.
 \end{aligned}$$

We wish to consider a potential metric based on this quantity to assess convergence. In particular, we define a normalized ELBO,

$$\overline{ELBO}(m) := \frac{1}{N_m} ELBO(m), \tag{4.12}$$

where N_m is the total number of observed events up to the end of the m -th time window. In Figure 4.7 we plot this normalized ELBO along with the full ELBO for this previously simulated data. We see that the convergence rate between these two metrics is similar, indicating it may serve as a possible tool to monitor convergence in an online fashion.

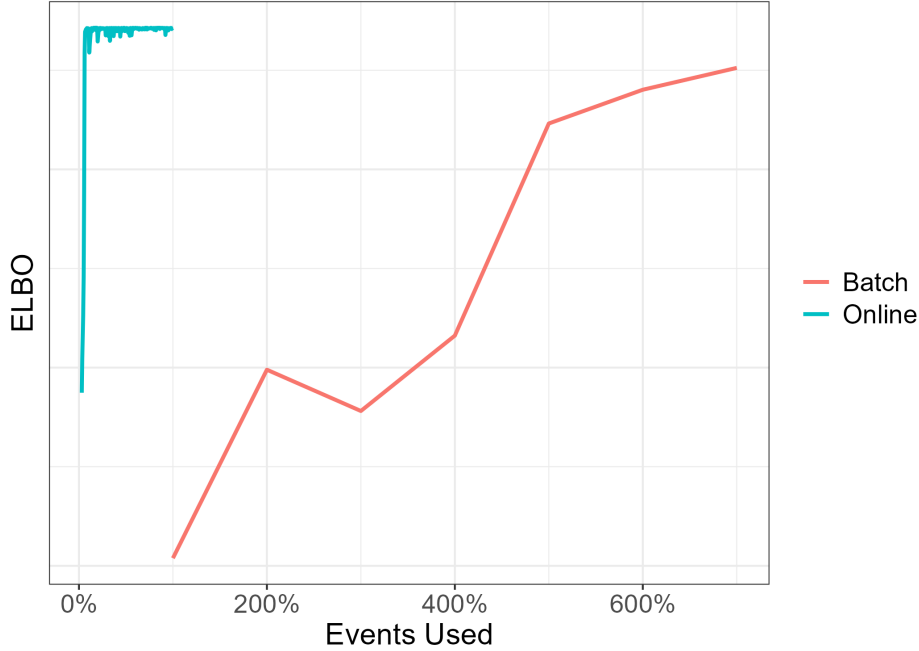


Figure 4.6: The full ELBO plotted against the percent of total events used to obtain the corresponding parameter estimates. We see that the online procedure obtains good estimates of the full ELBO using significantly fewer events.

We can also compute the corresponding values of this normalised ELBO for the same simulation setting as Figure 4.2, varying the number of nodes in the network for a fixed T . These results are shown in Figure 4.8. In all simulations this quantity converges towards 0 quickly. There does appear to be some evidence for differences between simulations which recover the true parameters and those which do not, evidenced by slightly slower convergence.

Parameter Recovery We can also assess our ability to recover the parameters of our model in an online fashion. To evaluate this for a block homogeneous Poisson model with true rate matrix B , we monitor the relationship between the estimate $\hat{B}^{(m)}$ and B by computing $\frac{1}{K^2} |\sum_{ij} \hat{B}_{ij}^{(m)} - \sum_{ij} B_{ij}|$, to account for possible permutation of the node labels. In particular, for a given observation period $[0, T]$, we simulate 50 event streams from the same network structure and estimate the parameters for each. Figure 4.9 shows the loess smoothed estimates of this quantity across simulations as we increase T . This difference shrinks quickly across all values of T .

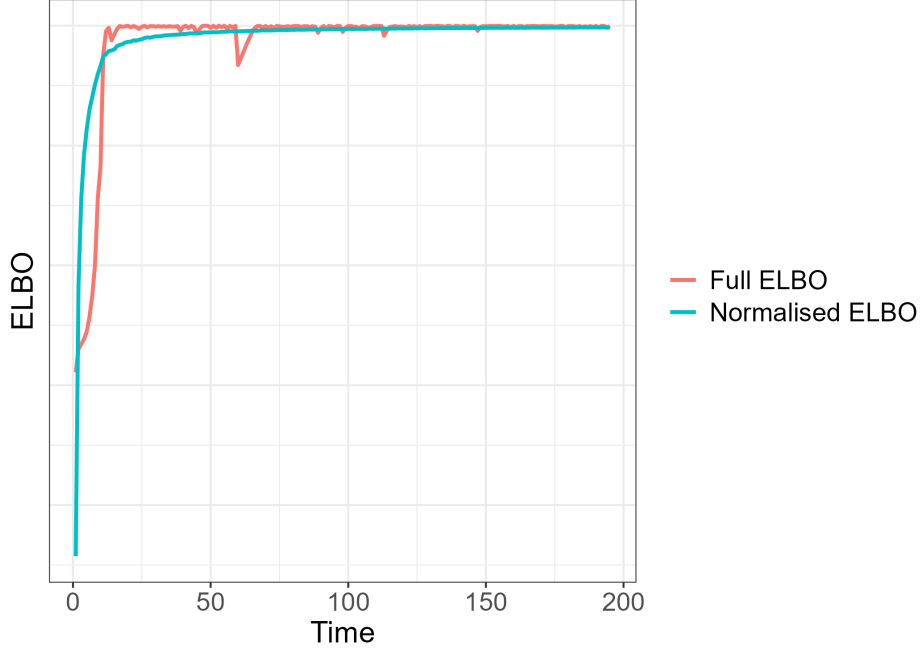


Figure 4.7: The normalized ELBO and the full ELBO for simulated data, suitably scaled. Empirically, these quantities converge similarly over time, indicating this normalized may be a suitable measure of convergence.

Regret Rate Quantifying the regret of an online estimation scheme is an important tool in the analysis of such a procedure. We can investigate the empirical performance of the regret for our method in simulation settings. To do this we need to consider a loss function. We define the loss function over the m -th time window as the negative normalized log-likelihood, i.e.

$$\tilde{l}_m(\theta|z) = -\frac{1}{|A|} \sum_{(i,j) \in A} \left\{ \int_{(m-1)dT}^{m dT} \log \lambda_{ij}(t|z) dN_{ij}(t) - \int_{(m-1)dT}^{m dT} \lambda_{ij}(t|z) dt \right\}, \quad (4.13)$$

and define the regret as

$$\text{Regret}(T) = \inf_{\theta^{(m)} \in \Pi(\theta)} \left\{ \sum_{m=1}^M \tilde{l}_m(\theta^{(m)}|z^*) \right\} - \sum_{m=1}^M \tilde{l}_m(\theta^*|z^*), \quad (4.14)$$

where $M = T/dT$. This regret function quantifies the gap of the conditional likelihood, given the true latent membership z^* , between the online estimator and the true optimal value. We note that this regret function is conditional on the true latent assignment being known, and as such, we need

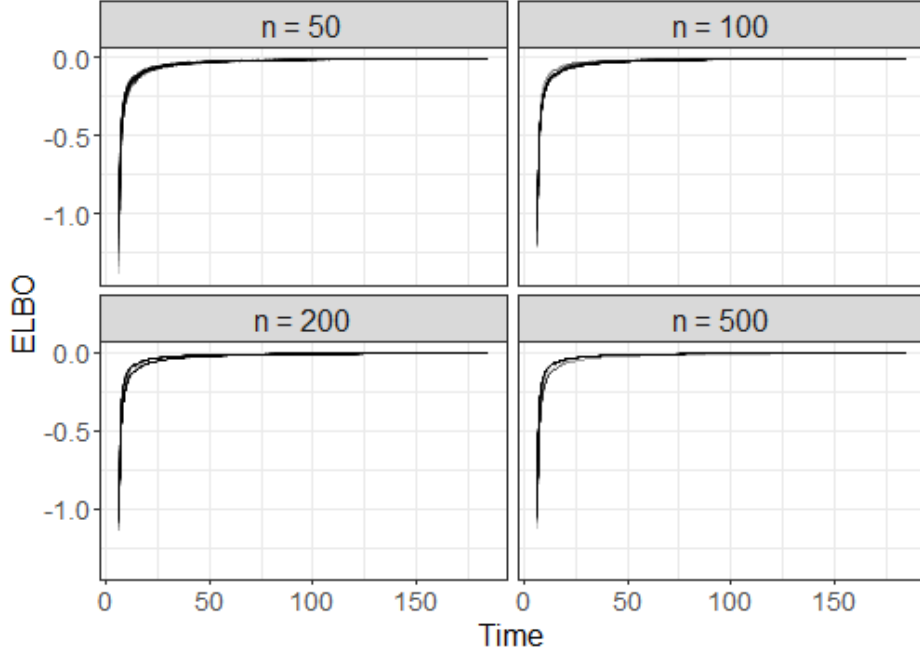


Figure 4.8: Normalized ELBO for a range of network sizes. This metric quickly converges for all simulations.

to account for possible permutations of the inferred parameters. While this regret quantity may be of theoretical interest, in practice it may be more appropriate to instead look at the empirical regret, using the estimated latent community memberships. We shall define this as

$$\text{Regret}_{EMP}(T) = \sum_{m=1}^M \tilde{l}_m(\theta^{(m)}|z^{(m)}) - \sum_{m=1}^M \tilde{l}_m(\theta^*|z^*), \quad (4.15)$$

measuring the cumulative difference between the estimated and true log likelihood, as we learn both z and θ over time.

Given these two regret definitions, we can simulate networks and compute the empirical regret for a fixed network, varying the range of time over which events are observed. This is shown in Figure 4.10. Here we compute each quantity across 50 simulations, showing smoothed estimates over time. We see that as we observe these networks for a longer period this regret grows slowly.

Online Loss A natural task when considering online learning is the ability to make online predictions and look at the loss of such a procedure over time. We consider such a scenario here for

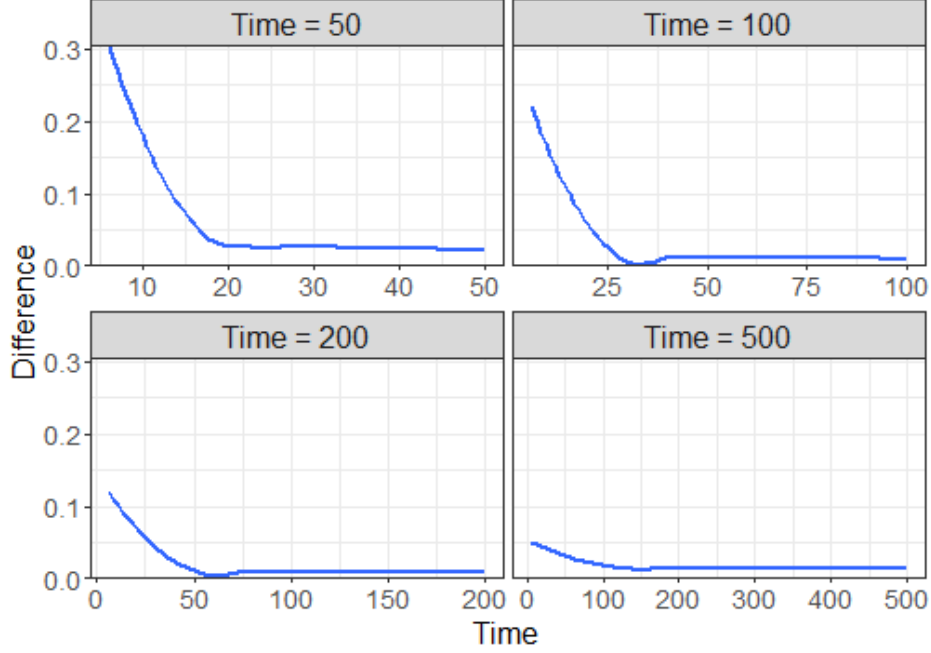


Figure 4.9: Average difference in estimated and true rate matrix across time across multiple simulations. We see that these converge quickly, particularly as events are observed for a longer period.

event data from a block model. In particular, we will use the negative log likelihood as our loss function. Given parameter estimates from the $(m - 1)$ -th time window, the negative log likelihood for the next window of observed data is given by

$$\ell_m(\theta) := -l_m(\theta^{(m-1)}|z^{(m-1)}) = - \int_{(m-1)dT}^{mdT} \log \lambda_{ij}^{(m-1)}(t|z) dN_{ij}(t) + \int_{(m-1)dT}^{mdT} \lambda_{ij}^{(m-1)}(t|z) dt. \quad (4.16)$$

We then define the average cumulative loss after m observation batches as $\frac{1}{m} \sum_{i=1}^m \ell_i(\theta)$. We wish to compare this quantity to the best average cumulative loss, without learning the model in an online fashion. To do this, we determine the best overall batch estimate, using all events repeatedly, which we denote as $\hat{\theta}, \hat{z}$, and use these estimates to compute $\bar{l} := \frac{1}{M} \sum_{m=1}^M -l_m(\hat{\theta}|\hat{z})$, the best average cumulative loss in hindsight. Note that here we are computing the average cumulative loss using both the current estimates of $\hat{\theta}$, the parameters of the point process and also \hat{z} , the current estimates of the latent community assignments.

We then repeat this procedure for a fixed network size, varying T , the total observation time.

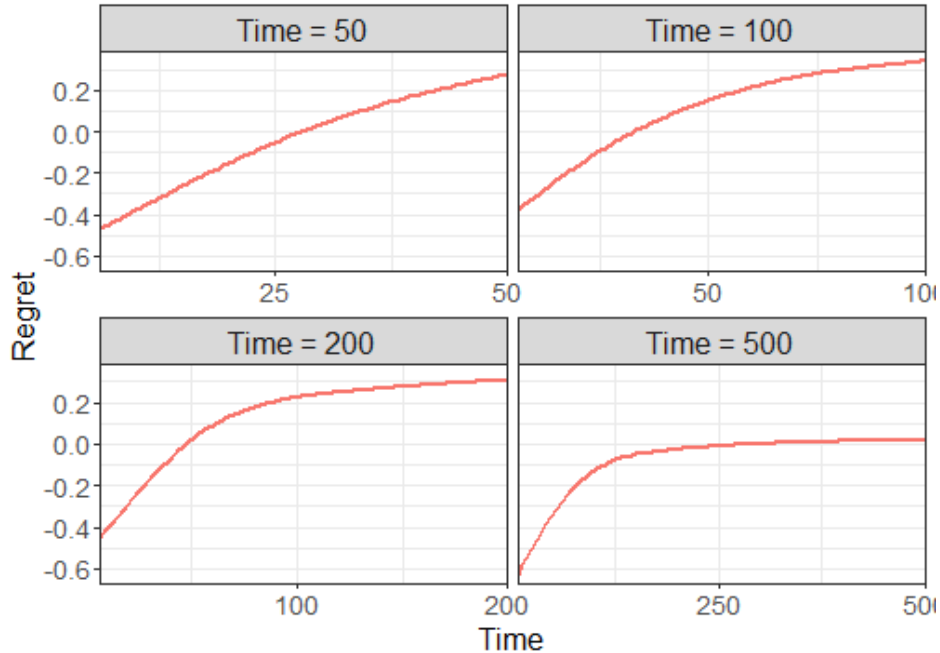


Figure 4.10: Smoothed regret estimates as a function of time for a fixed network structure observed for varying lengths of time.

For fixed simulated data, we run our online inference procedure 50 times. Each time we compute the online loss for the observations in the subsequent window of length $dT = 1$. We show this in Figure 4.11, where each black line denotes the average cumulative loss over time for one run of our online inference scheme. \bar{l} is shown with the red horizontal line. We see that initially the average cumulative loss of our inference procedure is large, as the network structure is being learned from a small number of events, before quickly decreasing towards the best batch loss over time, as the correct intensities are estimated.

4.5 Real Data Analysis

To evaluate our online algorithms on real data, we consider the problem of link prediction, using large temporal networks from the literature. We consider three such networks, available from the Stanford Large Network Dataset collection (Leskovec and Krevl, 2014). They consist of the timestamps of:

- A collection of emails sent by users in a large university. This consists of 300k emails

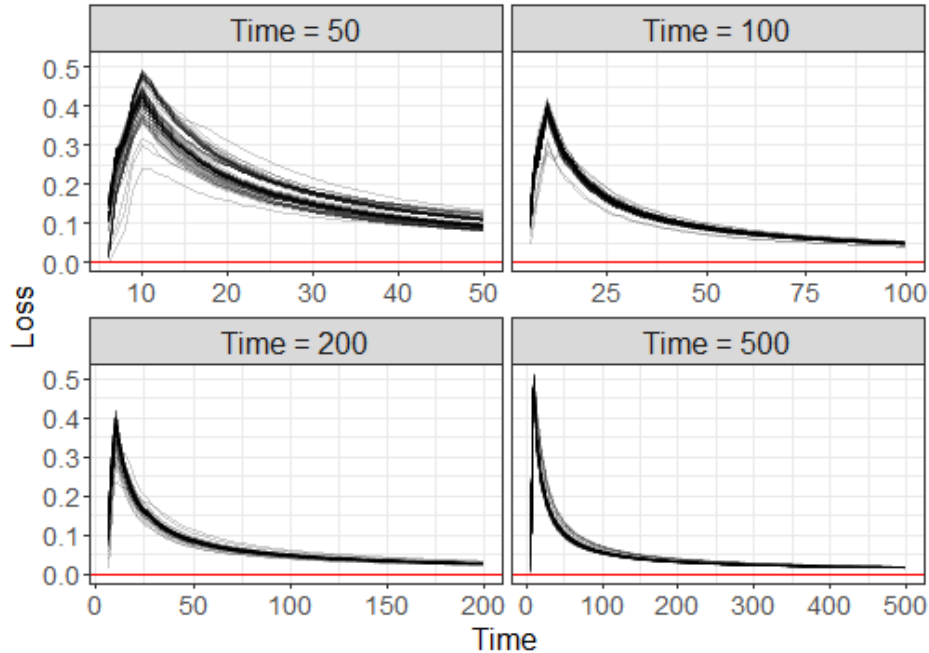


Figure 4.11: Average cumulative loss for our online inference procedure, with each line denoting one run of the procedure. The red horizontal line is the best average cumulative loss in hindsight. We see that as T increases, our online procedure comes close to this best loss.

between approximately 1000 users over 803 days.

- Messages sent between 2000 students on an online college social network platform over 193 days, consisting of 60k messages.
- Interactions from the Math Overflow website over 2350 days. Here we have 25k users and 500k directed interactions, where an interaction from user i to user j means that user i responded to a question posed by user j .

The temporal component in these networks changes over the observed time, with interactions much sparser towards the end of the observed time period. This makes link prediction a challenging problem in this setting. For each of these networks, we fix K , the number of communities, based on knowledge of the network structure, as we aim to compare link prediction for a given K . We use K as considered elsewhere for these examples (Miscouridou et al., 2018). We partition the events into training and test periods which contain 85% and 15% of events respectively. Note that we consider the edge structure, A , known in advance, although we could also learn this from the training data

and use that as our estimate of the overall edge list. Given the events observed initially, the goal is to predict the number of events that will occur between a directed pair over the test period.

To fit these models, we consider dT such that $M = \frac{T}{dT} \approx 100$ for the online estimators, with the same maximum number of iterations for our corresponding batch versions. For the inhomogeneous models, we consider 7 step functions as our basis functions, aiming to capture day of the week effects present in our event streams. We take the average of these basis functions as an estimate of our baseline rate. The results for this link prediction problem are shown in Table 4.1, with the corresponding computation times (in seconds) shown in Table 4.2. Our online procedure obtains comparable estimates to more expensive batch estimates, and is better suited to estimation for the large networks considered here, obtaining comparable predictions generally quicker.

Method	Email	College	Math
Poisson	11.73/12.9	5.16/13.96	2.13/1.99
Hawkes	19.42/12.74	5.32/5.09	2.06/2.14
In-Poisson ($H = 7$)	15.09/18.92	5.57/5.67	2.14/2.14
In-Hawkes ($H = 7$)	14.84/12.9	5.58/5.44	2.14/2.14

Table 4.1: Median RMSE of predicted event counts vs true event counts in held out test set across 50 simulations. Online/Non-online estimates.

Method	Email	College	Math
Poisson	0.71/24.56	0.09/1.54	7.35/4.49
Hawkes	1.44/14.89	1.51/0.42	235.86/314.14
In-Poisson ($H = 7$)	2.03/20.37	1.85/5.06	257.74/28.08
In-Hawkes ($H = 7$)	2.29/52.43	2.11/4.51	253.52/327.05

Table 4.2: Median computation time for Online/Full Model fitting (seconds) across 50 simulations.

4.6 Conclusion

In this chapter we propose an online learning framework for event streams on large networks. We develop a scalable online algorithm to uncover community structure using point process models on the network, considering both computational speed and memory requirements. In both simulations and experiments, we observe that our method is scalable compared with batch meth-

ods especially under large network settings when both n , the number of nodes and T , the total time, grows. We also investigate the empirical properties of our proposed estimation scheme.

There are many ways this work could be extended. There are several aspects of community detection which we have not addressed. In particular, further investigation could indicate better methods of initializing our algorithm in this online setting. Experiments indicate this is an important consideration for fitting these models with the goal of community detection, and it is necessary to start in a neighbourhood of the true estimates to recovery communities, particularly in small networks. Similarly, selecting the number of communities is an important problem in these models, and it is not immediate how to approach this with an online algorithm. Our algorithm also assumes that the edge structure A does not vary in time. It is of interest to consider a model where A can also evolve over time. In which case, it would be of interest to also estimate A in an online setting, along with deriving properties of estimators for this updated model.

The proposed framework is also connected to many popular longitudinal models (e.g. the dynamic latent space model (Sewell and Chen, 2015), the temporal exponential random graph model (Leifeld et al., 2018), and the varying coefficient model for dynamic networks (Lee, Li, et al., 2017)) which can be viewed as the discrete time event processes. With suitable modifications, our results can be incorporated into these related settings.

Chapter 5: Discussion and Future Directions

In this thesis we have considered latent variable models for events on networks. Although statistical network models are commonly concerned with a single observation of a static network, in reality the true data used to generate this representation of a network often consists of events observed across time between nodes in the network. Data of this form occurs in a vast array of settings and is widely collected. While in some fields this data can be labour intensive to collect, new tools in machine learning for video data are making automated collection easier (Mathis et al., 2018). This will result in event data on networks becoming even more available in the future.

Event data for networks is commonly aggregated to construct a static representation of the relationships between nodes in a network. While there are a myriad of methods for networks of this form, utilising them when the more complex temporal event data is available results in a loss of information. Methods which can incorporate event data into the network structure have the potential to better describe the relationships between nodes and the properties of the network.

Chapter 2 provides an overview of network models and inference for such models. Starting initially with models for static network data, we describe how such models have been extended to consider dynamic network data, consisting of repeated observations of an adjacency matrix. Finally, we introduce event data on networks, considering the event times for interactions between nodes directly.

In Chapter 3 we utilise latent variable models for event data on networks to learn latent rankings from aggressive animal interactions. This is a specific form of latent space model developed for data of this form. We incorporate known characteristics of this data into our model and learn a social hierarchy using the exact event times, unlike existing methods which aggregate this data. We show that this method can capture meaningful properties of these interactions.

Inference for data of this form is challenging and tools are required to aid computation. In

Chapter 4 we consider online inference to address this problem. Using the ideas of stochastic variational inference, we propose a natural inference procedure to learn block models for event data on networks, as the events are observed over the network. We thoroughly investigate the empirical properties of our inference scheme and show that it can successfully recover communities in an online fashion, while also investigating the performance in terms of online learning metrics such as regret rates and online loss.

There are several important future directions which can be considered in the context of this work. As described in Chapter 2, the evaluation of models for network data remains a topic which requires further work, especially given the widespread development of machine learning methods for network data in recent years. Often these methods are highly specialized, and as discussed in Ward et al. (2021), a formal framework to evaluate such methods would greatly aid in directing future developments. Point processes benefit from well understood theoretical properties and using point processes for event data on networks provides natural tools for model evaluation.

Concerning the latent ranking procedure of Chapter 3, one natural consideration is to relax the restriction to a one dimension latent ranking space, and the ability to consistently recover such rankings. A related problem has been considered in the psychometric literature, including Chen, Li, et al. (2020). Here one such goal could be to rank individuals along a specific trait, for example proficiency in a specific task. This is similar to the setting we consider here, where the trait could be thought of as some measure of position in the hierarchy. Chen, Li, et al. (2020) show that if the relationship between the latent factors and the observed responses is known in advance and can be encoded in a matrix Q , then conditions on the stability of Q can be used to ensure that a ranking of the latent factors can be recovered. This requires both N , the number of individuals (here animals) and J , the number of observed items (here potentially fights) to grow, however further work would be required to incorporate the continuous time nature of these repeated interactions, as in Xu, Fang, et al. (2018). A related idea would be to use such latent ranking vectors for classification of animals also. One challenge here, which also occurs in the current one dimensional model, is how to validate the inferred multidimensional latent ranking.

Throughout, we have assumed that our latent structure (such as a latent ranking or community assignment) is fixed in time. A natural question is to consider how one might identify changes in such structure over time. As highlighted in Matias and Miele (2017) for dynamic networks, there are potential identifiability concerns here. It is not clear if it is possible to distinguish changes in (say) community structure over time from changes in the intensity function of the point process. Related to this, a natural concern is the idea of identifying changes in the network structure in an online manner, as events are observed.

The models in this thesis have been considered for a range of data types, including human social interaction data such as email interactions or messages on a social network. Given the widespread adoption of online platforms which collect such data, it is important to be aware of the concerns surrounding such data. One natural concern is privacy. While publicly available social network data is anonymized by some tool, it is not always clear how precise this anonymization scheme is and what level of privacy does it afford users who may be present in the data. Is it possible to identify if someone is present in a given dataset of events on a network? Similarly, is it possible to identify who someone has interacted with? Such information, if it can be obtained, could potentially be used for nefarious purposes.

Several frameworks have been considered with the goal of preserving some level of privacy in a dataset. Differential privacy (Dwork and Roth, 2014) is one such procedure which has been widely studied in recent years and some initial work has considered privacy in the context of network data. However, further work is required in this setting, along with then considering mechanisms for privacy for event data on networks

References

- Airoldi, E. M. et al. (2008). “Mixed membership stochastic blockmodels”. In: *Advances in neural information processing systems* 21.
- Allman, E. S. et al. (2011). “Parameter identifiability in a class of random graph mixture models”. In: *Journal of Statistical Planning and Inference* 141.5, pp. 1719–1736.
- Alquier, P. and J. Ridgway (2020). “Concentration of tempered posteriors and of their variational approximations”. In: *The Annals of Statistics* 48.3, pp. 1475–1497.
- Amini, A. A. et al. (2013). “Pseudo-likelihood methods for community detection in large sparse networks”. In: *The Annals of Statistics* 41.4, pp. 2097–2122.
- Arastuie, M. et al. (2020). “CHIP: a Hawkes process model for continuous-time networks with scalable and consistent estimation”. In: *Advances in neural information processing systems* 33, pp. 16983–16996.
- Baumes, J. et al. (2004). “Discovering hidden groups in communication networks”. In: *International Conference on Intelligence and Security Informatics*. Springer, pp. 378–389.
- Bifet, A. and E. Frank (2010). “Sentiment knowledge discovery in twitter streaming data”. In: *International conference on discovery science*. Springer, pp. 1–15.
- Blei, D. M. et al. (2017). “Variational inference: A review for statisticians”. In: *Journal of the American statistical Association* 112.518, pp. 859–877.
- Blundell, C. et al. (2012). “Modelling reciprocating relationships with Hawkes processes”. In: *Advances in Neural Information Processing Systems* 25, pp. 2600–2608.
- Bonabeau, E. et al. (1999). “Dominance orders in animal societies: the self-organization hypothesis revisited”. In: *Bulletin of mathematical biology* 61.4, pp. 727–757.
- Brown, E. N. et al. (2002). “The time-rescaling theorem and its application to neural spike train data analysis”. In: *Neural computation* 14.2, pp. 325–346.
- Carpenter, B. et al. (2017). “Stan: A probabilistic programming language”. In: *Journal of statistical software* 76.1.
- Celeux, G. et al. (2000). “Computational and inferential difficulties with mixture posterior distributions”. In: *Journal of the American Statistical Association* 95.451, pp. 957–970.

- Celisse, A. et al. (2012). “Consistency of maximum-likelihood and variational estimators in the stochastic block model”. In: *Electronic Journal of Statistics* 6, pp. 1847–1899.
- Chase, I. and W. B. Lindquist (2017). “Dominance hierarchies”. In: *The Oxford Handbook of Analytical Sociology*. Oxford University Press, pp. 566–591.
- Chase, I. D. and K. Seitz (2011). “Self-structuring properties of dominance hierarchies: a new perspective”. In: *Advances in genetics*. Vol. 75. Elsevier, pp. 51–81.
- Chase, I. D., C. Tovey, et al. (2002). “Individual differences versus social dynamics in the formation of animal dominance hierarchies”. In: *Proceedings of the National Academy of Sciences* 99.8, pp. 5744–5749.
- Chen, S., A. Shojaie, et al. (2017). “The multivariate Hawkes process in high dimensions: beyond mutual excitation”. In: *arXiv preprint arXiv:1707.04928*.
- Chen, Y., X. Li, et al. (2020). “Structured latent factor analysis for large-scale data: Identifiability, estimability, and their implications”. In: *Journal of the American Statistical Association* 115.532, pp. 1756–1770.
- Chérif-Abdellatif, B.-E. et al. (2019). “A generalization bound for online variational inference”. In: *Asian Conference on Machine Learning*. PMLR, pp. 662–677.
- Clements, R. A. et al. (2011). “Residual analysis methods for space–time point processes with applications to earthquake forecast models in California”. In: *The Annals of applied statistics* 5.4, pp. 2549–2571.
- Daley, D. J. and D. V. Jones (2003). *An Introduction to the Theory of Point Processes: Elementary Theory of Point Processes*. Springer.
- Daudin, J.-J. et al. (2008). “A mixture model for random graphs”. In: *Statistics and computing* 18.2, pp. 173–183.
- David, H. A. (1987). “Ranking from unbalanced paired-comparison data”. In: *Biometrika* 74.2, pp. 432–436.
- De Bacco, C. et al. (2018). “A physical model for efficient ranking in networks”. In: *Science advances* 4.7, eaar8260.
- DeDeo, S. and E. A. Hobson (2021). “From equality to hierarchy”. In: *Proceedings of the National Academy of Sciences* 118.21.
- Drews, C. (1993). “The concept and definition of dominance in animal behaviour”. In: *Behaviour* 125.3, pp. 283–313.

- Dugatkin, L. A. (1997). “Winner and loser effects and the structure of dominance hierarchies”. In: *Behavioral Ecology* 8.6, pp. 583–587.
- Dugatkin, L. A. and R. L. Earley (2003). “Group fusion: the impact of winner, loser, and bystander effects on hierarchy formation in large groups”. In: *Behavioral Ecology* 14.3, pp. 367–373.
- Durante, D. and D. B. Dunson (2014). “Nonparametric Bayes dynamic modelling of relational data”. In: *Biometrika* 101.4, pp. 883–898.
- Dwork, C. and A. Roth (2014). “The algorithmic foundations of differential privacy.” In: *Found. Trends Theor. Comput. Sci.* 9.3-4, pp. 211–407.
- Erdős, P. and A. Rényi (1960). “On the evolution of random graphs”. In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1, pp. 17–60.
- Fan, X. et al. (2021). “Continuous-time edge modelling using non-parametric point processes”. In: *Advances in Neural Information Processing Systems* 34.
- Fortunato, S. and D. Hric (2016). “Community detection in networks: A user guide”. In: *Physics reports* 659, pp. 1–44.
- Fox, E. W. et al. (2016). “Modeling e-mail networks and inferring leadership using self-exciting point processes”. In: *Journal of the American Statistical Association* 111.514, pp. 564–584.
- Friel, N. et al. (2016). “Interlocking directorates in Irish companies using a latent space model for bipartite networks”. In: *Proceedings of the National Academy of Sciences* 113.24, pp. 6629–6634.
- Gilbert, E. N. (1959). “Random graphs”. In: *The Annals of Mathematical Statistics* 30.4, pp. 1141–1144.
- Gilks, W. R. et al. (1995). *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC.
- Giorgi, D. et al. (2018). *ppsbm: Clustering in Longitudinal Networks*. R package version 0.2.2.
- Girvan, M. and M. E. Newman (2002). “Community structure in social and biological networks”. In: *Proceedings of the national academy of sciences* 99.12, pp. 7821–7826.
- Glickman, M. E. (1999). “Parameter estimation in large dynamic paired comparison experiments”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 48.3, pp. 377–394.
- Gopalan, P. K. and D. M. Blei (2013). “Efficient discovery of overlapping communities in massive networks”. In: *Proceedings of the National Academy of Sciences* 110.36, pp. 14534–14539.

- Handcock, M. S. et al. (2007). “Model-based clustering for social networks”. In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 170.2, pp. 301–354.
- Hawkes, A. G. (1971). “Spectra of some self-exciting and mutually exciting point processes”. In: *Biometrika*, pp. 83–90.
- Hawkes, A. G. and D. Oakes (1974). “A cluster process representation of a self-exciting process”. In: *Journal of Applied Probability* 11.3, pp. 493–503.
- Hawkes, A. G. (Feb. 2018). “Hawkes processes and their applications to finance: a review”. In: *Quantitative Finance* 18.2, pp. 193–198.
- Hemelrijk, C. K. (2000). “Towards the integration of social dominance and spatial structure”. In: *Animal behaviour* 59.5, pp. 1035–1048.
- Hobson, E. A. (2020). “Differences in social information are critical to understanding aggressive behavior in animal dominance hierarchies”. In: *Current opinion in psychology* 33, pp. 209–215.
- Hobson, E. A. et al. (2021). “Aggression heuristics underlie animal dominance hierarchies and provide evidence of group-level social information”. In: *Proceedings of the National Academy of Sciences* 118.10.
- Hoff, P. D. (2005). “Bilinear mixed-effects models for dyadic data”. In: *Journal of the American Statistical Association* 100.469, pp. 286–295.
- Hoff, P. D. et al. (2002). “Latent space approaches to social network analysis”. In: *Journal of the American Statistical Association* 97.460, pp. 1090–1098.
- Hoffman, M., F. R. Bach, et al. (2010). “Online learning for latent dirichlet allocation”. In: *advances in neural information processing systems*, pp. 856–864.
- Hoffman, M. D., D. M. Blei, et al. (2013). “Stochastic variational inference”. In: *The Journal of Machine Learning Research* 14.1, pp. 1303–1347.
- Holland, P. W. et al. (1983). “Stochastic blockmodels: First steps”. In: *Social networks* 5.2, pp. 109–137.
- Hsu, Y. and L. L. Wolf (1999). “The winner and loser effect: integrating multiple experiences”. In: *Animal Behaviour* 57.4, pp. 903–910.
- Huang, S. and Y. Feng (2018). “Pairwise covariates-adjusted block model for community detection”. In: *arXiv preprint arXiv:1807.03469*.
- Hubert, L. and P. Arabie (1985). “Comparing partitions”. In: *Journal of classification* 2.1, pp. 193–218.

- Jonsson, P. F. et al. (2006). “Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis”. In: *BMC bioinformatics* 7.1, pp. 1–13.
- Karrer, B. and M. E. Newman (2011). “Stochastic blockmodels and community structure in networks”. In: *Physical review E* 83.1, p. 016107.
- Kim, B. et al. (2018). “A review of dynamic network models with latent variables”. In: *Statistics surveys* 12, p. 105.
- Klimt, B. and Y. Yang (2004). “Introducing the Enron corpus.” In: *CEAS*.
- Lazer, D. et al. (2009). “Co-citation of prominent social network articles in sociology journals: The evolving canon”. In: *Connections* 29.1, pp. 43–64.
- Lee, J., G. Li, et al. (2017). “Varying-coefficient models for dynamic networks”. In: *arXiv preprint arXiv:1702.03632*.
- Lee, W., J. Fu, et al. (2019). “Temporal microstructure of dyadic social behavior during relationship formation in mice”. In: *PloS one* 14.12, e0220596.
- Lehmann, E. L. and G. Casella (2006). *Theory of point estimation*. Springer Science & Business Media.
- Lei, J. and A. Rinaldo (2015). “Consistency of spectral clustering in stochastic block models”. In: *The Annals of Statistics* 43.1, pp. 215–237.
- Leifeld, P. et al. (2018). “Temporal exponential random graph models with btergm: Estimation and bootstrap confidence intervals”. In: *Journal of Statistical Software* 83.6.
- Leskovec, J. and A. Krevl (2014). *SNAP Datasets: Stanford large network dataset collection*.
- Lindquist, W. B. and I. D. Chase (2009). “Data-based analysis of winner-loser models of hierarchy formation in animals”. In: *Bulletin of mathematical biology* 71.3, pp. 556–584.
- Lu, Z. et al. (2015). “Algorithms and Applications for Community Detection in Weighted Networks”. In: *IEEE Transactions on Parallel and Distributed Systems* 26.11, pp. 2916–2926.
- Lusseau, D. (2003). “The emergent properties of a dolphin social network”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.suppl_2, S186–S188.
- Lusseau, D. et al. (2003). “The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations”. In: *Behavioral Ecology and Sociobiology* 54.4, pp. 396–405.

- Ma, Z. et al. (2020). “Universal Latent Space Model Fitting for Large Networks with Edge Covariates.” In: *J. Mach. Learn. Res.* 21, pp. 4–1.
- Mathis, A. et al. (2018). “DeepLabCut: markerless pose estimation of user-defined body parts with deep learning”. In: *Nature neuroscience* 21.9, pp. 1281–1289.
- Matias, C. and V. Miele (2017). “Statistical clustering of temporal networks through a dynamic stochastic block model”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.4, pp. 1119–1141.
- Matias, C., T. Rebafka, et al. (Sept. 2018). “A semiparametric extension of the stochastic block model for longitudinal networks”. In: *Biometrika* 105.3, pp. 665–680.
- McCormick, T. H. and T. Zheng (2015). “Latent surface models for networks using Aggregated Relational Data”. In: *Journal of the American Statistical Association* 110.512, pp. 1684–1695.
- McDaid, A. F. et al. (2013). “Improved Bayesian inference for the stochastic block model with application to large networks”. In: *Computational Statistics & Data Analysis* 60, pp. 12–31.
- Miscouridou, X. et al. (2018). “Modelling sparsity, heterogeneity, reciprocity and community structure in temporal interaction data”. In: *Advances in Neural Information Processing Systems* 31.
- Moreira-Matias, L. et al. (2013). “Predicting taxi–passenger demand using streaming data”. In: *IEEE Transactions on Intelligent Transportation Systems* 14.3, pp. 1393–1402.
- Morris, M. and M. Kretzschmar (1995). “Concurrent partnerships and transmission dynamics in networks”. In: *Social networks* 17.3-4, pp. 299–318.
- Nowicki, K. and T. A. B. Snijders (Sept. 2001). “Estimation and Prediction for Stochastic Block-structures”. In: *Journal of the American Statistical Association* 96, pp. 1077–1087.
- Ogata, Y. (1988). “Statistical models for earthquake occurrences and residual analysis for point processes”. In: *Journal of the American Statistical association* 83.401, pp. 9–27.
- Passino, F. S. and N. A. Heard (2021). “Mutually exciting point process graphs for modelling dynamic networks”. In: *arXiv preprint arXiv:2102.06527*.
- Pensky, M. and T. Zhang (2019). “Spectral clustering in the dynamic stochastic block model”. In: *Electronic Journal of Statistics* 13.1, pp. 678–709.
- Raftery, A. E. et al. (2012). “Fast inference for the latent space network model using a case-control approximate likelihood”. In: *Journal of computational and graphical statistics* 21.4, pp. 901–919.

- Rastelli, R. and M. Fop (2020). “A stochastic block model for interaction lengths”. In: *Advances in Data Analysis and Classification* 14.2, pp. 485–512.
- Rizoïu, M.-A. et al. (2017). “A tutorial on Hawkes processes for events in social media”. In: *arXiv preprint arXiv:1708.06401*.
- Rossetti, G. and R. Cazabet (2018). “Community discovery in dynamic networks: a survey”. In: *ACM Computing Surveys (CSUR)* 51.2, pp. 1–37.
- Rubin-Delanchy, P. et al. (2017). “A statistical interpretation of spectral embedding: the generalised random dot product graph”. In: *arXiv preprint arXiv:1709.05506*.
- Sarkar, P. and A. W. Moore (2006). “Dynamic social network analysis using latent space models”. In: *Advances in Neural Information Processing Systems*, pp. 1145–1152.
- Sewell, D. K. and Y. Chen (2015). “Latent space models for dynamic networks”. In: *Journal of the American Statistical Association* 110.512, pp. 1646–1657.
- Shalev-Shwartz, S. et al. (2012). “Online learning and online convex optimization”. In: *Foundations and Trends® in Machine Learning* 4.2, pp. 107–194.
- Sit, T. et al. (2021). “Event history analysis of dynamic networks”. In: *Biometrika* 108.1, pp. 223–230.
- Smith, A. L. et al. (2019). “The geometry of continuous latent space models for network data”. In: *Statistical science: a review journal of the Institute of Mathematical Statistics* 34.3, p. 428.
- Snijders, T. A. and K. Nowicki (1997). “Estimation and prediction for stochastic blockmodels for graphs with latent block structure”. In: *Journal of classification* 14.1, pp. 75–100.
- So, N. et al. (2015). “A social network approach reveals associations between mouse social dominance and brain gene expression”. In: *PloS one* 10.7, e0134509.
- Stan Development Team (2020). *RStan: the R interface to Stan*. R package version 2.21.2.
- Sun, S. et al. (2021). “ppdiag: Diagnostic Tools for Temporal Point Processes”. In: *Journal of Open Source Software* 6.61, p. 3133.
- Von Luxburg, U. et al. (2008). “Consistency of spectral clustering”. In: *The Annals of Statistics*, pp. 555–586.
- Vries, H. de (1998). “Finding a dominance order most consistent with a linear hierarchy: a new procedure and review”. In: *Animal Behaviour* 55.4, pp. 827–843.

- Vries, H. de and M. C. Appleby (2000). “Finding an appropriate order for a hierarchy: a comparison of the I&SI and the BBS methods”. In: *Animal Behaviour* 59.1, pp. 239–245.
- Ward, O. G. et al. (2021). “Next waves in veridical network embedding”. In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 14.1, pp. 5–17.
- Williamson, C. M. et al. (2016). “Temporal dynamics of social hierarchy formation and maintenance in male mice”. In: *Animal Behaviour* 115, pp. 259–272.
- Wu, J., T. Vallenius, et al. (2009). “Integrated network analysis platform for protein-protein interactions”. In: *Nature methods* 6.1, pp. 75–77.
- Wu, J., A. L. Smith, et al. (2021). “Diagnostics and Visualization of Point Process Models for Event Times on a Social Network”. In: *Applied Modeling Techniques and Data Analysis 1: Computational Data Analysis Methods and Tools* 7, pp. 129–145.
- Wu, J., O. G. Ward, et al. (2022). “Markov-Modulated Hawkes Processes for sporadic and bursty event occurrences”. In: *To Appear, Annals of Applied Statistics*.
- Xu, H., G. Fang, et al. (2018). “Latent class analysis of recurrent events in problem-solving items”. In: *Applied Psychological Measurement* 42.6, pp. 478–498.
- Xu, K. and A. Hero (2014). “Dynamic stochastic blockmodels for time-evolving social networks”. In: *IEEE Journal of Selected Topics in Signal Processing* 8.4, pp. 552–562.
- Yang, T. et al. (2011). “Detecting communities and their evolutions in dynamic social networks—a Bayesian approach”. In: *Machine learning* 82.2, pp. 157–189.
- Yin, G. G. and Q. Zhang (2012). *Continuous-time Markov chains and applications: a singular perturbation approach*. Vol. 37. Springer.
- Zhao, Y. et al. (2012). “Consistency of community detection in networks under degree-corrected stochastic block models”. In: *The Annals of Statistics* 40.4, pp. 2266–2292.

Appendix A: Additional Material for Chapter 4

A.1 Algorithms

Here we include Algorithm 4 for the online Hawkes process, as mentioned in the main text, along with Algorithm 5, which is used in this procedure. Some supporting functions in Algorithm 4 are given as below.

- $a+ = b$ represents $a = a + b$; $a- = b$ represents $a = a - b$.
- $impact(t)$ is given by $\sum_{t_1 \in timevec} \lambda \exp\{-\lambda(t - t_1)\}$.
- I_1 is given by $\sum_{t_1 \in timevec} \exp\{-\lambda(t - t_1)\}$.
- I_2 is given by $\sum_{t_1 \in timevec} (t - t_1) \lambda \exp\{-\lambda(t - t_1)\}$.
- $integral(t, t_{end}, \lambda)$ is given by $1 - \exp\{-\lambda(t_{end} - t)\}$.
- $integral(t, t_{start}, t_{end}, \lambda)$ is given by $\exp\{-\lambda(t_{start} - t)\} - \exp\{-\lambda(t_{end} - t)\}$.

As discussed in Chapter 4, we only need to store the sufficient statistics of the particular model under each setting. We show two examples in Table A.1. In the homogeneous Poisson case, we only need to store the cumulative counts for each pair of sender and receiver ($I_{user1, user2}$). In the Hawkes case, we only need to store the recent historical events since the impact of older information decays exponentially fast and thus has negligible effect on the current intensity.

A.2 Additional Simulations

Here we include additional simulations for the Hawkes process setting, similar to those considered in the main text for the homogeneous Poisson model. We demonstrate community recovery

Table A.1: The Data Structure for Storing Historic Events. The upper diagram shows the structure under the Poisson model, where the key is the pair of nodes and the value is the corresponding cumulative number of all past events. The bottom diagram shows the structure under the Hawkes model, where the key is still the nodes and the value is the corresponding time sequence between $t_{current} - R$ and $t_{current}$ stored in **queue** structure.

Poisson		Hawkes	
Key	Value	Key	Value
(User1, User3)	$l_{user1,user3}$	(User1, User3)	$t_{user1,user3}^{(start)}, \dots, t_{user1,user3}^{(end)}$
(User3, User8)	$l_{user3,user8}$	(User3, User8)	$t_{user3,user8}^{(start)}, \dots, t_{user3,user8}^{(end)}$
(User3, User1)	$l_{user3,user1}$	(User3, User1)	$t_{user3,user1}^{(start)}, \dots, t_{user3,user1}^{(end)}$
(User2, User4)	$l_{user2,user4}$	(User2, User4)	$t_{user2,user4}^{(start)}, \dots, t_{user2,user4}^{(end)}$
(User3, User5)	$l_{user3,user5}$	(User3, User5)	$t_{user3,user5}^{(start)}, \dots, t_{user3,user5}^{(end)}$
\vdots	\dots	\vdots	\dots
(User5, User3)	$l_{user5,user3}$	(User5, User3)	$t_{user5,user3}^{(start)}, \dots, t_{user5,user3}^{(end)}$
(User8, User3)	$l_{user8,user3}$	(User8, User3)	$t_{user8,user3}^{(start)}, \dots, t_{user8,user3}^{(end)}$
(User9, User2)	$l_{user9,user2}$	(User9, User2)	$t_{user9,user2}^{(start)}, \dots, t_{user9,user2}^{(end)}$
(User7, User1)	$l_{user7,user1}$	(User7, User1)	$t_{user7,user1}^{(start)}, \dots, t_{user7,user1}^{(end)}$

and other properties of our online inference procedure. We also expand on some issues components of the inference procedure discussed in Chapter 4.4. Finally, we compare the performance of

Window Size Throughout the simulation studies in Chapter 4 we have used a fixed window size such that $dT = 1$. Here we wish to investigate the effect that varying this window size has on the performance of our algorithm. We compare community recovery for varying window sizes from $dT = 0.25$ to $dT = 5$. ARI scores across 50 simulations for each are shown in Figure A.1. For the smallest values of dT we see somewhat improved performance at community recovery. This is unsurprising as the window size gets smaller we are processing a small number of events in each window. For sufficiently small windows, this could lead to a single event in each window, mirroring standard Stochastic Variational inference (Hoffman, Blei, et al., 2013). However, for dT in the range of 1 to 5 the performance is similar regardless of window size. Sufficiently small windows allows our online inference procedure to avoid getting stuck in local optima of the current

estimate of the ELBO.

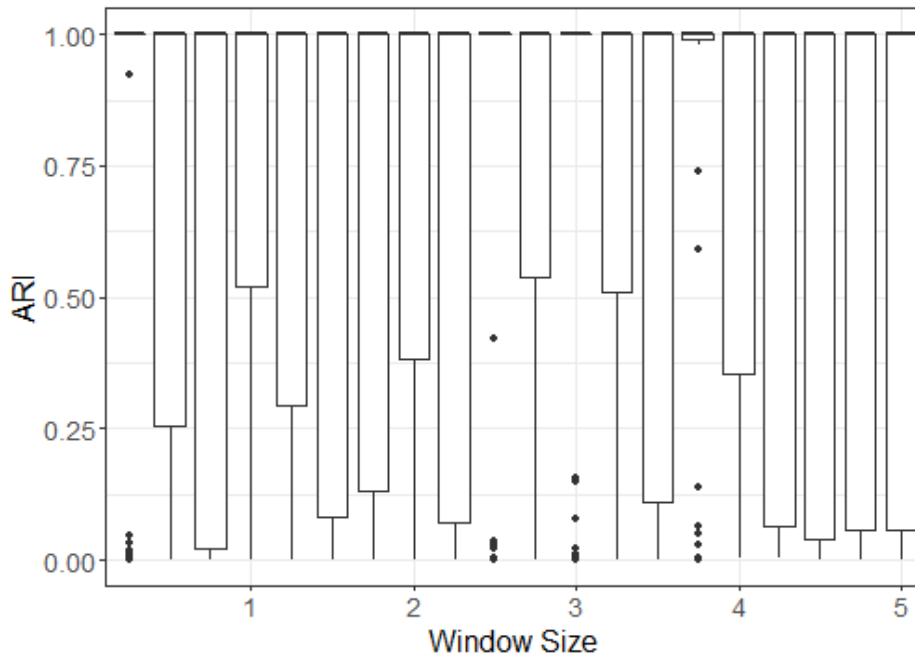


Figure A.1: Community recovery under Poisson simulated data for varying window size.

Hawkes Community Recovery The main experiments in Chapter 4 demonstrate the performance of our online learning algorithm where the intensity function follows a Poisson intensity function. Here we demonstrate the performance of this procedure for Hawkes block models also. In Figure A.2 we first investigate the performance of our procedure for community recovery, as we increase the number of nodes. As the number of nodes increase, we can more consistently recover the true community structure.

Hawkes, Varying Number of Communities We can also investigate the performance of our procedure for community recovery under the Hawkes model as we increase K , the number of communities. In Figure A.3 we show the performance as we consider more communities for a fixed number of nodes. As K increases, we are less able to recover the true community structure, which is seen by a decrease in the ARI.

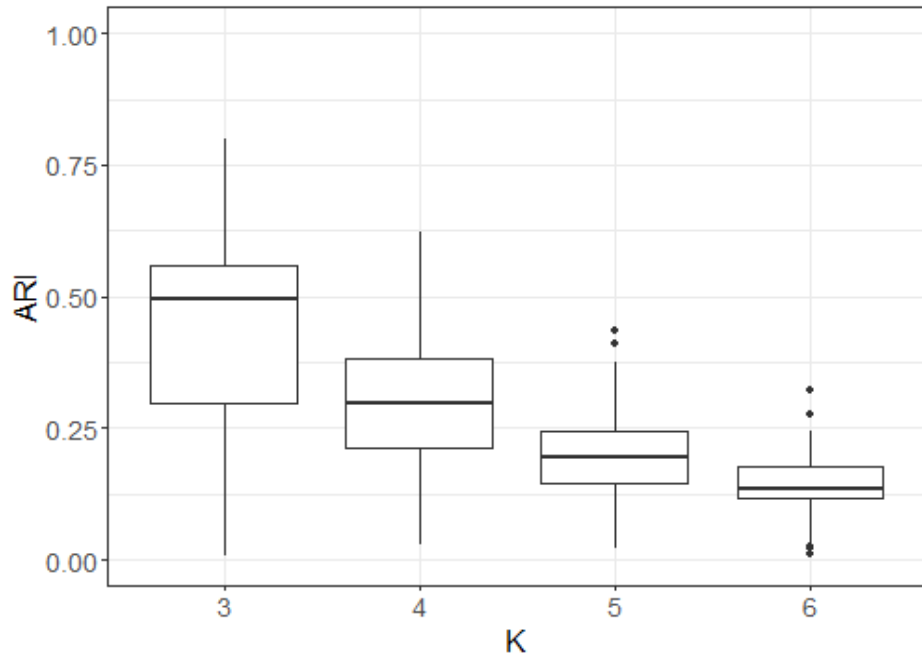


Figure A.3: Community recovery for the Hawkes model as we vary the number of communities.

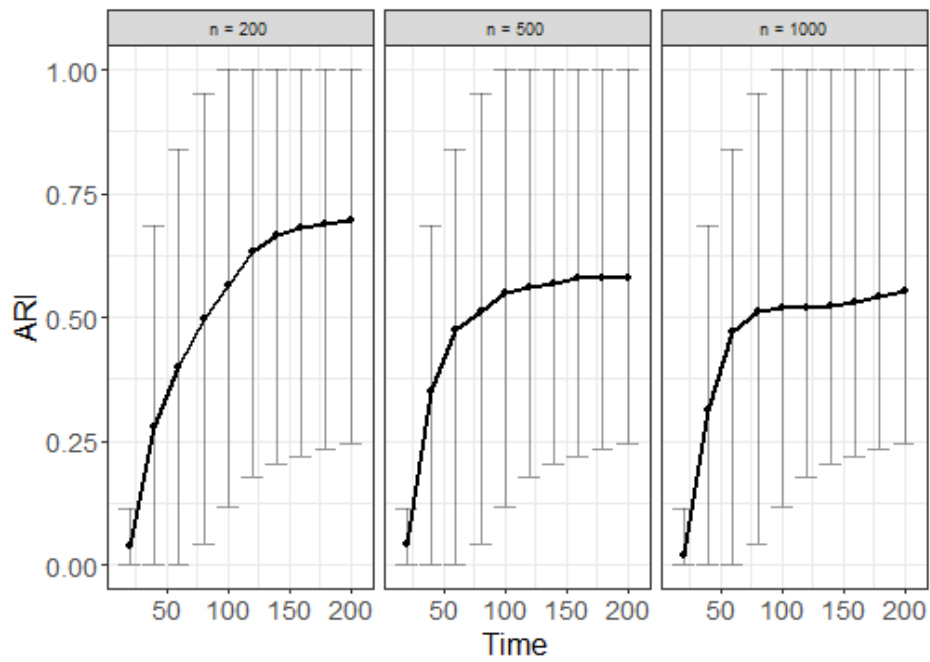


Figure A.4: Online community recovery under the Hawkes model as the number of nodes increases, for a fixed observation period.

Algorithm 4 Online-Hawkes

- 1: Input: $data$, number of groups K , window size dT , edge list A .
 - 2: Output: $\hat{\mu}, \hat{B}, \hat{\lambda}, \hat{\pi}$.
 - 3: Initialization: $S, \tau, \pi, B, \mu, \lambda$.
 - 4: Set $M = T/dT$ and create an empty map \mathcal{D} .
 - 5: **for** window $m = 1$ to M **do**
 - 6: Read new data between $[(m - 1) \cdot dT, m \cdot dT]$ and apply **Trim**.
 - 7: Create temporary variables: $\mu_{p1}, \mu_{p2}, B_{p1}, B_{p2}, S_p$.
 - 8: Set learning speed: $\eta = \frac{K^2}{\sqrt{mm_t}}$, where m_t is the number of events between $[(m - 1) \cdot dT, m \cdot dT]$.
 - 9: **for** key (i, j) in \mathcal{D} **do**
 - 10: Create sub temporary K by K matrix variables: $\mu_{p1,tp}, B_{p1,tp}, B_{p2,tp}, S_{p,tp}$ and λ_{st} .
 - 11: Update μ_{p2} by setting $\mu_{p2}(k, l) += \tau_{ik}\tau_{jl}dT$ for $k, l \in [K]$.
 - 12: Update S_p by setting $S_p(i, k) -= \tau_{jl}\mu_{kl}dT$.
 - 13: Get time stamps, $timevec$, corresponding to (i, j) .
 - 14: **for** t in $timevec$ **do**
 - 15: **if** $t > (m - 1)dT$ **then**
 - 16: Compute the impact function value, $impact(t)$.
 - 17: Compute I_1 and I_2 .
 - 18: Compute Λ , where $\Lambda(k, l) = \mu_{kl} + B_{kl} impact(t)$.
 - 19: $\lambda_{st} += B \cdot (I_1 - I_2)/\Lambda - B \cdot (T_e - t) \exp\{-\lambda(T_e - t)\}$.
 - 20: $\mu_{p1,tp}(k, l) += 1/\Lambda(k, l)$.
 - 21: $B_{p1,tp}(k, l) += impact(t)/\Lambda(k, l)$.
 - 22: $S_{p,tp}(k, l) += \log(\Lambda(k, l))$.
 - 23: $B_{p2,tp}(k, l) += integral(t, t_{end}, lam)$.
 - 24: **end if**
 - 25: **if** $t \leq (n - 1)dT$ **then**
 - 26: $B_{p2,tp} += integral(t, t_{start}, t_{end}, lam)$.
 - 27: $\lambda_{st} += B_{kl}(T_s - t) \exp\{-\lambda(T_s - t)\} - (T_e - t) \exp\{-\lambda(T_e - t)\}$.
 - 28: **end if**
 - 29: **end for**
 - 30: $\mu_{p1}(k, l) += \tau_{ik}\tau_{jl}\mu_{p1,tp}(k, l)$.
 - 31: $B_{p1}(k, l) += \tau_{ik}\tau_{jl}B_{p1,tp}(k, l)$.
 - 32: $B_{p2}(k, l) += \tau_{ik}\tau_{jl}B_{p2,tp}(k, l)$.
 - 33: $S_p(i, k) += \sum_l \tau_{jl}(S_{p,tp}(k, l) - B_{kl}B_{p2,tp}(k, l))$.
 - 34: **end for**
 - 35: $S += S_p$.
 - 36: Compute the negative gradients: $grad_B = B_{p1} - B_{p2}$, $grad_\mu = \mu_{p1} - \mu_{p2}$, $grad_\lambda = \sum_{kl} \tau_{ik}\tau_{jl}\lambda_{st}(k, l)$.
 - 37: Update parameters: $B = B + \eta \cdot grad_B$, $\mu = \mu + \eta \cdot grad_\mu$, $\lambda = \lambda + \eta \cdot grad_\lambda$.
 - 38: Update τ by setting $\tau_{ik} = \frac{\pi_k S_{ik}}{\sum_k \pi_k S_{ik}}$ for $i \in [n]$ and $k \in [K]$.
 - 39: Update π by setting $\pi_k = \frac{1}{n} \sum_i \tau_{ik}$ for $k = 1, \dots, K$.
 - 40: **end for**
-

Algorithm 5 Trim

```
1: Input:  $\mathcal{D}$ , truncated length  $R$ , current time  $t_{current}$ ,  $data_{new}$ .
2: Output:  $\mathcal{D}$ .
3: for  $event$  in  $data_{new}$  do
4:   Get node pair  $(i, j)$  and time stamp  $t$ .
5:   if key  $(i, j)$  is already in  $\mathcal{D}$  then
6:     We get the corresponding queue. We then push  $t$  at the back of this queue and update  $\mathcal{D}$ .
7:   end if
8:   if key  $(i, j)$  does not exist in  $\mathcal{D}$  then
9:     We create an empty queue, push  $t$  to it and update  $\mathcal{D}$ .
10:  end if
11: end for
12: for key  $(i, j)$  in  $\mathcal{D}$  do
13:   Get the queue  $timequeue$  corresponding to key  $(i, j)$  and let  $t_{front}$  be the first element of  $timequeue$ .
14:   while  $t_{current} - t_{front} > R$  do
15:     Pop the first element of  $timequeue$ .
16:     Set  $t_{front}$  be the first element of current  $timequeue$ .
17:   end while
18: end for
```

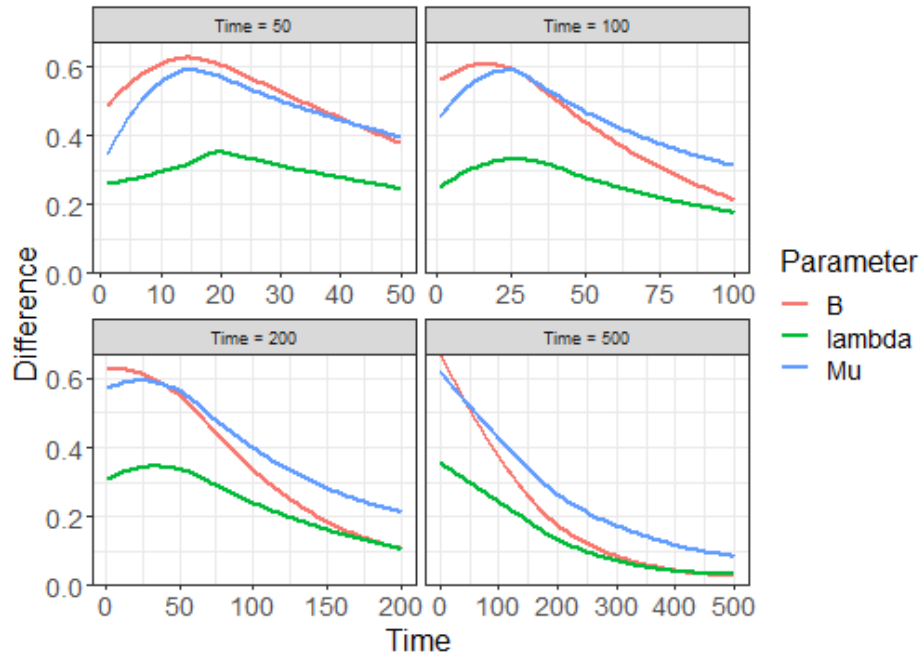


Figure A.5: Parameter Recovery for block Hawkes model as the observation time increases for a fixed network size.